Jens Tonne

Robust Model Predictive Control for Large-Scale Manufacturing Systems subject to Uncertainties

Jens Tonne

Robust Model Predictive Control
for Large-Scale Manufacturing Systems subject to Uncertainties

Die Ergebnisse, Meinungen und Schlüsse dieser Dissertation sind nicht notwendigerweise die der Volkswagen AG.

Contents

Acknowledgments

First of all, I would like to thank my advisor Prof. Olaf Stursberg for his support and encouragement during the last years. His input and advice during numerous discussions mainly contributed to my research and this thesis. Moreover, I would like to thank Prof. Andreas Kroll, Prof. Arno Linnemann, and Prof. Bernhard Sick for being on the thesis committee. Additionally, I would like to thank Prof. Arno Linnemann for his valuable advice concerning stability theory.

A special thanks goes to my friend and former colleague Martin Jilg. In various discussions and meetings on- and off-topic, he influenced my scientific career and contributed to the decision to start the endeavor that resulted in this thesis.

I would like to express my special appreciation and thanks to my superiors at Volkswagen Kassel, Dr. Jörg Clobes and Dr. Michael Alsmann, who strongly supported and encouraged me during my studies. They gave me the opportunity and the freedom needed to conduct my research and complete this thesis. In addition, I would like to thank my colleagues at Volkswagen Kassel and at the department of Control and System Theory at the University of Kassel who supported me in many occasions and provided fruitful advice and discussions: Leonard Asselborn, Jan Eilbrecht, Patricia Engelhardt, Ilda Hujdur, Damian Kontny, Zonglin Liu, Manuel Opfer, Konstantin Schaab, Moritz Theißen, and Tim Wicke. Moreover, several students and interns supported this work directly or indirectly by taking some of the workload off me. In this context, I would like to thank Tobias Becker, Rami Elshinawy, Daniel Gleim, Jannik Huber, Martin Comiche Perez, Tim Schaake, Lena Elisa Schneegans, Niklas Riemann, and Yanbing Zhou.

Last but not least, special thanks go to my family – in particular my parents Iris Theis and Jörg Tonne-Jenett as well as my sister Saskia Theis – for their constant support and encouragement, and to Svea Rieping, who encouraged me and cheered me up – especially during the hard times and long hours of working and writing.

Without you this would not have been possible!

Jens Tonne
Kassel, 2017

Summary

Large scale manufacturing systems are often run with constant process parameters although continuous and abrupt disturbances influence the process. To reduce quality variations and scrap, a closed-loop control of the process variables becomes indispensable. In this thesis, a modeling and control framework for multistage manufacturing systems is developed, in which the systems are subject to abrupt faults, such as component defects, and continuous disturbances. In this context, three main topics are considered: the development of a modeling framework, the design of robust distributed controllers, and the application of both to the models of a real hot stamping line. The focus of all topics is on the control of the product properties considering the available knowledge of faults and disturbances.

To account for the abrupt disturbances, each production unit of the overall manufacturing system is modeled by a jump Markov system (JMS). The continuous dynamics of the JMS may be nonlinear, and is used to model the state of the production system and the properties of the products being processed. The Markov chain models the occurrence of faults, where the corresponding transition probabilities are calculated from the recording of earlier faults. The dynamics of the production units are coupled by the subsequent processing of the products.

For the control of the manufacturing system, the use of robust distributed model predictive control (DMPC) is proposed. In the setup, each production unit (subsystem) is controlled by a local MPC that may communicate with the MPCs of the other production units in order to minimize a global cost function. For the formulation of the optimization problems, efficient algorithms for the prediction of the expected value of the states and the costs are presented. To guarantee recursive feasibility, robust control invariant sets are calculated offline for all subsystems. Employing these sets, the local MPCs are formulated as quadratically constrained quadratic programs, that can be solved fast enough for an online execution. The local MPCs are robust to the transitions of the Markov state and the disturbances, consider state and input constraints, and guarantee closed-loop stability.

These approaches are tested with models of an existing hot stamping line. First, the robust MPC approach is tested for a single production unit employing the model of a roller hearth furnace. The results show that the MPC approaches can be used for both reference tracking applications and for economic MPC setups. Finally, the DMPC approaches are applied to the model of the overall production line and are compared to a standard control setup. The results illustrate the potential of the developed modeling and control framework in terms of costs and robustness, and they illustrate the effects of the cooperation between the local controllers.

Part I.

Introduction and Theoretical Background

1. Introduction and Literature Review

1.1. Introduction

In industrial mass manufacturing processes, e.g., in the automobile or electronics industry, piece goods are produced in high volumes with low cycle times. In addition to realizing high production rates, the main goal is to produce goods with predefined properties that vary as little as possible over time. However, most manufacturing processes are influenced by continuous disturbances, e.g., temperature drifts, tool wear, noise processes, or dirt buildup, and by abrupt – often more severe – disturbances, like component faults or base material changes (batch changes). These disturbances may cause undesired quality variations or even violation of specifications. Furthermore, in multistage manufacturing processes, variations propagate through the processes and may influence all downstream production steps in a negative way. Despite the ongoing and upcoming process control trends, for example, in the context of "Industry 4.0", many processes are still run with a constant set of production variables, which have been optimized before the start of production. Often, the process variables are only adapted manually by the operating personnel based on experience, if significant process variations occur (caused for example by the disturbances mentioned above). This is the case in, e.g., stamping, hot stamping, casting, or hardening processes. For these processes, a robust, closed-loop control of process variables provides the potential to significantly reduce quality variations, scrap rates, and resource consumption.

Manufacturing process control is often integrated in an enterprise planning and control hierarchy, as shown in Fig. 1.1 [85]. On the top level, enterprise wide planning with enterprise resource planning systems is applied. It provides the basis for production planning, which determines the amounts of product's to be produced. The fourth layer provides schedules defining which products are produced when and on what production line. Based on these schedules, optimal operation set points are determined for the specific production lines, possibly consisting of several production units (plant-wide optimization). The local control laws (often model predictive control (MPC) approaches) determine the process variables in order to keep the production process, i.e., the product properties, as close as possible to the specified optimal set points. Finally, the actuators are controlled (often by PI or PID controllers) in order to realize the specified process variables (cf. [107, 113]).
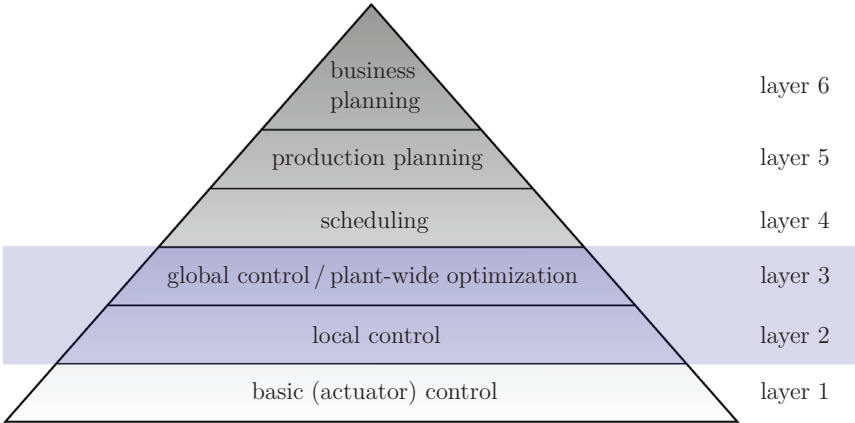
Figure 1.1.: Production planning, scheduling, and control hierarchy (based on hierarchies in [85, 107]). This thesis focuses on the highlighted layers.

This thesis is concerned with an optimal process variable control for multistage manufacturing processes with an emphasis on the robustness to continuous and abrupt disturbances (also faults). Each production stage may consists of several different production units. Hence, the concepts of this thesis can be regarded as a global control of a production line, as well as the local control of the process variables of the production units (layers 2 and 3 in Fig. 1.1). The main focus of the control is to determine optimal processes variables such that quality variations are minimized and the resulting product properties meet all specifications despite the presence of disturbances and faults. In each production unit, continuous disturbances as well as abrupt stochastic disturbances influencing the product properties may occur. It is assumed that the product properties are measurable or observable in some production units. Furthermore, the occurrence of abrupt disturbances, such as component break downs or production line halts, can be detected by sophisticated self-diagnosis and monitoring systems in modern production units[1]. The production stages are coupled by the (properties of) the products processed by and transferred between the production units. Hence, the disturbances at one production unit may influence all downstream production stages. For this problem setup, a modeling framework and robust control approaches are developed in this thesis.

The main challenge of controlling the production system is that state and input constraints have to be satisfied robustly for all disturbances. The state constraints

---

[1]In addition, there exist numerous approaches to fault detection and isolation in literature and practice, see, e.g., [155].

Figure 1.2.: A hot stamping line with two stacked heating furnaces (left) and the press in the picture's center. The picture was taken at the Volkswagen plant in Baunatal.

consider specifications on the product properties and safety bounds for the operation of the production units. The input constraints result from actuator limits. The most common control approach capable of handling constraints is model predictive control (MPC) [71, 88, 93, 107]. In MPC, a system model is used in order to predict the state trajectory for a given prediction horizon as a function of the input trajectory. Using this prediction model, an optimal input trajectory, that considers input and state constraints, is determined by solving optimization problems online.

In addition, the distributed structure of a multistage manufacturing system introduces further challenges to the overall control problem. Due to the coupling of the different production stages, a simple approach with independent local (decentralized) MPCs for each production unit may result in a poor overall control performance, and a coordination between the production unit controllers becomes necessary. Hence, the development of a distributed MPC for the presented modeling framework is another topic of this thesis.

To illustrate and evaluate the modeling and control approaches in a realistic setup, the whole concept is investigated by simulations with models from a real hot stamping manufacturing line (cf. Fig. 1.2). Hot stamping is a process for producing high-strength car body parts [65, 98]. The process is influenced strongly by numerous disturbances ranging from line halts or defects of heating systems to pressing tool wear and dirt accumulations. The process has a larger number of process variables, which are not controlled in closed-loop currently. This renders hot stamping a suitable test scenario for the developed approaches.

The following sections provide a literature review and the concrete aim and contribution of this dissertation. Finally, the outline of this thesis is presented.

1.2. Literature Review

This section provides a review of the existing literature on modeling and control of manufacturing systems and fault tolerant control. In addition, the state of the art in modeling and control of hot stamping production lines is described.

1.2.1. Modeling and Control of Manufacturing Processes

There are numerous approaches and research areas concerning modeling and control of manufacturing processes. First, this section gives a brief overview of general modeling and control frameworks commonly used in the context of manufacturing systems. Then, specific approaches for multistage processes are presented.

General Modeling and Control Frameworks used for Manufacturing Systems

Manufacturing processes exhibit both a continuous and a discrete event behavior. The continuous behavior results from the processing, such as heating, forming, etc. Discrete events result from processing starts, halts, and ends as well as faults. Hence, the modeling and control approaches can be categorized by whether they are concerned with the continuous behavior, the discrete event behavior, or both.

Continuous Behavior Common continuous variable models based on (partial) differential equations or difference equations have been proposed for various sorts of manufacturing processes (see, e.g., [73]) as well as their robust control (e.g., [50]). Due to the sheer number of different manufacturing processes, this chapter rather aims at presenting the different modeling and control frameworks often used in the context of manufacturing instead of reviewing specific models.

In addition to the direct control of process variables and product properties, the modeling and control of production volumes, inventory, and corresponding quantities is an active field of research. In this context, dynamic production models describe the number of products rather by (product) flows and flow rates ("fluid models" [79]) than by discrete values. For these continuous formulations conservation constraints [52] or partial differential equations known from traffic modeling [54, 79] can be employed. The resulting continuous dynamics are used, e.g., for production planning and control based on linear programming [52]. Furthermore, common control theory, including stability analysis and controller design, is applied by [54, 79]. However, these approaches are used mainly in the production planning and scheduling layers, and the influence on the product properties is neglected.

Discrete Event Behavior The modeling and control of the discrete event behavior of manufacturing systems is considered thoroughly in literature. The discrete event behavior includes the sequence of processing in each production unit, resource allocation, transportation between production units, routing between production units,

and its scheduling (see, e.g., [26] and references therein). The most common discrete event models proposed in the manufacturing context are various forms of deterministic and stochastic Petri nets (cf. [26, Ch. 1 & 3][27]) as well as deterministic and stochastic finite state automata (cf. [26, 27]). For both Petri nets and automata, supervisory control concepts have been proposed, e.g., for resource allocation [26, Ch. 6-10].

Queuing Theory A well established theory for the analysis of the steady state behavior of production systems is queuing theory [26, 27, 54]. This framework can be used for modeling, design, and analysis of the stochastic behavior of buffers (queues) and servers, i.e., production units, in terms of average waiting times, processing times, or buffer usage. However, this kind of analysis is performed at the scheduling layer and is not considered in this thesis.

Hybrid Models To describe the continuous process behavior affected by uncontrollable discrete events, both the continuous and discrete event behavior have to be combined in one model. This combination results in hybrid models [26, 27]. For most manufacturing applications, the control of the discrete behavior operates at a higher control level triggering the control of the continuous processing.

An optimal control approach employing hybrid system models, in which the continuous dynamics describe the physical properties of the processed products and the discrete events describe start and end times of this process, is proposed in [105]. Its application to steel annealing is described in [34]. Here, a sequence of production line speeds is determined in order to achieve an optimal compromise between quality and processing time. However, the reaction to (abrupt) faults is not considered.

In hybrid Petri nets, the common marking concept (often modeling the production line status) can be extended by continuous places and corresponding marking flows that model the product flow (see [26, Ch. 3 & 14] for applications in inventory control). Similarly, hybrid automata, in which discrete events describe production line breakdowns, repairs, or restarts, are used for production rate control and routing [12]. The stochastic nature of these events can be modeled by Markov chains, where the transition probabilities are given by, e.g., break-down probabilities or repair rates. For each state of the Markov chain, different continuous dynamics can be used to model the state of the production unit. The resulting system belongs to the class of jump Markov systems (JMS) [39, 120]. This framework can be used to control production rates [2] or maintenance intervals [21]. However, there are no approaches known to the author that model the influence of stochastic faults on the product properties.

Modeling and Control Approaches for Multistage Processes

For multistage manufacturing systems, the question of how variations of the properties of semi-finished products at certain production stages affect the properties

of the final product is of great interest. The so-called *stream of variation* (SoV) modeling and analysis addresses this question [53, 60, 118]. In this framework, the deviation of product properties at the n-th production stage is described by a vector x^n. The propagation of deviations from one production stage x^n to another x^{n+1} is described (similar to discrete-time state space models) by a linear [53, 60, 118] or a nonlinear [62] equation. By recursive application of these equations, the propagation of variations and the influence of the inputs on the product quality can be formulated similar to condensed MPC formulations (cf. Section 2.5). The resulting equations can be used to determine inputs u^n that minimize a quality-related cost function [53]. These approaches are used, e.g., in assembly processes [60] or wafer production [53]. The advantage is the easy way of prediction and control of the deviations of the product properties. However, the only dynamics considered in this framework are (auto-regressive) disturbance models [53]. Other dynamics, such as temperature dynamics in the different production stages, are not considered.

Control approaches for dynamical multistage processes are often called *plant-wide control* and often arise in the chemical process industry, see, e.g., [28] and the references therein. In the context of plant-wide control, different control architectures, which are also known from control of distributed systems, are proposed [28, 113]:

- In **centralized control**, one global control law determines the input signals for all production stages. The measurements and control actions are normally transmitted by a communication network (Fig. 1.3 (a)).

- In **decentralized control**, each production unit is controlled by a local control law that determines the inputs independently of the other units (Fig. 1.3 (b)).

- **Distributed control** is a combination of centralized and decentralized control. Each production unit is controlled by a local controller that exchanges information with the other controllers via a communication network (Fig. 1.3 (c)).

- In **hierarchical control**, the production units are controlled by local control laws. These local control laws are coordinated by a global entity, often based on real time optimization (Fig. 1.3 (d)).

In most approaches considered in literature, the local controllers are MPC formulations which can cope with constraints, time-variant behavior, and nonlinearities, cf. [28, 113]. For a more detailed overview of decentralized and distributed MPC approaches see Section 10.1.

Combined Scheduling and Control

Some approaches aim at determining optimal schedules and process variable trajectories (for a certain planning horizon) at the same time [99, 100, 159]. The problem results in an open-loop optimization problem in the form of mixed integer programs.
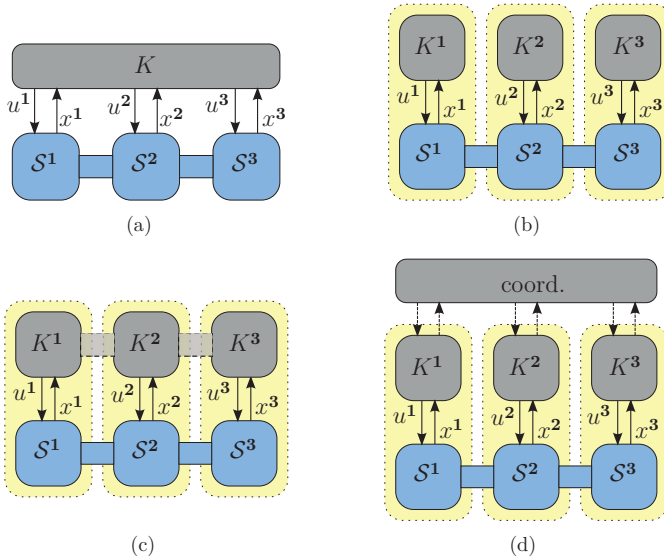
Figure 1.3.: Illustration of centralized (a), decentralized (b), distributed (c), and hierarchical control architectures (d), cf. [129]. The yellow boxes indicate the localization of the controllers.

A closed-loop type behavior can be achieved if a recalculation is performed once the actual value of the continuous state deviates from the predicted trajectory more than a predefined threshold [159]. If the recalculation is performed at fixed intervals, a sort of MPC is established for simultaneous process variable control and scheduling. An application to slab reheating has been proposed in [47, 128]. However, the proposed optimization problems are computationally costly which renders this concept inapplicable to process variable control of the most mass production processes. Furthermore, none of these approaches consider stochastic faults[2].

More general approaches to handle faults (not necessarily in the context of manufacturing systems) have been investigated in the context of fault tolerant control.

1.2.2. Fault Tolerant Control

The field of fault tolerant control aims at developing approaches to control plants that are subject to faults, such as actuator or sensor breakdowns. Since the early

---

[2]In [128], production line halts are considered. These halts are caused if the slab temperature is too low at the end of the furnace. Since the slab temperature is the controlled variable, these halts are deterministic and can be influenced by the control actions.

70s, the topic is enjoying a steady interest – especially in aerospace, marine, and chemical process applications, but also in the context of manufacturing systems. For an overview of approaches and applications see [20, 96, 155]. In the context of fault tolerant control, two main fields are considered [20, 155]:

1. fault detection and diagnosis (including fault isolation and estimation of its magnitude) and

2. the adaption of control laws to detected faults.

This thesis focuses on the last topic. The interested reader is referred to [155] and the references therein for an overview of theoretical and practical results on fault detection and diagnosis.

Furthermore, approaches to fault tolerant control can be divided into:

- **Passive approaches**: one control law is designed offline to robustly control the plant during nominal operation and faults. This coincides with the large field of robust control (see, e.g., [50]).

- **Active approaches**: the control law is adapted in order to accommodate to faults [155].

Often, better control results can be achieved with active approaches, if the necessary online computation time is small enough with respect to the cycle time. In this thesis, only active approaches are considered. Active approaches can be classified as reactive approaches, i.e., the control law is adapted once a fault has occurred, or proactive approaches that use information about possible future faults, e.g., component breakdown probabilities, and proactively adapt the control actions.

Reactive Approaches

The majority of fault tolerant control approaches propose a reactive setup. Corresponding approaches have been proposed for most common system models, such as linear models [20, 155], nonlinear models [66, 96, 155], discrete-event models [26, Ch. 23], switched systems [42], and hybrid models (see [147] and the references therein). Recently, fault tolerant MPC has been proposed frequently (often in the context of chemical processes) [66, 96, 155]. The basic idea of these approaches is to apply a common stabilizing MPC scheme as long as the system is in nominal operation. If a fault is detected, the prediction model, the constraints, the terminal set, and / or the reference are changed according to the (new) dynamics of the faulty system [66]. In addition, few fault tolerant MPC schemes for distributed systems have been proposed (e.g., in [30]). However, these approaches do not employ any knowledge of possible future faults or their probability of occurrence. Hence, the prediction of the plant behavior is not accurate and does not include all knowledge that is available.

Proactive Approaches

In order to prepare for faults, the effect of future faults can be included in an MPC scheme directly. One possibility is to predict the future behavior for a set of possible fault scenarios [11] or even all possible faults [25] within the prediction horizon and ensure that state and input constraints are satisfied for all scenarios. These approaches establish a type of scenario-based robust fault tolerant MPC. In other approaches, denoted as *proactive fault tolerant control* [69, 76], it is assumed that the occurrence of (actuator) faults in a certain time interval is known in advance ("incipient faults"). This knowledge is used to adapt the prediction model and constraints in order to drive the system state into a safe (control invariant) set [69]. However, all these MPC approaches do not consider the fault probabilities directly.

In common fault tolerant control approaches that consider fault probabilities, robust $\mathcal{H}_\infty$-norm optimal feedback laws are determined for JMS that describe the fault-prone system [119]. However, MPC approaches for fault tolerant control considering fault probabilities are rarely stated in literature. In [90], a fault tolerant MPC for a nonlinear discrete-time system is proposed, in which the faults are modeled as an additive Gaussian process. While this approach addresses dynamic faults well, abrupt faults, like component breakdowns, can be modeled more accurately by switched stochastic systems, such as JMS. While MPC approaches for JMS are rarely stated in the context of fault tolerant control (see, e.g., [55]), the topic itself enjoys increasing attention in literature. For a detailed review, see Section 5.1.

1.2.3. Hot Stamping

As stated in the introduction, the hot stamping process is used as an example process throughout this thesis. Hence, the state of the art in hot stamping is reviewed in this section. Hot stamping of sheet metal is a manufacturing process for the production of car body parts with high strength, like B-pillars [65, 98]. The hot stamping process is shown in Fig. 1.4. First, the sheet metal blanks are cut out of a coil. Commonly, the manganese–boron steel 22MnB5 with an aluminum-silicon coating is used as
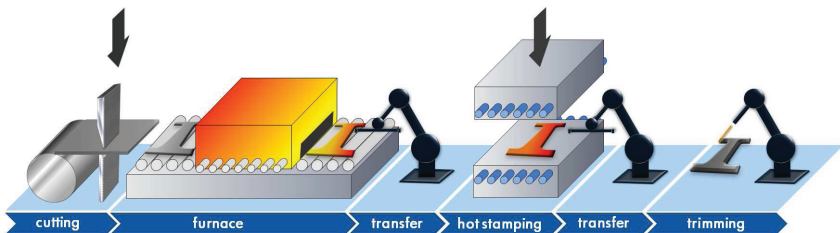
Figure 1.4.: Sketch of the hot stamping process.
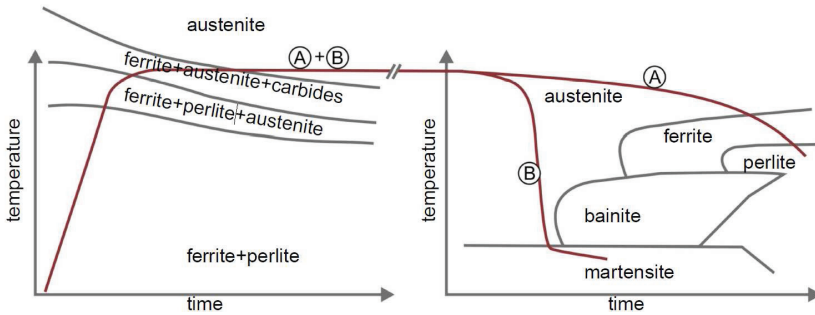
Figure 1.5.: Time-temperature routes of the blanks during heating (left) and quen-
ching (right) [125]. Route A shows a slowly cooled part resulting in a
soft material. Route B demonstrates a temperature route with a cooling
rate higher than 27 $^K/_s$ resulting in a fully martensitic material.

base material. Then, the blanks are heated up to 900 °C - 950 °C in a roller hearth
furnace (RHF) [65]. Subsequently, the hot blanks are formed and quenched at the
same time in a press with a water-cooled pressing tool. The quenching results in a
martensitic matrix with a tensile strength exceeding 1500 MPa. Finally, the parts
are trimmed to their final shape [65, 98].

The research and industrial development concerning hot stamping focuses on
the determination of process windows that guarantee the desired properties, the
production of parts with tailored properties, process simulation and design, and –
more recently and rarely – non-destructive testing as well as process control[3]. The
advances in these areas are summarized in the following paragraphs.

Process Window The final mechanical and tribological properties of the parts are
determined by the time-temperature route of the blanks. The mechanical properties,
such as tensile strength, hardness, uniform elongation, and geometry, are mainly
determined by the properties of the base material. To reach the high tensile strength,
the base material has to be fully austenitized and quenched fast enough (cf. route
B in Fig. 1.5). The cooling rate during the pressing has to exceed 27 $^K/_s$ [77]. A
thorough investigation of the minimum furnace time for a full austenitization of the
base material can be found in [77]. Finally, the quenching should not be stopped
before the blank temperature is below the martensite finish temperature M_f, to
avoid local annealing and minimize springback. The martensite finish temperature
M_f for the alloy considered is about 230 °C [97].

---

[3]Other important research topics concern the wear behavior of the tools and the development of
alternative materials, coatings, as well as post-processing steps [65, 98]. However, these fields
are not further explored in this thesis.
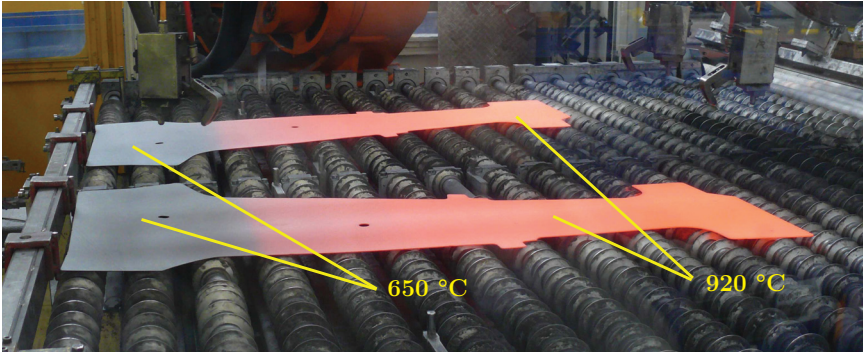
Figure 1.6.: Two B-pillars after the heating in a roller heart furnace and subsequent local cooling in the lower part of the B-pillar (left). The picture was taken at the Volkswagen plant in Baunatal.

Paint adhesion and weldability are determined by the properties of the coating. Spot weldability decreases with increasing furnace time, while a minimum furnace time is required to guarantee good paint adhesion. Constraints for heat treatment time and heating rate, that ensure paint adhesion and weldability, have been proposed in [121].

All in all, there is a rich body of literature on constraints that guarantee the desired final part properties. These constraints can be used for process design and for the MPC parameterization in this thesis.

Tailored Properties Parts with tailored properties, e.g., locally differing tensile strength and hardness, are produced, in order to optimize the load distribution and energy absorption during a crash. For example, the lower part of a B-pillar is of lower hardness than the upper part in order to absorb the energy of a side impact in the lower area while maintaining the structural integrity in the upper part of the car which is close to the passengers.

To produce parts with tailored properties, the martensitic transformation of the base material has to be prevented in those sections that are supposed to be of lower hardness (cf. route A for the soft part and route B for the hard part in Fig. 1.5). During the last years, several process variants for the production of parts with tailored properties have been developed [65, 95]: A lower hardness can be achieved by keeping the temperature below the austenitization temperature Ac_1 (which is at about 750 °C [98]) in the intended areas before the forming and quenching operation is started [77]. This can be achieved by differential heating, i.e., by only locally heating some areas above the austenitization temperature Ac_3 [36], or by cooling the other areas slowly below the austenitization temperature Ac_1 subsequent to the

furnace process [95]. Alternatively, the cooling rate during the quenching operation can be reduced locally below 27 $^{K}/_{s}$ by partial heating of the tool or by tools with spatially varying heat transfer coefficients [65, 95, 98].

In this thesis, only partial cooling on ambient air at the end of the furnace, referred to by *partial hardening*, is considered. In Fig. 1.6, two B-pillars are shown directly after the local cooling and before the transport to the press. The different blank temperatures are clearly visible by the different colors of the upper and lower parts of the blanks. To achieve the desired spatial distribution of the hardness, corresponding bounds for the local blank temperatures have to be considered by the MPC.

Process Design and Simulation   For process design and optimization, a simulation of the different processing steps has become indispensable. Modeling and simulation mainly focuses on the forming and quenching and is often performed during tool design [65]. To this purpose, numerical models based on finite element methods (FEM) are employed, that may be used for prediction of the final mechanical properties of the produced parts [58, 65, 108]. Based on these models, detailed investigations of the final blank temperature and the corresponding influencing factors have been conducted [94]. Basically, all simulation approaches for the forming and quenching process make use of a high number of elements to achieve a high accuracy. The resulting computation times (ranging from several minutes to several days), however, render these models unsuitable for online process control.

The simulation, control, and optimization of the heating in an RHF is rarely considered in literature. A simple simulation model for the blank temperature evolution based on radiation and convection has been proposed in [130]. This model can be used to determine optimal furnace process variables offline, but it does not consider the furnace temperature dynamics and cannot be used for online control of the furnace temperatures.

In summary, different modeling approaches for the time-temperature evolution, the mechanical properties, and their combination exist both for the pressing and the furnace process. However, these models are either computationally very expensive or do not consider process dynamics, i.e., cannot be used for online control. There are few approaches using static black box models for process control. In the next section, these approaches are discussed in more detail. Furthermore, to the best of the author's knowledge, there is no model that considers the influence of the part flow (or production line halts) on the part properties.

Process Supervision and Process Control   During the process design phase, the aforementioned models and methods are used to design the desired process route as well as the machinery. The focus is on the robustness to disturbances and variable variations [115]. In almost all cases, these offline optimized tools, machinery, and parameter sets are used independently of the actual disturbances. The result is

an increased variance of the final part properties, especially when the tool wear is increasing or abrupt changes in the production flow occur. To mitigate this problem, in-line measurements of the final part properties and a closed-loop process control are necessary. For the latter very few results exist.

In [75], a process control approach based on a knowledge data base is proposed. Based on experiments, a table-like knowledge data base is created to derive regression models that are used to determine the process variables online. This approach does not consider any dynamics of the process, like the tool temperature evolution. Furthermore, it is not clear how control actions are determined for uncommon measurements, e.g., if errors in the production system occur.

A hierarchical control approach for a hot stamping line based on the "real time control system architecture" of the National Institute of Standards and Technology was developed in [141]. Here, process variables are controlled locally based on the discrete status of the whole production line (errors, pre-heating of furnace, etc.) which is determined by an upper hierarchy layer. The upper layer also coordinates the part flow and task planning. However, neither control nor coordination consider error probabilities or stability and robustness.

Furthermore, adaptive fuzzy control for the heating of functionally graded materials has been proposed [36]. The approach employs offline computed optimal temperature profiles based on FEM simulations and online heating control based on adaptive fuzzy control. The approach is implemented for the induction heating of metal cylinders, but it is claimed that it can be adopted for hot stamping processes. The aim of [36] is to find optimal heating strategies adaptively for disturbances like varying base materials or ambient temperatures. However, the control approach does not provide a guarantee for stability and robustness nor does it consider abrupt changes in the production system directly.

None of the aforementioned approaches makes use of the knowledge about possible errors and its probabilities of occurrence. Furthermore, there is no control approach known to the author that guarantees robustness to faults and disturbances such that operation can be continued (possibly with temporarily degraded performance).

1.2.4. Summary

All in all, the literature review shows that in the context of manufacturing systems, there are no approaches for process control that focus on the part properties and consider the knowledge of the disturbances and abrupt faults of the production units. The existing approaches either consider abrupt disturbances but focus on scheduling and production rate control instead of product properties or consider the control of product properties but do not consider the (stochastic) knowledge of existing disturbances and faults. Even in the context of fault tolerant control, no MPC-based approaches that consider fault probabilities and disturbances are proposed for manufacturing systems.

JMS can be used to model abrupt disturbances (faults) in manufacturing systems

and their influence on continuous quantities, such as the production rate [2, 21]. However, to the best of the author's knowledge, JMS have not been used for modeling the product properties of piece goods. In addition, current robust MPC approaches for JMS are not applicable to large scale systems (cf. Section 5.1).

Concerning the distributed nature of a multistage manufacturing process consisting of different production units, two main modeling frameworks apply. First, the SoV approach addresses the propagation of process and product property variations through the manufacturing stages. However, dynamic processes, like tool temperatures, are not considered in this framework. On the other hand, the field of distributed (model predictive) control considers the control of coupled dynamic systems. But the special sequential structure of manufacturing systems and the resulting variation propagation effects are rarely exploited explicitly in this context. Hence, a combination of concepts from the SoV framework, JMS that model the production units, and distributed MPC (DMPC) proposes a powerful framework for the modeling and control of manufacturing systems that are subject to faults. However, there are no approaches combining these three elements.

This particularly concerns possible applications to mass production systems. For example, there are no approaches for process control of hot stamping production lines that consider the knowledge about both disturbances and production unit faults.

1.3. Aim and Outline of this Dissertation

In summary, the aim of this thesis is to develop a modeling and control framework for multistage manufacturing systems for piece goods affected by disturbances and faults. To this end, the issues described in the introduction and the literature review are addressed. In detail, the following goals are pursued:

- **Modeling**: Modeling of the multistage manufacturing process as a distributed system consisting of coupled JMS, each modeling a production unit. The faults of the production units are modeled by the Markov chains of the JMS. The definition of the local JMS and their couplings is based on the ideas of the SoV framework.

- **Control**: Design of a robust DMPC approach for the resulting model of the manufacturing system. The main focus is on the robustness to both continuous disturbances as well as abrupt faults and strict satisfaction of constraints. The key for applying MPC for JMS to real systems is the efficient formulation of the prediction model and the optimization problem. The resulting calculation times have to be short compared to the cycle times. In particular, for nonlinear models, this is a challenging task. Furthermore, the formulation of constraints making the MPC robust to disturbances is of great importance. Both aspects will be considered in detail throughout this thesis.

- **Application**: Simulation-based application, test, and evaluation of the modeling and control approaches for small illustrative example systems and, more importantly, for a complete hot stamping production line.

These topics are addressed by the following outline.

Outline of this Dissertation

In the following Chapter 2, the general notation is introduced. Definitions and the theoretical background concerning, inter alia, JMS, MPC, and probability theory, is presented.

The modeling framework for multistage manufacturing processes is described in Chapter 3. After a brief description of the overall modeling problem, the modeling of a production unit by a JMS and the definition of all related components is shown in detail. Finally, the modeling of the couplings and the whole manufacturing line is presented. In Chapter 4, the presented modeling framework is applied to a hot stamping line from the Volkswagen plant in Baunatal.

The following chapters are dedicated to the development of a robust MPC approach for a general JMS modeling a single production unit. A thorough review of the literature concerning MPC for JMS and a formal problem definition are presented in Chapter 5. In Chapter 6, the cost and constraint reformulation as well as the basic MPC formulation as a quadratic program is presented for JMLS. The following Chapter 7 is dedicated to the derivation of constraints that guarantee stability, robustness, and recursive feasibility of the MPC. Finally, the extension to nonlinear JMS (Chapter 8) and a brief excursus to estimation of the continuous state (Chapter 9) are presented.

The extension of the derived MPC concepts to a distributed JMS that models the whole manufacturing process is addressed in Chapter 10 and Chapter 11. A presentation of the requirements and a brief introduction to the state of the art in DMPC is given in Chapter 10. The required extensions of the isolated robust MPC to a robust DMPC are explained in detail in Chapter 11.

The control approaches developed here are demonstrated and evaluated with the model of the hot stamping process described in Chapter 4. First, the properties of the developed MPC for a single production unit are investigated in detail, using the model of the roller hearth furnace of the hot stamping line in Chapter 12. An evaluation of the DMPC and the error models can be found in Chapter 13.

The thesis concludes in Chapter 14 by a summary of the presented approaches and a discussion of the results. In addition, open issues and possible future research directions are addressed.

2. Definitions and Preliminaries

In this chapter, the notation as well as the theoretical background and definitions regarding ellipsoids, probability, time discretization, MPC, linear matrix inequalities, and JMS are presented.

2.1. Notation

The transpose of a matrix $M \in \mathbb{R}^{n \times m}$ is denoted by $M^\top$. Transposed entries in a symmetric matrix are abbreviated by a "$\star$". All vectors are column vectors. However, for the sake of notation, vectors introduced in the text by row vectors without transposing each element. Identity matrices, zero matrices, and one matrices of appropriate size are denoted by I, $\mathbf{0}$, and $\mathbf{1}$. The dimensions are indicated by additional indices, if not apparent from the context. Let the norm and the weighted norm of a vector be given by: $\|x\|^2 = x^\top x$ and $\|x\|_M^2 = x^\top M x$. The smallest and the largest eigenvalue of a matrix M are given by $\lambda_{\min}(M)$ and $\lambda_{\max}(M)$, respectively. Open intervals are indicated by parenthesis, e.g., (c_1, c_2), while closed intervals are denoted by brackets, e.g., $[c_1, c_2]$.

The set of integers from n_1 to n_2 is denoted by $\mathbb{N}_{n_1:n_2}$. The special case for $n_1 = 1$ is abbreviated by $\mathbb{N}_{n_2}$. The ":" is also used to indicate intervals of indices as well as the indexed quantities (similar to the MATLAB notation). Sets of symmetric, positive definite, and positive semi-definite $n \times n$-matrices are indicated by $\mathbb{S}_n$, $\mathbb{S}_n^{>0}$, and $\mathbb{S}_n^{\geq 0}$. Finally, a set of n matrices ψ_i is abbreviated by the corresponding symbol without indices: $\psi = \{\psi_1, \ldots, \psi_n\}$.

This thesis considers a discrete-time setting. The sampling instants $t_k = t_0 + k\,\Delta t$, with $t_0 \in \mathbb{R}^{\geq 0}$, sampling interval $\Delta t \in \mathbb{R}^{>0}$, and $k \in \mathbb{N}$ are indicated by the time index k in brackets, e.g., $x[k] := x(t_k)$. If continuous time is used, it is denoted by t in parenthesis, e.g., $x(t)$. Quantities predicted for j time steps into the future at time k are denoted by double brackets instead of the conventional notation, i.e., $x[\![j]\!] := x[k+j|k]$. The same notation is employed for matrices used for prediction at time k. Symbols, aggregating vectors or matrices for the whole prediction horizon N, are denoted by the corresponding bold faced symbols, such as $\boldsymbol{x}[k] := \begin{bmatrix} x[\![0]\!] & \cdots & x[\![N-1]\!] \end{bmatrix}$.

For two compact sets $\mathbb{X}_1, \mathbb{X}_2 \subset \mathbb{R}^n$ the Minkowski sum and difference (also known as Pontryagin difference) are defined as follows, cf. [67]:

$$\mathbb{X}_1 \oplus \mathbb{X}_2 := \{x_1 + x_2 \,|\, x_1 \in \mathbb{X}_1, x_2 \in \mathbb{X}_2\}, \tag{2.1}$$

$$\mathbb{X}_1 \ominus \mathbb{X}_2 := \{x \in \mathbb{R}^n \,|\, \mathbb{X}_2 \subseteq \mathbb{X}_1 : x + \mathbb{X}_2 \subseteq \mathbb{X}_1\}. \tag{2.2}$$

The interval operator $\lfloor \cdot \rceil$ for a vector b with values in a bounded set $\mathbb{R}' \subset \mathbb{R}^n$ returns an over-approximation of $\mathbb{R}'$ in form of an n-dimensional hyperbox:

$$\lfloor b \rceil := \left[\left[\min_{b \in \mathbb{R}'} b_1, \ \max_{b \in \mathbb{R}'} b_1 \right] \ \cdots \ \left[\min_{b \in \mathbb{R}'} b_n, \ \max_{b \in \mathbb{R}'} b_n \right] \right]^\top . \tag{2.3}$$

The more specific notation is introduced in the following sections.

2.2. Ellipsoids

This section mainly bases on results from [74]. General ellipsoids on $\mathbb{R}^n$ are denoted by $\mathcal{E}$ and defined as follows:

Definition 2.1. *An ellipsoid $\mathcal{E}$ with center $c \in \mathbb{R}^n$ and shape matrix $\Lambda \in \mathbb{S}_n^{>0}$ is given by:*

$$\mathcal{E}(c, \Lambda) = \left\{ x \in \mathbb{R}^n \mid \|x - c\|_{\Lambda^{-1}}^2 = (x - c)^\top \Lambda^{-1} (x - c) \leq 1 \right\} . \tag{2.4}$$

If $c = \mathbf{0}$, the shorter notation $\mathcal{E}(\Lambda)$ is used.

For an affine transformation of the ellipsoid $x \in \mathcal{E}(c, \Lambda)$, it holds [74]:

$$Mx + b \in \mathcal{E}\left(Mc + b, M\Lambda M^\top\right) . \tag{2.5}$$

Support Function: The maximum expansion of an ellipsoid $\mathcal{E}$ in direction $b \in \mathbb{R}^n$ is given by the support function $\rho(b \mid \mathcal{E}(c, \Lambda))$, which can be calculated by [74]:

$$\rho(b \mid \mathcal{E}(c, \Lambda)) = b^\top c + \sqrt{b^\top \Lambda b} = b^\top c + \|b\|_\Lambda . \tag{2.6}$$

Minkowski Sum and Difference: The Minkowski sum and difference of two ellipsoids $\mathcal{E}_1 \oplus \mathcal{E}_2$ and $\mathcal{E}_1 \ominus \mathcal{E}_2$ are defined as in (2.1) and (2.2). The effect of both operators is shown in Fig. 2.1. If both ellipsoids are centered at the origin, i.e., $c_1 = c_2 = \mathbf{0}$, both the Minkowski sum and difference can be interpreted geometrically as follows: The Minkowski sum is the area or volume that is covered by all points of the ellipsoid $\mathcal{E}_2$ when moved with its center into all points of the ellipsoid $\mathcal{E}_1$. The Minkowski difference is the area or volume that remains of $\mathcal{E}_1$ if all points are removed, which are covered by an interior point of the ellipsoid $\mathcal{E}_2$ that is moved along all points of the surface of ellipsoid $\mathcal{E}_2$.

In general, neither the Minkowski sum nor difference of two ellipsoids $\mathcal{E}(c_1, \Lambda_1)$ and $\mathcal{E}(c_2, \Lambda_2)$ is an ellipsoid, but a general nonlinear set. For these sets, ellipsoidal inner and outer approximations are often employed, where the shape matrix takes certain values in a family of matrices defined by [74]:

$$\Lambda(\alpha) := \left(1 + \alpha^{-1}\right) \Lambda_1 + (1 + \alpha) \Lambda_2 . \tag{2.7}$$

Employing this family of shape matrices, the following results for the inner and outer approximations hold:
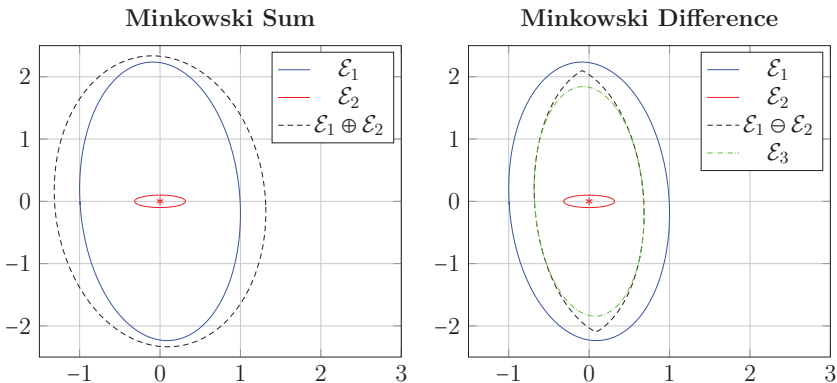
Figure 2.1.: Minkowski sum (left) and difference (right) of the ellipsoids $\mathcal{E}_1$ and $\mathcal{E}_2$. The ellipsoid $\mathcal{E}_3 \subset \mathcal{E}_1 \ominus \mathcal{E}_2$ shows one ellipsoidal inner approximation of the Minkowski difference.

Lemma 2.1. *([74]) An outer approximation of the Minkowski sum $\mathcal{E}_1 \oplus \mathcal{E}_2$ is given by the ellipsoid:*

$$\mathcal{E}(c_1 + c_2, \Lambda(\alpha)) \supseteq \mathcal{E}_1 \oplus \mathcal{E}_2 \tag{2.8}$$

for all $\alpha > 0$ with $\Lambda(\alpha)$ according to (2.7). The unique outer approximation of minimum volume is obtained, if α is selected to be the root of the equation

$$\sum_{m=1}^{n_{\mathrm{x}}} \frac{1}{\lambda_m + \alpha} - \frac{n_{\mathrm{x}}}{\alpha(1 + \alpha)} = 0, \tag{2.9}$$

that falls in the interval $[\lambda_{\min}^{1/2}(\Lambda_1, \Lambda_2), \lambda_{\max}^{1/2}(\Lambda_1, \Lambda_2)]$. Here, λ_i are the generalized eigenvalues of Λ_1 and Λ_2.

Proof. The proof is presented in [74] (see proofs of Lemmata 2.2.1 and 2.5.3). $\qquad\square$

Corollary 2.1. *An outer approximation of $\mathcal{E}_1 \oplus \mathcal{E}_2$ is given by the ellipsoid:*

$$\mathcal{E}(c_1 + c_2, \Lambda(\varrho)) \supseteq \mathcal{E}_1 \oplus \mathcal{E}_2 \tag{2.10}$$

with

$$\Lambda(\varrho) := (1 - \varrho)^{-1} \Lambda_1 + \varrho^{-1} \Lambda_2 \tag{2.11}$$

for all $\varrho \in (0, 1)$. This formulation follows from Lemma 2.1 by substituting $\alpha = \varrho^{-1} - 1$.

Lemma 2.2. *([74]) An inner approximation of the Minkowski difference $\mathcal{E}_1 \ominus \mathcal{E}_2$ is given by the ellipsoid:*

$$\mathcal{E}(c_1 - c_2, \Lambda(-\alpha)) \subseteq \mathcal{E}_1 \ominus \mathcal{E}_2 \tag{2.12}$$

for all $\alpha \in (1, \lambda_{\min}(\Lambda_1, \Lambda_2))$. The unique inner approximation of maximum volume is obtained if α is selected to be the root of equation (2.9) that falls in the interval $[\lambda_{\min}^{1/2}(\Lambda_1, \Lambda_2), \lambda_{\max}^{1/2}(\Lambda_1, \Lambda_2)] \cap (1, \lambda_{\min}(\Lambda_1, \Lambda_2))$.

Proof. The proof is presented in [74] (see proofs of Lemmata 2.2.2 and 2.5.3). □

2.3. Theoretical Background of Probability Theory

In this section, definitions and results concerning probability theory used throughout this thesis are presented. Let $\Pr(\cdot)$ and $\mathrm{E}(\cdot)$ define the probability and expectancy operator, respectively. The expectancy is also indicated by a bar over the corresponding random variable, e.g., $\bar{x}[k] = \mathrm{E}(x[k])$.

Moments: The first and second central moment, i.e., the expected value and the co-variance matrix, of a random variable x taking values from a discrete finite set are defined by:

Definition 2.2. *The expected value of a random variable x is defined by [57]:*

$$\bar{x} := \sum_{i \in \mathcal{I}} x_i \cdot \Pr(x = x_i), \tag{2.13}$$

where $\mathcal{I}$ denotes the index set of all possible values of the random variable x. If the set of outcomes x is continuous, the sum is replaced by the integral over this set.

Definition 2.3. *The co-variance matrix of a random variable x is defined by [117]:*

$$\mathrm{Var}(x) := \mathrm{E}\left((x - \bar{x})(x - \bar{x})^\top\right) = \mathrm{E}\left(x\,x^\top\right) - \bar{x}\,\bar{x}^\top \tag{2.14}$$

while the co-variance matrix of two random variables x_1 and x_2 is given by [117]:

$$\mathrm{Cov}(x_1, x_2) := \mathrm{E}\left((x_1 - \bar{x}_1)(x_2 - \bar{x}_2)^\top\right). \tag{2.15}$$

This notation is also used for matrix valued random variables.

For two random variables x_1 and x_2, the following basic properties hold:

- The expectancy of a sum of several summands equals the sum of the expectancies of the summands (cf. [117]):

$$\mathrm{E}(M_1 x_1 + M_2 x_2 + b) = M_1 \bar{x}_1 + M_2 \bar{x}_2 + b. \tag{2.16}$$

- From Equation (2.15), it follows that the expectancy of a product of two scalars is given by:

$$\mathrm{E}(x_1 x_2) = \bar{x}_1 \bar{x}_2 + \mathrm{Cov}(x_1, x_2). \tag{2.17}$$

  Hence, the expectancy of a product is given by the product of the expectancies of the factors, if the factors are stochastically independent.

Distributions: Throughout this thesis, the multi-variate normal distribution and the χ^2-distribution are used.

Definition 2.4. *[72] A multi-variate normal distribution with mean $c \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{S}_n^{>0}$ is denoted by $\mathcal{N}(c, \Sigma)$. Its probability density function (PDF) is given by:*

$$\Pr(x) = \frac{1}{\sqrt{(2\pi)^n \det(\Sigma)}} e^{-\frac{1}{2}(x-c)^\top \Sigma^{-1}(x-c)}. \tag{2.18}$$

A plot of a two-dimensional normal distribution is shown in Fig. 2.2 on page 24. From Equation (2.18) it follows that the absolute value of the random variable may take very high values but with very low probabilities.

Definition 2.5. *(cf. [1]) For an n-dimensional random variable $x \sim \mathcal{N}(0, I_n)$, the distribution of the square $x^\top x$ is denoted χ^2-distribution. Its PDF is given by:*

$$f_{\chi^2}(c, n) = \Pr(x^\top x = c) = \frac{c^{\frac{n}{2}-1} e^{-\frac{c}{2}}}{\Gamma\left(\frac{n}{2}\right) 2^{\frac{n}{2}}} \tag{2.19}$$

and the cumulative density function is given by:

$$F_{\chi^2}(c_1, n) = \Pr(x^\top x \le c_1) = \int_0^{c_1} f_{\chi^2}(c, n) \, \mathrm{d}c = \frac{\gamma\left(\frac{n}{2}, \frac{c_1}{n}\right)}{\Gamma\left(\frac{n}{2}\right)}, \tag{2.20}$$

where $\Gamma(n)$ and $\gamma(n, c)$ are the gamma function and incomplete gamma function[1], respectively.

Determination of Confidence Ellipsoids of Multi-Variate Normal Distributions[2]:
The surfaces of equal density of a normal distribution $\mathcal{N}(c, \Sigma)$ is given by [72]:

$$x \in \mathbb{R}^n \; : \; (x - c)^\top \Sigma^{-1}(x - c) = \delta. \tag{2.21}$$

In Equation (2.21), δ is distributed as χ^2. Let the confidence level of all points contained in this ellipsoid be $\beta \in (0, 1)$:

$$\Pr\left((x - c)^\top \Sigma^{-1}(x - c) \le \delta\right) = F_{\chi^2}(\delta, n) = \beta. \tag{2.22}$$

The confidence ellipsoid for a predefined confidence level β can be derived from (2.22). Therefore, the value of δ is determined by the inverse of $F_{\chi^2}(\delta, n)$:

$$\delta := F_{\chi^2}^{-1}(\beta, n). \tag{2.23}$$

Bringing the ellipsoid defined in (2.22) into the form defined in Definition 2.1, gives the confidence ellipsoid:

$$\mathcal{E}_\beta = \mathcal{E}\left(c, F_{\chi^2}^{-1}(\beta, n) \Sigma\right). \tag{2.24}$$

For a visualization of the three-sigma confidence ellipsoid of a two-dimensional normal distribution, see Fig. 2.2 at the following page.

---

[1]See [1] for more details on the gamma functions.
[2]This section cites results from [10].
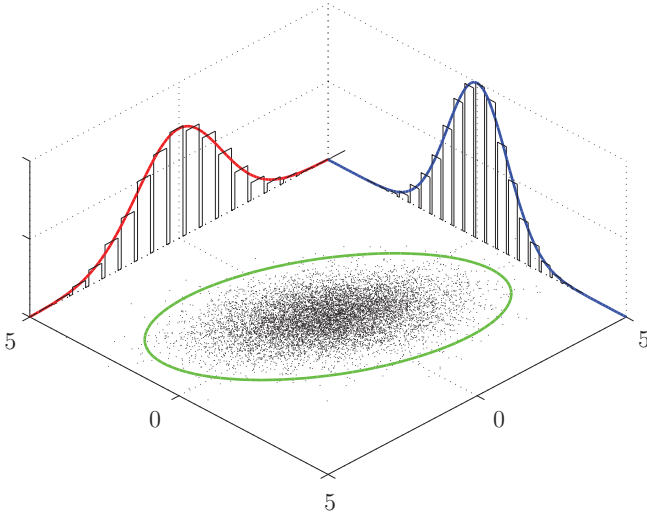
Figure 2.2.: Density of a two-dimensional normal distribution with its three-sigma ellipsoid [145].

2.4. Time Discretization

Throughout this thesis, a discrete-time setting is considered. Since many models of real systems are continuous-time models, the corresponding discrete-time systems have to be determined by a time discretization. Therefore, some approaches for the time discretization of continuous-time systems are presented in this section.

Assumption 2.1. *It is assumed that the inputs $u(t)$ and disturbances $w(t)$ are constant during each sampling interval.*

This assumption is justified when looking at production systems, where the inputs are process variables that are constant for each production cycle and the disturbances, such as ambient temperatures, are changing slowly compared to the sampling time.

Linear Systems: The zero-order-hold (ZOH) time discretization with sampling time $\Delta t \in \mathbb{R}^{>0}$ of the continuous-time linear system

$$\dot{x}(t) = A_{\mathrm{c}}x(t) + B_{\mathrm{c}}u(t) + G_{\mathrm{c}}w(t) \tag{2.25}$$

returns a discrete-time system

$$x[k+1] = Ax[k] + Bu[k] + Gw[k] \tag{2.26}$$

which is an exact representation of the continuous-time system. The matrices of the discrete-time system (2.26) are given by (cf. [7]):

$$A = e^{A_c \Delta t}, \qquad B = \int_0^{\Delta t} e^{A_c \tau} B_c \, d\tau, \qquad G = \int_0^{\Delta t} e^{A_c \tau} G_c \, d\tau. \qquad (2.27)$$

Nonlinear Systems: To discretize a general nonlinear system over time, the following differential equation has to be solved for $t \in [t_0 + k\Delta t, t_0 + (k+1)\Delta t]$:

$$\dot{x}(t) = f_c\big(x(t), u(t), w(t), t\big). \qquad (2.28)$$

This results in a discrete-time system of the following form:

$$x[k+1] = f\big(x[k], u[k], w[k], k\big) = x(t_0 + (k+1)\Delta t). \qquad (2.29)$$

If an analytical solution exists, it should be used as the discrete-time system. A special case are nonlinear systems that are affine in $x(t)$:

$$\dot{x}(t) = A_c\big(u(t), w(t)\big) x(t) + g_c\big(u(t), w(t)\big), \qquad (2.30)$$

where $A_c(\cdot, \cdot)$ and $g_c(\cdot, \cdot)$ are arbitrary nonlinear matrix-valued and vector-valued functions of appropriate dimensions. Systems of this form arise, e.g., from thermal models, where the thermal conductivity depends on the inputs (cf. Section 4.2.4).

The matrix $A_c(u(t), w(t))$ and the vector $g_c(u(t), w(t))$ are constant for each time step due to Assumption 2.1. Hence, exact time discretization can be applied:

$$x[k+1] = e^{A_c(u[k], w[k]) \Delta t} x[k] + \int_0^{\Delta t} e^{A_c(u[k], w[k]) \tau} \, d\tau \cdot g_c\big(u[k], w[k]\big). \qquad (2.31)$$

An analytical calculation of the matrix exponentials and integrals is only possible for small dimensions n_x. For most applications, this is not the case and approximations in terms of truncated power series and numeric integration have to be exploited.

In general, for nonlinear systems (2.28), no analytical solution exists and approximations have to be applied. Some approaches are listed below:

- **Linearization and ZOH:** The continuous-time system can be linearized first and subsequently discretized over time by ZOH-discretization [139].

- **Euler discretization:** A simple approach is to assume that the right hand side of the differential equation (2.28) is constant over the sampling interval, and the discrete-time system is given by (cf. [68]):

$$x[k+1] = x[k] + \Delta t \, f_c\big(x[k], u[k], w[k], t_0 + k\,\Delta t\big). \qquad (2.32)$$

  To reduce the discretization error, the sampling interval can be divided into sub-intervals for which the Euler discretization is applied recursively [68].

- **Runge-Kutta methods:** In order to achieve a better approximation of the system dynamics, the well known Runge-Kutta methods or other multi-step approaches can be used (see, e.g., [49]).

2.5. Model Predictive Control

The idea of MPC [71, 88, 93, 107] is to use a model of the plant to predict the system behavior $\boldsymbol{x}^\top[1] := \begin{bmatrix} x^\top[1] \cdots x^\top[N] \end{bmatrix}$ for a certain control horizon N as a function of the input trajectory $\boldsymbol{u}^\top[k] := \begin{bmatrix} u^\top[0] \cdots u^\top[N-1] \end{bmatrix}$. At each sampling instant k, the optimal input trajectory $\boldsymbol{u}'[k]$ is determined by solving an optimization problem subject to state and input constraints [88, 93]. A standard MPC formulation for an LTI system:

$$x[k+1] = A\,x[k] + B\,u[k] \tag{2.33}$$

with state and input constraints given by the sets $\mathbb{X}$ and $\mathbb{U}$ as well as references for states and inputs defined by $\boldsymbol{x}_\mathrm{r}[k]$ and $\boldsymbol{u}_\mathrm{r}[k]$ is given by (cf. [63, 88]):

$$\boldsymbol{u}'[k] = \arg\min_{\boldsymbol{u}[k]} \quad \|\boldsymbol{x}[1] - \boldsymbol{x}_\mathrm{r}[1]\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}[k] - \boldsymbol{u}_\mathrm{r}[k]\|_{\boldsymbol{R}}^2 \tag{2.34a}$$

$$\text{s.\,t.} \quad \boldsymbol{x}[1] \in \mathbb{X}, \quad \boldsymbol{u}[k] \in \mathbb{U} \tag{2.34b}$$

where $\boldsymbol{Q} \in \mathbb{S}_{n_\mathrm{x}\cdot N}^{\geq 0}$, $\boldsymbol{R} \in \mathbb{S}_{n_\mathrm{u}\cdot N}^{> 0}$. The state trajectory $\boldsymbol{x}[1]$ is given by:

$$\boldsymbol{x}[1] = \begin{bmatrix} x[1] \\ x[2] \\ \vdots \\ x[N] \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x[k] + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \ddots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \boldsymbol{u}[k]. \tag{2.35}$$

This problem formulation is sometimes referred to as *condensed MPC formulation* [63]. For long prediction horizons and unstable systems, it may result in numerical problems [63]. The numerical problems may be mitigated by a different formulation of the optimization problem (referred to as *sparse MPC formulation*) [63, 93]:

$$\boldsymbol{u}'[k] = \arg\min_{\boldsymbol{u}[k],\,\boldsymbol{x}[1]} \sum_{j=1}^{N} \left(\|x[j] - x_\mathrm{r}[j]\|_Q^2 + \|u[j-1] - u_\mathrm{r}[j-1]\|_R^2 \right) \tag{2.36a}$$

$$\text{s.\,t.} \quad x[j] = A\,x[j-1] + B\,u[j-1] \qquad\qquad \forall j \in \mathbb{N}_N \quad (2.36\mathrm{b})$$

$$x[0] = x[k] \tag{2.36c}$$

$$x[j] \in \mathbb{X} \qquad\qquad \forall j \in \mathbb{N}_N \quad (2.36\mathrm{d})$$

$$u[j-1] \in \mathbb{U} \qquad\qquad \forall j \in \mathbb{N}_N, \quad (2.36\mathrm{e})$$

with $Q \in \mathbb{S}_{n_\mathrm{x}}^{\geq 0}$ and $R \in \mathbb{S}_{n_\mathrm{u}}^{> 0}$. Once $\boldsymbol{u}'[k]$ is determined by either of the formulations, the first element $u'[0]$ is applied to the system and the procedure is repeated at the time $k+1$ employing the measured value of $x[k+1]$.

Stability and Recursive Feasibility

To guarantee stability and recursive feasibility[3] of the MPC, different approaches have been proposed. Approaches for deterministic systems without disturbances

---

[3]Recursive feasibility is the property that the MPC optimization problem is feasible for all times $k+j$, if the optimization problem is feasible in time k.

are presented in this section. Robust approaches that consider disturbances are presented in the following section.

First of all, an infinite prediction horizon N guarantees stability and recursive feasibility. In general, it is impractical to formulate and solve the optimization problem for an infinite horizon, but it can be shown that stability can still be guaranteed if the prediction horizon is sufficiently large [88, 93]. The necessary length depends on the system dynamics.

Most approaches consider additional terminal constraints of the form $x[\![N]\!] \in \mathbb{X}_f$ and a special selection of the terminal costs $\|x[\![N]\!]\|_{Q_f}^2$ [93, 110]. In general, the terminal set $\mathbb{X}_f$ is selected to be a control invariant set and Q_f defines a control Lyapunov function (for details see, e.g., [110]). For linear systems, an infinite horizon cost function can be realized, if the terminal cost is defined by the solution of a Lyapunov or Riccati equation [88, 110].

For stability analysis of MPC, the state costs are often used as a Lyapunov function, for which a cost decrease is shown [88, 110]. This decrease can be guaranteed indirectly by the approaches presented above, or directly by additional constraints requiring that the state $x[k+1]$ results in a lower value of the cost (Lyapunov) function [67].

Uncertain Systems and Robustness

In general, the system model is not an exact representation of the real plant. This results in a certain level of uncertainty of the prediction. To guarantee stability, constraint satisfaction, and recursive feasibility, the MPC design has to consider the uncertainties. Robust MPC approaches can be divided into approaches that consider additive disturbances, multiplicative / parameter uncertainties, or a mixture of both [88, 110]. For linear systems, additive disturbances are modeled by the disturbance input $w[k] \in \mathbb{W}$, where $\mathbb{W}$ is a compact set (cf. [71, 88, 110]), and the dynamics are:

$$x[k+1] = A\,x[k] + B\,u[k] + G\,w[k]. \tag{2.37}$$

Parametric uncertainties are often modeled by system matrices $A[k]$, $B[k]$, and $G[k]$ which are unknown and may take values from a bounded set. This set is commonly defined by matrix polytopes $\mathbb{A}, \mathbb{B}$, and $\mathbb{G}$, which are defined by the convex hull of known vertices $A_{[l]}, B_{[l]}$, and $G_{[l]}$ with $l \in \mathbb{N}$. For details, see [71, 88, 110].

Due to the disturbances, the state costs cannot be predicted exactly. In addition to the trivial approach to neglect the disturbances and formulate the costs for the nominal system [93, 110], two formulations are proposed:

- The worst case costs for all possible disturbances are considered, resulting in a min-max optimization problem [93, 110].

- In the context of stochastic MPC [71, 93], the expected value of the cost function is considered instead of the deterministic formulation.

While the control input sequence $\boldsymbol{u}'[k]$ of the open-loop optimization problems is the optimal control law for the deterministic system (2.33), this is not the case, if disturbances are present. In this case, the optimal control law results from the optimization over control policies $u[\![j]\!] = K(x[k+j])$ instead of the optimization over control actions $u[\![j]\!]$ directly, in order to account for the disturbances $w[k+l]$ with $0 \leq l < j$. This optimization can be solved by dynamic programming. However, in general the computational effort prohibits an application to real systems. Hence, in most approaches, open-loop predictions or a control parametrization of the form $u[\![j]\!] = K[\![j]\!]\, x[k+j] + v[\![j]\!]$ are employed, where $v[\![j]\!]$ is the new decision variable of the MPC. Variations of this type of closed-loop prediction are employed in most robust MPC approaches [88, 93, 110].

For both open- and closed-loop predictions as well as the different cost functions, various robust MPC approaches have been proposed that guarantee constraint satisfaction, convergence to a neighborhood of the reference, and recursive feasibility (see, e.g, [88, 93, 110]). The remainder of this subsection aims at sketching the main ideas of robust MPC rather than describing robust MPC approaches in detail.

First of all, the MPC can be formulated ignoring all disturbances. In this case, it has to be checked whether the convergence rate is large enough to guarantee robust stability [93, 110]. However, this approach may fail for larger disturbance sets and constraints may be violated. Hence most approaches to robust MPC consider the disturbances directly in the MPC formulation.

An approach to consider all possible disturbance realizations is *tube-based MPC*[4] [71, 93, 110]: The uncertain dynamics are separated into a nominal dynamics and an error dynamics, such that $x[k] = x_{\mathrm{n}}[k] + e[k]$. Employing a control law of the form $u[k] = K(x[k] - x_{\mathrm{n}}[k]) + v[k]$, the closed-loop dynamics of both parts are $x_{\mathrm{n}}[k+1] = (A+BK)\, x_{\mathrm{n}}[k] + B\, v[k]$ and $e[k+1] = (A+BK)\, e[k] + G\, w[k]$, respectively. In most cases, the feedback gain K is determined offline, and the input for the nominal system $v[k]$ is the optimization variable of the MPC. Employing the error dynamics, the error sets $\mathbb{E}[\![j]\!]$ containing the prediction errors $e[\![j]\!]$ can be determined offline by $\mathbb{E}[\![j]\!] = \bigoplus_{l=0}^{j-1}(A+BK)^l \mathbb{W}$. Centering these sets at the states $x_{\mathrm{n}}[\![j]\!]$ of the nominal system, a tube[5] results that contains the future states for all possible disturbance realizations, i.e., $x[\![j]\!] \in x_{\mathrm{n}}[\![j]\!] \oplus \mathbb{E}[\![j]\!]$. Finally, the tube-based MPC is formulated using the nominal dynamics and tightened constraints, requiring that $x_{\mathrm{n}}[\![j]\!] \in \mathbb{X} \ominus \mathbb{E}[\![j]\!]$ and $v[\![j]\!] \in \mathbb{U} \ominus K\,\mathbb{E}[\![j]\!]$ holds for all $j \in \mathbb{N}_N$. The original input and state constraints are satisfied for all possible disturbance realizations due to the definition of smaller input and state sets by the Minkowski difference with the error sets. The process of reducing the constraint sets $\mathbb{X}$ and $\mathbb{U}$ in order to guarantee robust constraint satisfaction is referred to as *constraint tightening*. Asymptotic stability of the nominal system and recursive feasibility can be guaranteed by appropriate terminal costs and the

---

[4]Here, the basic ideas are shown for additive disturbances and linear systems only. The approach can also be adopted to parametric uncertainties and nonlinear systems [93, 110].

[5]A tube is a sequence of sets.

utilization of *robust control invariant sets* (RCIS) as terminal sets $\mathbb{X}_f$.

Another group of approaches determines feedback controllers $u[k] = Kx[k]$ that satisfy all input and state constraints, robustly stabilize the system, and keep the state in a RCIS (often in the form of ellipsoids). In this case, the RCIS is not only used to constrain the state at the end of the prediction horizon but at all times. The controllers and RCIS can be determined by solving semi-definite programs. For the MPC setup, basically two approaches are used: Either the controllers and RCIS are determined online at each time step and $u[k]$ is computed using the control law [70, 88], or the controllers and RCIS can be determined offline and only the RCIS are used online as additional state constraints [67].

Finally, scenario based approaches (cf. [17, 93]) formulate the whole optimization problem (constraints and costs) just for a set of randomly generated disturbance realizations and not for all possible disturbance realizations. This results in a simple problem formulation but does not guarantee robustness to all disturbance realizations. However, depending on the number of scenarios, a certain probability of constraint satisfaction can be guaranteed [93].

2.6. Linear Matrix Inequalities

Linear matrix inequalities (LMIs) are widely used to define convex sets and constraints in control theory applications. This section presents a brief overview of results concerning LMIs.

Definition 2.6. *[23] A strict LMI can be transformed into the form*

$$M(\alpha) = M_0 + \sum_{m=1}^{n_\alpha} \alpha_m M_m > 0, \tag{2.38}$$

with a vector of variables $\alpha \in \mathbb{R}^{n_\alpha}$ and matrices $M_m \in \mathbb{S}_n, m \in \mathbb{N}_{n_\alpha}^0$. A non-strict LMI can be stated in the same way, requiring that $M(\alpha) \geq 0$ holds.

Remark 2.1. *The LMI* (2.38) *defines a convex constraint on the variable α [23].*

To ease the notation, LMIs are not written in this sum form depending on α in the remainder of this thesis. Here, $M(\alpha)$ is replaced by the matrix variable $\mathcal{M}$. Variable matrices are denoted by calligraphic symbols.

Matrix Equality Reformulations

The following results can be used to transform general matrix inequalities into LMIs.

Lemma 2.3. *(Schur Complement) Let $\mathcal{M}_{1,1}$ and $\mathcal{M}_{2,2}$ be symmetric matrices and let $\mathcal{M}_{1,2}$ be of appropriate dimensions. Then, the condition*

$$\begin{bmatrix} \mathcal{M}_{1,1} & \mathcal{M}_{1,2} \\ \mathcal{M}_{1,2}^\top & \mathcal{M}_{2,2} \end{bmatrix} \geq 0 \tag{2.39}$$

is equivalent to:

$$\mathcal{M}_{1,1} - \mathcal{M}_{1,2}\,\mathcal{M}_{2,2}^{-1}\,\mathcal{M}_{1,2}^{\top} \geq 0, \qquad \mathcal{M}_{2,2} > 0. \tag{2.40}$$

The result holds also for strict LMIs if $\mathcal{M}_{1,1} - \mathcal{M}_{1,2}\,\mathcal{M}_{2,2}^{-1}\,\mathcal{M}_{1,2}^{\top} > 0$ holds.

Proof. See [23]. □

Lemma 2.4. *A concurrency transformation retains the definite properties. Hence, for any transformation matrix $\mathcal{T} \in \mathbb{S}_n^{>0}$ and $\mathcal{M} \in \mathbb{S}_n$, it holds [24]:*

$$\mathcal{M} > 0 \;\;\Leftrightarrow\;\; \mathcal{T}^{\top}\mathcal{M}\mathcal{T} > 0, \tag{2.41}$$

$$\mathcal{M} \geq 0 \;\;\Leftrightarrow\;\; \mathcal{T}^{\top}\mathcal{M}\mathcal{T} \geq 0. \tag{2.42}$$

Proof. Let $x, \tilde{x} \in \mathbb{R}^n$ with $x = \mathcal{T}\tilde{x}$. Then, $\tilde{x} = \mathcal{T}^{-1}x \neq \mathbf{0}$ holds for all $x \neq \mathbf{0}$, since $\mathcal{T} > 0$, and it follows that:

$$\mathcal{M} > 0 \;\;\Leftrightarrow\;\; x^{\top}\mathcal{M}x = \tilde{x}^{\top}\mathcal{T}^{\top}\mathcal{M}\mathcal{T}\tilde{x} > 0 \;\;\Leftrightarrow\;\; \mathcal{T}^{\top}\mathcal{M}\mathcal{T} > 0 \tag{2.43}$$

for all $x \neq \mathbf{0}$. The same reasoning holds for the non-strict case. □

Lemma 2.5. *Let $\mathcal{M} \in \mathbb{S}_n^{>0}$ and $\mathcal{G} \in \mathbb{R}^{n \times n}$. Then the following inequality holds:*

$$\mathcal{G}\,\mathcal{M}^{-1}\mathcal{G}^{\top} \geq \mathcal{G}^{\top} + \mathcal{G} - \mathcal{M}. \tag{2.44}$$

Proof. See [102]. □

2.7. Jump Markov Systems

Jump Markov systems are a class of nonlinear stochastic switched systems, where the switching is governed by the state of a Markov chain $\mathcal{M}$, cf. [120].

Definition 2.7. *Let a discrete-time JMS be defined as follows:*

$$\mathcal{S} := \begin{cases} x[k+1] &= f_{\theta_k}\big(x[k], u[k], w[k], k\big) \\ y[k] &= g_{\theta_k}\big(x[k], u[k], w[k], k\big) \\ \mathcal{M} &= \big(\Theta, P[k], \mu[k]\big) \end{cases}, \tag{2.45}$$

where $x[k] \in \mathbb{R}^{n_{\mathrm{x}}}$ describes the system states, $u[k] \in \mathbb{R}^{n_{\mathrm{u}}}$ the inputs, $w[k] \in \mathbb{R}^{n_{\mathrm{w}}}$ the disturbance inputs, and $y[k] \in \mathbb{R}^{n_y}$ the outputs. The corresponding spaces are denoted by $\mathbb{X}, \mathbb{U}, \mathbb{W}$, and $\mathbb{Y}$. The differentiable functions $f_{\theta_k}(\cdot)$ and $g_{\theta_k}(\cdot)$ describe the dependencies of the state $x[k+1]$ and output $y[k]$ on the current states, inputs, and disturbances for each Markov state θ_k. At the following page, the components of the Markov chain (the set of Markov states Θ, the transition probability matrix $P[k]$, and the probability distribution $\mu[k]$) are described in detail.

Definition 2.8. *A corresponding jump Markov linear system (JMLS) is defined as follows, cf. [39]:*

$$\mathcal{S}_{\mathrm{l}} := \begin{cases} x[k+1] & = & A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[k] + G_{\theta_k}[k]\,w[k] \\ y[k] & = & C_{\theta_k}[k]\,x[k] + D_{\theta_k}[k]\,u[k] + F_{\theta_k}[k]\,w[k] \quad , \\ \mathcal{M} & = & (\Theta, P[k], \mu[k]) \end{cases} \tag{2.46}$$

where the system matrices $A_{\theta_k}[k]$ to $G_{\theta_k}[k]$ are of appropriate dimension.

Assumption 2.2. *The disturbance set $\mathbb{W}$ is compact. The mean $\bar{w}[k]$ and covariance Σ_{w} of the disturbance $w[k]$ are known. Furthermore, the disturbance is independent of the state $x[k]$, the input $u[k]$, the output $y[k]$, the Markov state θ_k, and the transition probabilities $P[k]$.*

The boundedness and the knowledge of the mean and covariance are well motivated for production systems. In most cases, historical data of the corresponding values is available for these systems. The independence of $x[k]$, $u[k]$, $y[k]$, and θ_k is assumed to simplify the derivations presented in this thesis, but it is not necessary for the applicability of the proposed approaches. In addition, in most cases, an dependency of the disturbances on the Markov state can be modeled by the system matrix G_{θ_k}, and $w[k]$ can be defined independent of θ_k.

Where necessary, the disturbance is split into its mean and a zero mean part:

$$w[k] = \bar{w}[k] + \tilde{w}[k]. \tag{2.47}$$

Markov Chain

The Markov chain $\mathcal{M}$ with Markov state θ_k is defined by the following triple [27]:

- The set of Markov states $\Theta \equiv \mathbb{N}_{n_\theta}$.

- The transition probabilities $p_{i,m}[k] := \Pr(\theta_{k+1} = m \,|\, \theta_k = i), (i,m) \in \Theta^2$ define the elements of the transition probability matrix $P[k]$.

- The probability distribution $\mu[k] \in [0,1]^{n_\theta}$, where $\mu_i[k] := \Pr(\theta_k = i)$.

If the transition probabilities are time-invariant, the Markov chain is called homogenous. Otherwise, it is called inhomogeneous [27].

Lemma 2.6. *The probability distribution can be predicted as follows:*

$$\mu[\![j]\!] = \prod_{l=0}^{j-1} P^\top[k+l] \cdot \mu[k]. \tag{2.48}$$

For a homogenous Markov chain, this formulation reduces to (cf. [39]):

$$\mu[\![j]\!] = \left(P^\top\right)^j \cdot \mu[k]. \tag{2.49}$$

Proof. Due to the memorylessness of the Markov chain, these statements follow directly from the application of the definition of the transition probabilities. □

To single out certain Markov states, an indicator function is used:

Definition 2.9. *Let the indicator function for the Markov states be defined as follows (cf. [39]):*

$$\mathbb{1}_{\{\theta_k = i\}} := \begin{cases} 0, & \text{if } \theta_k \neq i \\ 1, & \text{if } \theta_k = i \end{cases}. \tag{2.50}$$

Lemma 2.7. *For indicator functions of subsequent time steps, it holds (cf. [135]):*

$$\mathrm{E}\left(\mathbb{1}_{\{\theta_{k+j} = i\}}\right) = \mathrm{E}\left(\sum_{m=1}^{n_\theta} p_{m,i}[k+j-1] \cdot \mathbb{1}_{\{\theta_{k+j-1} = m\}}\right). \tag{2.51}$$

Proof. The equality follows directly from the memorylessness of the Markov chain and the definition of the transition probabilities. □

Note that the indicator function and the probability distribution of the Markov state $\mu[k]$ are related as follows:

$$\mathrm{E}\left(\mathbb{1}_{\{\theta_k = i\}}\right) = \mu_i[k]. \tag{2.52}$$

Let the following time-variant linear operators be defined for a set of Markov state dependent matrices $\psi = \{\psi_1, \ldots, \psi_{n_\theta}\}$:

$$\mathcal{V}_i(\psi, j) := \sum_{m=1}^{n_\theta} p_{m,i}[k+j] \cdot \psi_m \qquad i \in \Theta, \tag{2.53}$$

$$\mathcal{T}_i(\psi, j) := \sum_{m=1}^{n_\theta} p_{i,m}[k+j] \cdot \psi_m \qquad i \in \Theta. \tag{2.54}$$

If the transition probabilities are time-invariant, the time arguments j and $k+j$ are omitted. In this case, the definitions coincide with the operators commonly used in the context of JMLS (see, e.g., [39]). The operator $\mathcal{V}_i(\cdot)$ calculates a weighted sum of matrices ψ_m for all transitions leading to the Markov states i. On the other hand, the operator $\mathcal{T}_i(\cdot)$ calculates a weighted sum of matrices ψ_m for all transitions that originate from the Markov states i.

Steady State Definition

Considering the Markov state transitions, three possibilities arise to define steady states for a JMS:

1. For states, outputs, and inputs, a separate steady state is defined for each Markov state.

2. For states and outputs, a steady state independent of the Markov state is defined; the corresponding input signals depend on the Markov state.

3. A common steady state independent of the Markov state is defined for states, outputs, and inputs.

Throughout this thesis, the second definition is used. This is motivated by the fact that the states and outputs are used to model product properties and production unit states. The goal is to keep these quantities as close as possible to a desired value, despite abrupt faults modeled by the Markov states. Furthermore, it is highly unlikely that the corresponding inputs are independent of the Markov state. For the same reason, u_r is allowed to be time-variant if the system dynamics is time-variant. Hence, the steady states for a nonlinear and a linear JMS are defined as follows:

Definition 2.10. *The steady state* x_r, y_r, *and* $U_r[k] = \{u_{r,i}[k] : i \in \Theta\}$ *for a JMS* $\mathcal{S}$ *is defined as follows:*

$$x_r = f_i(x_r, u_{r,i}[k], \bar{w}, k) \qquad \forall\ i \in \Theta, \tag{2.55}$$
$$y_r = g_i(x_r, u_{r,i}[k], \bar{w}, k) \qquad \forall\ i \in \Theta. \tag{2.56}$$

Definition 2.11. *The steady state* x_r, y_r, *and* $U_r[k] = \{u_{r,i}[k] : i \in \Theta\}$ *for a JMLS* $\mathcal{S}_l$ *is defined as follows:*

$$x_r = A_i[k]\,x_r + B_i[k]\,u_{r,i}[k] + G_i[k]\,\bar{w} \qquad \forall\ i \in \Theta, \tag{2.57}$$
$$y_r = C_i[k]\,x_r + D_i[k]\,u_{r,i}[k] + F_i[k]\,\bar{w} \qquad \forall\ i \in \Theta. \tag{2.58}$$

Stability

Common deterministic stability definitions, such as exponential stability, become very restrictive for JMS, due to the switching of the Markov state. Numerous probabilistic stability definitions have been proposed in literature to mitigate this issue (see, e.g., [39, 44, 81]). To state the stability definitions for an arbitrary steady state, a "delta system" state $x_\delta[k]$ is introduced as the deviation from the steady state:

$$x_\delta[k] := x[k] - x_r. \tag{2.59}$$

Stability Definitions: Using this definition, a collection of the most common stability definitions is presented below:

Definition 2.12. *Stochastic n-th Moment Stability [44]: A system is said to be stochastically n-th moment stable if:*

$$\sum_{k=0}^{\infty} \mathrm{E}\left(\|x_\delta[k]\|^n\right) < \infty. \tag{2.60}$$

Definition 2.13. ***Asymptotic n-th Moment Stability [81]:*** *A system is said to be asymptotically n-th moment stable if:*

$$\lim_{k \to \infty} \mathrm{E}\left(\|x_\delta[k]\|^n\right) = 0. \tag{2.61}$$

For $n = 2$, this definition coincides with the well known definition of mean square stability (MSS). The notion of MSS is most commonly used in the context of JMS and implies stochastic stability [39]. Obviously, the definition only applies for JMS with finite energy disturbances. For persistent, normally distributed disturbances, the second moment is required to converge to a constant covariance matrix [39]. However, MSS does not guarantee any convergence rate for general JMS. A stronger definition is given by practical n-th moment stability:

Definition 2.14. ***Practical n-th Moment Stability:*** *A system is called practically n-th moment stable, if there exist positive constants c_1, c_2, and $\lambda \in [0, 1)$, such that:*

$$\mathrm{E}\left(\|x_\delta[k+j]\|^n\right) \le c_1 \lambda^j \|x_\delta[k]\|^n + c_2, \qquad \forall \ k, j \in \mathbb{N}_0. \tag{2.62}$$

A zero disturbance ($w[k] = 0$ for all k) allows $c_2 = 0$ (cf. [15, 81] for the disturbance-free case). If $n = 2$, the system is called practically mean square stable (PMSS).

In this thesis, the notion of PMSS is used, since this definition implies MSS and stochastic stability (cf. [39]) and deals with persistent disturbances explicitly.

Stability Conditions: In the context of JMS, a switched quadratic Lyapunov function is commonly used to state stability conditions.

Definition 2.15. *Let the matrix set $\mathscr{P} = \left\{ \mathscr{P}_i \in \mathbb{S}_{n_x}^{>0} : i \in \Theta \right\}$ define a switched quadratic Lyapunov function of the form:*

$$V\left(x_\delta[k], \theta_k\right) := \|x_\delta[k]\|_{\mathscr{P}_{\theta_k}}^2. \tag{2.63}$$

Based on this definition, a condition for MSS stability can be stated:

Lemma 2.8. *A time-invariant, autonomous JMLS $\mathcal{S}_l$ is MSS if a set $\mathscr{P}$ exists, such that*

$$\mathscr{P}_i - A_i^\top \mathcal{T}_i(\mathscr{P}) A_i > 0 \qquad\qquad \forall \ i \in \Theta. \tag{2.64}$$

Proof. See [39] for the detailed proof. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Lemma 2.9. *Let the following inequality hold for a time-invariant, autonomous JMLS $\mathcal{S}_l$:*

$$\|A_i x_\delta[k]\|_{\mathcal{T}_i(\mathscr{P})}^2 < \|x_\delta[k]\|_{\mathscr{P}_i}^2 \tag{2.65}$$

with $\|x_\delta[k]\|_{\mathscr{P}_i} = 1$, $\mathscr{P}_i > 0$, and $i \in \Theta$. Then, inequality (2.65) holds for all $x_\delta[k] \ne \mathbf{0}$.

Proof. Let $x'[k] \coloneqq \frac{x_\delta[k]}{\|x_\delta[k]\|_{\mathscr{P}_i}}$ for all $x_\delta[k] \neq \mathbf{0}$. Employing that $\|x'[k]\|_{\mathscr{P}_i} = 1$ and inequality (2.65), one has that:

$$\|A_i x_\delta[k]\|^2_{\mathcal{T}_i(\mathscr{P})} = \|A_i x'[k]\|^2_{\mathcal{T}_i(\mathscr{P})} \cdot \|x_\delta[k]\|^2_{\mathscr{P}_i} < \|x'[k]\|^2_{\mathscr{P}_i} \cdot \|x_\delta[k]\|^2_{\mathscr{P}_i} = \|x_\delta[k]\|^2_{\mathscr{P}_i} \quad (2.66)$$

holds for all $x_\delta[k] \neq \mathbf{0}$. □

A more general stability condition for PMSS is given by [133]:

Lemma 2.10. *A JMLS $\mathcal{S}_\mathrm{l}$ is PMSS if a Lyapunov matrix set $\mathscr{P}$ and a matrix $L \in \mathbb{S}^{>0}_{n_\mathrm{x}}$ exist, such that*

$$\|A_i[k] x[k] + B_i[k] u[k] + G_i[k] \bar{w}[k] - x_\mathrm{r}\|^2_{\mathcal{T}_i(\mathscr{P},0)} - \|x_\delta[k]\|^2_{\mathscr{P}_i} \leq -\|x_\delta[k]\|^2_L \quad (2.67)$$

holds for all $i \in \Theta$ in each time k with $w[k] = \bar{w}[k] + \tilde{w}[k]$ and $\mathrm{E}(\tilde{w}[k]) = 0$.

Proof. The proof is a generalization of results from [15] and [32]. Employing (2.47) one obtains:

$$\mathrm{E}\left(V(x_\delta[k+1], \theta_{k+1})\right) - V\left(x_\delta[k], \theta_k\right) \quad (2.68)$$

$$= \mathrm{E}\left(\|A_{\theta_k}[k] x[k] + B_{\theta_k}[k] u[k] + G_{\theta_k}[k] w[k] - x_\mathrm{r}\|^2_{\mathscr{P}_{\theta_{k+1}}}\right) - \|x_\delta[k]\|^2_{\mathscr{P}_{\theta_k}}$$

$$= \|A_{\theta_k}[k] x[k] + B_{\theta_k}[k] u[k] + G_{\theta_k}[k] \bar{w}[k] - x_\mathrm{r}\|^2_{\mathcal{T}_{\theta_k}(\mathscr{P},0)} - \|x[k]\|^2_{\mathscr{P}_{\theta_k}} + \mathrm{E}\left(\|G_{\theta_k} \tilde{w}[k]\|^2_{\mathscr{P}_{\theta_{k+1}}}\right)$$

$$+ 2\mathrm{E}\left((A_{\theta_k}[k] x[k] + B_{\theta_k}[k] u[k] + G_{\theta_k}[k] \bar{w}[k] - x_\mathrm{r}) \mathscr{P}_{\theta_{k+1}} G_{\theta_k} \tilde{w}[k]\right)$$

$$\leq \mathrm{E}\left(\|G_{\theta_k} \tilde{w}[k]\|^2_{\mathscr{P}_{\theta_{k+1}}}\right) - \|x_\delta[k]\|^2_L.$$

The last inequality directly follows from (2.67), the fact that $\mathrm{E}(\tilde{w}[k]) = 0$, and the stochastic independence of the disturbance $\tilde{w}[k]$. For the right-most side of (2.68), the following over-approximation holds (cf. [32]):

$$\mathrm{E}\left(\|G_{\theta_k} \tilde{w}[k]\|^2_{\mathscr{P}_{\theta_{k+1}}}\right) - \|x_\delta[k]\|^2_L \leq \max_{i \in \Theta} \lambda_{\max}\left(\|G_i\|^2_{\mathcal{T}_i(\mathscr{P},0)}\right) \mathrm{E}\left(\|\tilde{w}[k]\|^2\right) - \lambda_{\min}(L) \|x_\delta[k]\|^2$$

$$= \underbrace{\max_{i \in \Theta} \lambda_{\max}\left(\|G_i\|^2_{\mathcal{T}_i(\mathscr{P},0)}\right) \mathrm{tr}\left(\Sigma_\mathrm{w}\right)}_{=\alpha} - \lambda_{\min}(L) \|x_\delta[k]\|^2$$

$$\leq \frac{-\lambda_{\min}(L)}{\max_{i \in \Theta} \lambda_{\max}(\mathscr{P}_i)} V\left(x_\delta[k], \theta_k\right) + \alpha. \quad (2.69)$$

From (2.68) and (2.69), it follows that:

$$\mathrm{E}\left(V(x_\delta[k+1], \theta_{k+1})\right) - V\left(x_\delta[k], \theta_k\right) \leq \frac{-\lambda_{\min}(L)}{\max_{i \in \Theta} \lambda_{\max}(\mathscr{P}_i)} V\left(x_\delta[k], \theta_k\right) + \alpha$$

$$\Rightarrow \mathrm{E}\left(V(x_\delta[k+1], \theta_{k+1})\right) \leq \underbrace{\left(1 - \frac{\lambda_{\min}(L)}{\max_{i \in \Theta} \lambda_{\max}(\mathscr{P}_i)}\right)}_{=:\lambda} V\left(x_\delta[k], \theta_k\right) + \alpha.$$

$$(2.70)$$

When (2.70) is applied recursively j-times, it follows:

$$
\begin{aligned}
\mathrm{E}\left(V\left(x_\delta[k+j],\theta_{k+j}\right)\right) &\leq \lambda^j V\left(x_\delta[k],\theta_k\right) + \alpha \sum_{l=1}^{j-1} \lambda^{j-l-1} \\
&\leq \lambda^j V\left(x_\delta[k],\theta_k\right) + \alpha \sum_{l=0}^{\infty} \lambda^l \\
&\leq \lambda^j V\left(x_\delta[k],\theta_k\right) + \frac{\alpha}{1-\lambda}.
\end{aligned}
\tag{2.71}
$$

In the last inequality, the geometric series can be applied, since $\mathscr{P}_i > 0$ and $L > 0$ imply that $\lambda < 1$, and $V\left(x_\delta[k+j],\theta_{k+j}\right) \geq 0$ implies that $\lambda \geq 0$. Hence, the resulting expression can be approximated by:

$$
\min_{i\in\Theta} \lambda_{\min}\left(\mathscr{P}_i\right) \mathrm{E}\left(\|x_\delta[\![j]\!]\|^2\right) \leq \lambda^j \max_{i\in\Theta} \lambda_{\max}\left(\mathscr{P}_i\right) \|x_\delta[k]\|^2 + \frac{\alpha}{1-\lambda}.
\tag{2.72}
$$

All in all, the following approximation follows:

$$
\mathrm{E}\left(\|x_\delta[\![j]\!]\|^2\right) \leq \underbrace{\frac{\max\limits_{i\in\Theta} \lambda_{\max}\left(\mathscr{P}_i\right)}{\min\limits_{i\in\Theta} \lambda_{\min}\left(\mathscr{P}_i\right)}}_{=:c_1} \lambda^j \|x_\delta[k]\|^2 + \underbrace{\frac{\alpha(1-\lambda)^{-1}}{\min\limits_{i\in\Theta} \lambda_{\min}\left(\mathscr{P}_i\right)}}_{=:c_2}.
\tag{2.73}
$$

This inequality coincides with the definition of PMSS in (2.62). If no disturbance is present, Σ_{w} equals the zero matrix and $\alpha = 0$ holds. $\qquad\square$

Corollary 2.2. *Let $\mathcal{S}_l$ be a linearization of the nonlinear JMS $\mathcal{S}$ with bounded linearization error. Then, $\mathcal{S}$ is PMSS if (2.67) is satisfied for $\mathcal{S}_l$ in each time k.*

Proof. (sketch) PMSS is guaranteed by condition (2.67), since the expected value of the influence of the disturbance $\mathrm{E}\left(\|G_{\theta_k}\tilde{w}[k]\|^2_{\mathscr{P}_{\theta_{k+1}}}\right)$ is bounded by α (cf. Equation (2.69)). Lemma 2.10 also holds for bounded disturbances with unknown mean and covariance, e.g., resulting from the linearization error. In this case, only the formulation of the bound α has to be changed in the proof of Lemma 2.10. $\qquad\square$

Part II.

Modeling of Distributed Manufacturing Systems

3. A General Modeling Framework for Manufacturing Systems

In this chapter, a modeling framework is presented for multistage manufacturing systems which are subject to continuous and abrupt disturbances. A special focus is on the dynamic behavior of the product properties.

3.1. Problem Description and Modeling Concept

In the manufacturing processes under consideration, semi-finished products pass multiple production stages sequentially until a final product is produced. For capacity reasons, each stage can consist of one or more parallel production units. The production units are also referred to as subsystems $\mathcal{S}^n$, where n is the subsystem index. All quantities related to this subsystem are indicated by a bold faced upper index. Each production unit may produce one or more products in parallel (referred to as a product group). Hence, a production unit receives a product group from a set of preceding production units, and delivers the processed product group to a set of subsequent production units in each production cycle. The first production units receive the raw materials or components, while the final products are delivered from the last production units. For the product flow, the following assumptions hold:

Assumption 3.1. *No buffers are present within the whole production process.*

The resulting problem is not as general as a production process with arbitrary buffer positions. However, since the focus is on the control of the product properties, this assumption makes the controller design more challenging. This is due to the fact that the cycle times of all subsystems are synchronized and cannot be changed for just one subsystem. Furthermore, the dynamic couplings are stronger. Hence, for most processes with buffers, the presented approach is still applicable.

Assumption 3.2. *A product group is processed at most once by a production unit.*

Hence, the production process is governed by an arbitrary acyclic topology. This assumption is made to ease the notation and derivations of the distributed MPC. However, it would be possible to extend the presented approaches to consider cyclic processes. In addition, this assumption is motivated by the fact that for most large-scale manufacturing processes each production unit has a very specialized purpose, like pressing, welding, or drilling, that cannot be repeated for the same product.
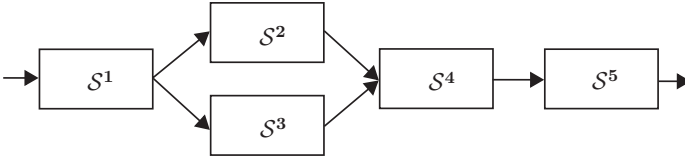
Figure 3.1.: Production process consisting of five subsystems.

For some processes, such as heat treatment in furnaces, the processing time can be considerably larger than the cycle time of the production line. Hence, two options arise as no buffers are present:

- Multiple product groups of different production cycles are processed in one production unit in parallel. The product groups are at different processing steps.

- Multiple production units process product groups in parallel at higher cycle times in an alternating pattern. Since buffers do not exist, the cycle times of subsequent production units have to be integral multiples of the fastest subsystem.

Example 3.1. In Fig. 3.1, a block diagram of an exemplary production process $\mathcal{S}$ with $n_{\mathrm{s}} = 5$ subsystems is shown. Subsystem 1 receives the raw material, processes it, and provides the resulting product group to subsystems 2 and 3. These subsystems process the products and both deliver their products to subsystem 4, and so forth. This setup could define a scenario in which $\mathcal{S}^{\mathbf{1}}$ models a separating process with subsequent processing of the two products and a successive joining process in $\mathcal{S}^{\mathbf{4}}$. A different scenario results if all production units process same product groups. In this case, the production units $\mathcal{S}^{\mathbf{2}}$ and $\mathcal{S}^{\mathbf{3}}$ can be operated at double cycle time compared to $\mathcal{S}^{\mathbf{1}}$ and $\mathcal{S}^{\mathbf{4}}$ and supply the product groups alternatingly to $\mathcal{S}^{\mathbf{4}}$.   △

As stated in the introduction, a modeling framework for this setup can be derived by combining ideas from the Stream of Variation (SoV) framework with JMS that model the production units. The main ideas adopted from the SoV framework are (cf. Section 1.2.1):

- Each system is modeled by a nominal model. Continuous disturbances and model errors are modeled by auto regressive models.

- The output of a production unit becomes the disturbance of the downstream (subsequent) production units.

- Each production unit has a sampling time that equals its cycle time.

In contrast to the SoV framework, the production units are modeled by JMS, in order to describe abrupt disturbances and the dynamics of the subsystems more accurately. Both, the modeling of the subsystems by JMS and their couplings according to the SoV ideas, are presented in the following sections.

3.2. Modeling of a Production Unit

In this section, a modeling setup for the production units is introduced. For the sake of simple notation, the time index k^n of subsystem $\mathcal{S}^n$ is denoted by just k if it indexes a quantity of the same subsystem, e.g., $x^n[k^n]$ is abbreviated by $x^n[k]$. The same holds for the Markov state, i.e., $f^n_{\theta^n_k} = f^n_{\theta_k}$.

3.2.1. Modeling Concept

A production unit $\mathcal{S}^n$ is considered a dynamic process that transforms a product group entering the production unit into a new product group. During that process, a set of properties that characterizes the quality of the produced parts is transformed into a new set of properties. The properties are physically measurable quantities, like temperature, geometry, or hardness. The properties of the product group entering the production unit in production cycle k are considered as a disturbance $w^n_{\mathrm{p}}[k]$, since these cannot be influenced by production unit $\mathcal{S}^n$. The properties of the product group leaving the production unit $\mathcal{S}^n$ are considered as outputs $y^n[k]$. The resulting properties $y^n[k]$ depend on the process variables (inputs $u^n[k]$), disturbances $w^n_{\mathrm{d}}[k]$ and $w^n_{\mathrm{p}}[k]$, and the state of the production system $x^n[k]$. There are basically three types of disturbances influencing the production process:

- Abrupt disturbances and changes in the process dynamics caused by, e.g., component defects. These disturbances are modeled by a Markov chain $\mathcal{M}^n$.

- Continuous disturbances that are independent of the production process (the number of produced parts), such as ambient temperature or humidity. These disturbances are collected in $w^n_{\mathrm{d}}[k]$.

- Continuous disturbances that are increasing with the number of produced parts due to wear processes. This may lead to drifts in the states $x^n[k]$ or the product properties $y^n[k]$. These effects are modeled by additive processes $\nu[k]$.

All in all, the production process can be modeled as follows:

Definition 3.1. *The processing in each production unit is modeled by nonlinear discrete-time systems. For the different faults and operation modes, different dynamics are determined. The transitions between these models are described by a Markov chain $\mathcal{M}^n$ resulting in a JMS (cf. Section 2.7):*

$$\mathcal{S}^n := \begin{cases} x^n[k+1] & = & f^n_{\theta_k}\big(x^n[k], u^n[k], w^n[k]\big) + \nu^n_{\mathrm{x}}[k] \\ y^n[k] & = & g^n_{\theta_k}\big(x^n[k], u^n[k], w^n[k]\big) + \nu^n_{\mathrm{y}}[k] \\ \mathcal{M}^n & = & \big(\Theta^n, P^n[k], \mu^n[k]\big) \end{cases} . \tag{3.1}$$

The system state $x^n[k]$ is composed of the states of the production system and the states of currently processed product groups. The disturbance input $w^n[k]$ consists
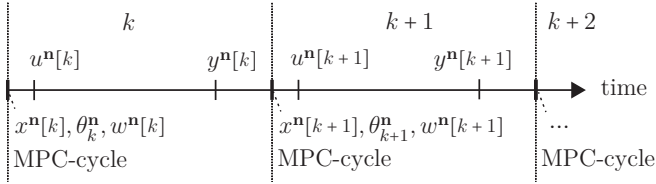
Figure 3.2.: Timing diagram of the n-th subsystem. It illustrates that k is a cycle counter and when the quantities of the subsystem are determined.

of the properties of the product group entering the production unit $w_\mathrm{p}^m[k]$ and disturbances independent of the products $w_\mathrm{d}^m[k]$, e.g., ambient temperature. Here, the functions $f_{\theta_k}^n(\cdot)$ and $g_{\theta_k}^n(\cdot)$ describe the model of the production unit approximately. The additive processes $\nu_\mathrm{x}^n[k]$ and $\nu_\mathrm{y}^n[k]$ describe a possible plant model mismatch. The mismatch can be caused, e.g., by the time discretization, by model abstractions, and by the continuous uncertainties, caused e.g. by wear or dirt.

In this setup, all system quantities are sampled once in each production cycle. The time index k does not necessarily indicate quantities sampled at the same time, but refers to quantities that are related to the k-th production cycle (cf. the timing diagram in Fig. 3.2). Hence, k can be interpreted also as a cycle index. For example, the properties of the product group entering the production unit are valid at the beginning of the production cycle and change during the cycle. The process variables may influence the process at different times during the production cycle. The properties of the product group exiting the production unit are available at the end of the production cycle. To be able to use a common discrete-time setting, these signals are assumed to be constant during one production cycle and are sampled at the moment when the MPC execution is started. In order to state the dynamics of the subsystems properly and use the cycle counter k also as the time index, the sampling time, i.e., the cycle time, of the subsystems has to be constant / time-invariant. The output $y^n[k]$ is not determined when the subsystem is sampled, since it results from the processing with the variables $u^n[k]$. Thus, the output equation can be interpreted as a prediction of the resulting product properties based on the properties of the entering product group, the system states, the disturbances, and the process variables. This level of abstraction is used because the control approach aims at the determination of process variables for one production cycle and not of the actuator signals during that production cycle. Hence, the modeling framework focuses on the resulting product properties as a function of the process variables by abstracting the dynamics and output equations to nonlinear discrete-time equations.

If the processing time is larger than the cycle time, multiple product groups may be processed simultaneously. Hence, $w_\mathrm{p}^n[k]$ and $y^n[k]$ describe properties of different product groups, i.e., of the product groups entering and exiting the production unit in the cycle k. The properties of the intermediate product groups (that are currently processed) are modeled by additional subsystem states.

3.2.2. Disturbance Models

The accuracy of the prediction model is crucial for MPC. The concept of enhancing the prediction quality by autoregressive disturbance models is well known and has been proposed for different applications, e.g., industrial applications [59, 101].

To improve the model accuracy, the plant model mismatch processes $\nu_x^n[k]$ and $\nu_y^n[k]$ are approximated by autoregressive integrated moving average (ARIMA) models wherever the corresponding states or outputs are measurable. Here, ARIMA models of the following form are used for each component $\nu_{x,l}^n[k]$ of $\nu_x^n[k]$, cf. [22]:

$$\check{\nu}_{x,l}^n[k] = \nu_{x,l}^n[k-1] + \sum_{j=1}^{m} \varphi_{l,j}[k] \cdot \left(\nu_{x,l}^n[k-j] - \nu_{x,l}^n[k-j-1] \right). \tag{3.2}$$

The $\check{}$ indicates that the corresponding quantity is an estimation of the real plant model mismatch, and m is the order of the ARIMA model. Analogous definitions are used for $\check{\nu}_y^n[k]$. An appropriate choice of the model order can be determined by an analysis of the autocorrelation function and the partial autocorrelation function of historical measurement data [22]. The model parameters $\varphi_{l,j}[k]$ are assumed to be time-varying, since the model mismatch is caused partly by time-varying processes, such as tool wear. To determine the parameters $\varphi_{l,j}[k]$, the well known *recursive least squares* algorithm is employed (see [122] for details), using the measurements of $x^n[k]$ and $y^n[k-1]$.

If not stated otherwise, the disturbance models are not included in the state vector $x^n[k]$ of the subsystem, in order to keep the system dimension and the calculation time for the MPC as low as possible. Instead, the n-th order ARIMA models are used to predict the disturbance trajectories $\check{\nu}_x^n[1], ..., \check{\nu}_x^n[N]$ and $\check{\nu}_y^n[0], ..., \check{\nu}_y^n[N]$, when the corresponding measurements are performed and no computation capacity is needed for the MPC. Then, these trajectories are used by the MPC at the next sampling instant. The advantage of this approach becomes clear when the model of the hot stamping press is considered (see Section 4.2.4). Here, disturbance models for 24 temperatures are introduced. For each second order ARIMA model, three additional states would be needed. Hence, the dimension of the state vector would increase from 26 to 98. This results in a severe increase of the computational effort for the MPC.

3.2.3. Markov Chain Modeling of Breakdown and Repair Events

The Markov chain introduced in (3.1) is used to model abrupt process uncertainties. Most of these uncertainties arise from the breakdown of a component of a production unit or even the complete production line. Markov chain models for breakdown and repair events can be derived from historical production data. Assuming that breakdown and production times can be approximated by an exponential distribution, a simple two state Markov chain that models the normal and faulty state can be derived (cf. [14]). The Markov chain (depicted in Fig. 3.3) is completely parameterized by the breakdown and repair rates λ_f and λ_r, which can be derived as follows [14]:
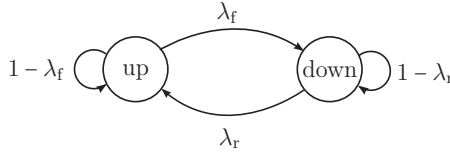
Figure 3.3.: Markov chain modeling breakdowns and repairs, cf. [14].

$$\lambda_{\text{f}} = \text{MTBF}^{-1} \qquad \lambda_{\text{r}} = \text{MTTR}^{-1}. \tag{3.3}$$

The mean time between failures (MTBF) and mean time to repair (MTTR) can be obtained from historical production fault recordings. The MTBF is calculated by the quotient of the whole normal operation time and the total number of faults. The MTTR is determined by the quotient of the total fault time and the number of faults recorded over a certain time. For the modeling of the production unit according to (3.1), a discrete-time Markov chain is needed. Hence, the operation time and fault duration have to be used as multiples of the cycle time and not in continuous time. Using this approach, the determination of Markov chains for component faults is straightforward. Once these chains are determined for all significant faults, an overall Markov chain can be derived by parallel composition of these Markov chains (for details see [129]).

This approach is only an approximation of the real breakdown and repair processes. The histograms depicted in Fig. 3.4 show repair times of production unit components. The diagrams show distributions similar to exponential distributions,
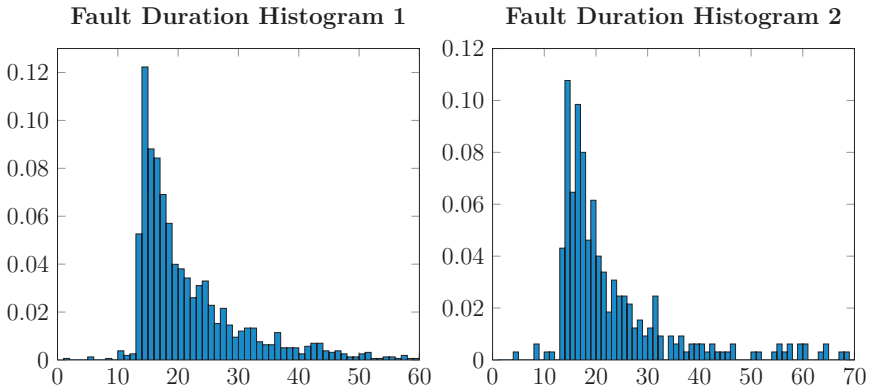
Figure 3.4.: Fault duration histogram of two production line components. The abscissa shows the fault duration in multiples of the cycle time.

Fault Duration Histogram 3
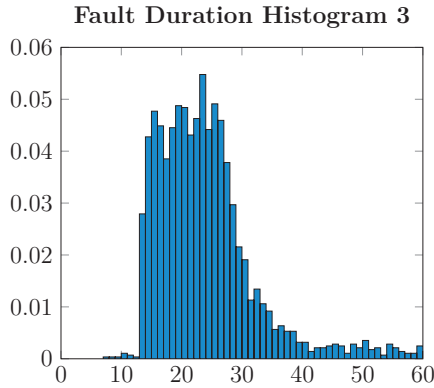
Figure 3.5.: A fault duration histogram of a production line component. The abscissa shows the fault duration in multiples of the cycle time.

but with a dead time. The dead time is caused by a minimal reaction and repair time. This behavior could still be modeled by a homogeneous Markov chain using waiting states. However, this approach would increase the number of Markov states significantly. Furthermore, fault distributions may exhibit much broader peaks (see Fig. 3.5). This behavior cannot be modeled accurately by homogeneous Markov chains. However, non-exponential distributions can be modeled quite accurate by a non-homogeneous Markov chain, i.e., by defining a time-varying repair rate $\lambda_r[k]$. This is particularly true for distributions with dead times. Hence, the repair process for each component can be modeled as follows: As long as the system is in normal operation, a homogeneous mean repair rate λ_r is used. Once a specific fault occurs, the repair rate is replaced by a time-varying one, such that the resulting distribution fits the fault distribution from fault recordings. The time-varying repair rate can be determined by transforming the non-conditional repair probabilities of the fault distribution p_d (from recordings) into a conditional probability (conditioned on the fact that the component has not been repaired until now):

Algorithm 3.1 Calculation of time-varying transition rates from a distribution p_d.

1: **Initialization** $\lambda_r[k] = p_d(1)$
2: **for** $n = 1, 2, \ldots$ **do**
3: $\quad \lambda_r[k+n] = \dfrac{p_d(n+1)}{\prod\limits_{m=0}^{n-1} (1-\lambda_r[k+m])}$
4: **end for**

The accuracy of the breakdown modeling can be increased analogously with a time-varying fault probability $\lambda_f[k]$.

3.3. Coupling of the Production Units

The model of the whole manufacturing system results from an appropriate coupling of the JMS that model the different production units. In the manufacturing systems under consideration, the couplings are caused by the products that are processed by the different production units and their properties. The formalization of these couplings is described in this section.

Cycle Times and Time Indices

As demonstrated in Section 3.1, the cycle times may vary between the subsystems. Due to the lack of buffers, the cycle times have to be an integral multiple of the lowest cycle time $t_{c,b}$. Hence, the time indices k^n are incremented by one in certain multiples of the time index of the fastest subsystem. The corresponding time index defines a base cycle for the whole production process and is denoted by k_b. Then, the local time indices k^n and the base cycle k_b are linked by the following functions:

$$k^n = \kappa_n(k_b) := \left\lfloor \frac{t_{c,b}}{t_c^n}(k_b + o^n) \right\rfloor, \tag{3.4}$$

$$k_b = \kappa_n^{-1}(k^n) := \frac{t_c^n}{t_{c,b}}k^n - o^n, \tag{3.5}$$

where $o^n \in \{0, \dots, {}^{t_c^n}/_{t_{c,b}} - 1\}$ is a counter offset that can be used to define in which base cycle parallel production units process products. $\lfloor \cdot \rfloor$ denotes the floor function. Equation (3.5) describes the first k_b for which k^n is incremented to its current value.

Product Flow

The product group flow is modeled by a directed, acyclic graph $\mathfrak{G}[k_b] = (\mathfrak{N}, \mathfrak{E}[k_b])$, where the nodes $\mathfrak{N} = \mathbb{N}_{n_s}$ represent the n_s production units $\mathcal{S}^n$, and the edges $\mathfrak{E}[k_b] \subset \mathfrak{N} \times \mathfrak{N}$ represent all transfers of product groups between the production units in the base cycle k_b. The sequence of production units, by which each product group is processed, may change with time. Hence, the edge set $\mathfrak{E}[k_b]$ may be time-varying.

Example 3.2. Considering the process in Example 3.1 from page 40, the corresponding node set is given by $\mathfrak{N} = \mathbb{N}_5$. If both $\mathcal{S}^2$ and $\mathcal{S}^3$ are operated in parallel, the resulting edge set is time-invariant:

$$\mathfrak{E} = \{(1,2),(1,3),(2,4),(3,4),(4,5)\}. \tag{3.6}$$

If these production units are operated alternatingly at double cycle time $2t_{c,b}$, the edge set becomes time-varying, and two possible setups exist:

$$1: \quad \mathfrak{E}[k_b] = \left\{\big(1, 2 + k_b \bmod 2\big), \big(2 + (k_b - 1) \bmod 2, 4\big), (4,5)\right\}, \tag{3.7}$$

$$2: \quad \mathfrak{E}[k_b] = \left\{\big(1, 2 + (k_b + 1) \bmod 2\big), \big(2 + k_b \bmod 2, 4\big), (4,5)\right\}. \tag{3.8}$$

The offsets o^2 and o^3 can be used to discriminate between these two setups. $\quad \triangle$

Production units delivering products to $\mathcal{S}^n$ are called predecessors of $\mathcal{S}^n$, and production units receiving products from $\mathcal{S}^n$ are called successors or downstream processes of $\mathcal{S}^n$. The corresponding index sets are given by:

$$\text{pre}(\mathcal{S}^n, k_{\text{b}}) = \text{pre}(n, k_{\text{b}}) \quad := \quad \{m \in \mathbb{N}_{n_{\text{s}}} \mid (\mathfrak{N}_m, \mathfrak{N}_n) \in \mathfrak{E}[k_{\text{b}}]\}, \tag{3.9}$$

$$\text{succ}(\mathcal{S}^n, k_{\text{b}}) = \text{succ}(n, k_{\text{b}}) \quad := \quad \{m \in \mathbb{N}_{n_{\text{s}}} \mid (\mathfrak{N}_n, \mathfrak{N}_m) \in \mathfrak{E}[k_{\text{b}}]\}. \tag{3.10}$$

In some sections, the index set of all downstream subsystems until the end of the production process (not only the direct successors) is needed. Let this set be given by $\text{Succ}(n, k_{\text{b}})$.

Dynamic Coupling

The following assumption is made concerning the coupling of subsystems:

Assumption 3.3. *The dynamics of the production units are only coupled by the properties of the processed product groups.*

This assumption is well motivated by real production systems. In many cases, there are no direct input or state couplings of different production units, but the resulting properties of one production unit may influence the dynamic of the subsequent processes. Hence, the whole production system $\mathcal{S}$ is defined by the dynamics of the subsystems $\mathcal{S}^n$ coupled by the product flow defined by the edges $\mathfrak{E}[k_{\text{b}}]$. From the modeling of the subsystems according to Section 3.2, it follows that these couplings require that the system output $y[k]$ equals the disturbance vector $w_{\text{p}}[k]$ of the subsequent production units. Hence, the following implication has to hold:

$$(n_1, n_2) \in \mathfrak{E}[k_{\text{b}}] \quad \Rightarrow \quad w_{\text{p}}^{n_2}[\kappa_{n_2}(k_{\text{b}})] = y^{n_1}[\kappa_{n_1}(k_{\text{b}})] \quad \forall \ k_{\text{b}}. \tag{3.11}$$

If products are delivered to multiple successors, the implication (3.11) has to be defined for the corresponding subvectors of $w_{\text{p}}^{n_2}$ and y^{n_1}.

The subsystems are sampled at the moment when the product group is ready for processing and the MPC procedure is started. Combined with the coupling structure presented in this section, this results in an **asynchronous sampling and MPC execution in the different subsystems**. This is illustrated in Fig. 3.6. The dashed arrows illustrate the transfer of a product group to the subsequent production unit. The sampling literally follows the product flow. The asynchronous execution of the MPCs has a strong impact on the implementation of a distributed MPC architecture. The details are discussed in the Chapters 10 and 11.

The influence of the abrupt faults on the product properties are modeled by the local JMS. As long as these faults do not cause a complete production unit breakdown, the influence of faults in one production unit on the dynamics of the successive production units is already modeled by the described output-disturbance couplings. However, if one production unit stops completely, the whole product flow is changed. Preceding production units cannot deliver their products to the faulty
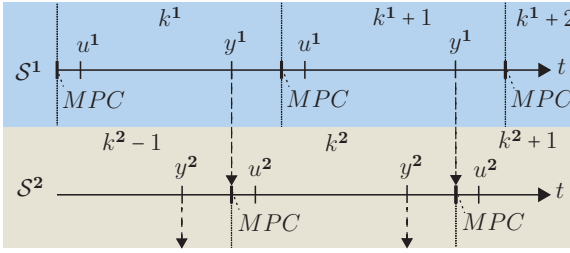
Figure 3.6.: Exemplary timing diagram of two subsequent subsystems.

production unit and the supply of successive production units stops due to the lack of buffers. If these 'stopping' events influence the dynamic behavior of a production unit, the coupling is not only based on the product groups' properties but also on the discrete event behavior of all production units. For example, this is the case for heat treatment processes and processes where tool temperatures have a significant influence on the product properties. In these cases, the events have to be modeled by the Markov chains of the production units in addition to the local faults. Then, the local transition probabilities of the Markov chains depend on the fault and repair rates of other production units.

To implement these couplings, the overall fault and repair rate for a set of production units has to be calculated from the individual fault and repair rates. To this end, completely serial manufacturing processes and manufacturing processes with parallel production units have to be considered separately.

Serial Processes: The whole production stops if one production unit stops, i.e., the overall fault rate is given by:

$$\lambda_{\mathrm{f}} = 1 - \prod_{n=1}^{n_{\mathrm{s}}}\left(1 - \lambda_{\mathrm{f},n}\right). \tag{3.12}$$

The overall repair rate λ_{r} can be obtained by calculating the overall MTTR for all subsystems. Once a production units fails, the repair rate λ_{r} has to be replaced by the repair rate of the faulty production unit.

Parallel Processes: Three cases have to be considered for the calculation of overall fault rates of parallel architectures:

- The parallel production units produce different products, all processed at once by the successors. In this case, the whole production stops if one production unit stops, i.e., the calculation of λ_{f} and λ_{r} is the same as for serial processes.

- The parallel production units deliver the products to the successors alternatingly. Hence, the products are processed by different sequences of subsystems in the

different base cycles (in Example 3.1: {1,2,4,5} and {1,3,4,5}). Each sequence can be regarded as a separate serial process, for which λ_f and λ_r can be calculated.

- The parallel production units are redundant and the production stops not until all production units fail. In this case, the failure rates of a network of redundant and serial processes have to be calculated. This can be done employing exponential reliability functions. Since this case is rare for the considered type of application, it is not considered in detail. The interested reader is referred to [111].

The combination of the JMS that model the different production units and their coupling according to this section provides the modeling framework for the manufacturing systems to be controlled. The application of this framework to a hot stamping production line is presented in the following chapter.

4. Modeling of a Hot Stamping Line

In this chapter, the modeling framework presented in Chapter 3 is applied to a hot stamping production line. After the hot stamping line is described, the modeling of the subsystems and their interconnection is presented.

4.1. Description of the Production Line

The hot stamping production line, considered throughout this thesis, is a production line located at the Volkswagen production plant in Baunatal. A similar production line is shown in Fig. 1.2 on page 5. The production line is used to produce the left and right rear wheelhouse frame rails (see Fig. 4.1 for the shape and location of these parts). Both parts are partially hardened and produced at the same time as pairs on the production line with a cycle time of about 18 seconds.

The line consists of two roller hearth furnaces (RHF) stacked over another, a chamber furnace that transports the metal sheets from the upper to the lower level, a transportation robot, and the press with two separate pressing tools for the left and right part. The two RHFs process the blanks alternatingly with the double cycle time $2t_c$ compared to the chamber furnace, the robot, and the press. A block diagram of the production line is shown in Fig. 4.2. The production process is defined by the following steps:

1. A pair of blanks is inserted alternatingly into RHF 1 or RHF 2. The blank pair is transported through the RHF at a constant speed v.
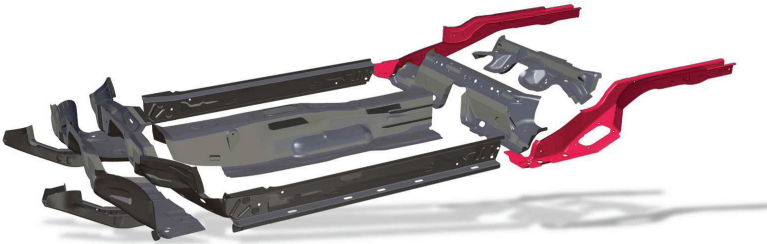
Figure 4.1.: The Volkswagen "Modularer Querbaukasten" (MQB) platform with highlighted rear wheelhouse frame rails.
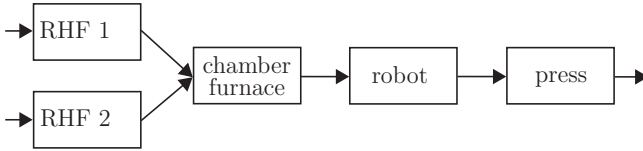
Figure 4.2.: Block diagram of the hot stamping production line.

2. Once the last furnace segment is reached, the blanks are ejected fast until the part of the blanks, that is supposed to be of lower hardness (referred to as soft part), has left the RHF. The hard part of the blanks remains in the RHF. The blanks are held in this position for the partial hardening time $t_{\mathrm{ph,1}}$.

3. The blank is transported into the chamber furnace and kept inside for a specified time t_{f}. In addition, a second (short) partial hardening process is performed at the exit of the chamber furnace for the time $t_{\mathrm{ph,2}}$.

4. The blanks are ejected from the chamber furnace and transported to the press by a robot. While the blanks exit the chamber furnace and after the blanks have been positioned in the pressing tool, a 2D temperature measurement of the blanks is performed (see left plot of Fig. 4.3).

5. The blanks are formed and quenched simultaneously in the press. After the tool has reopened, a third temperature measurement is performed (see right plot of Fig. 4.3).

Concerning the dynamic behavior, the modeling focuses on the blank temperatures and processing times, but not on the geometry. This is motivated by the fact that the mechanical properties of the final part are mainly influenced by the
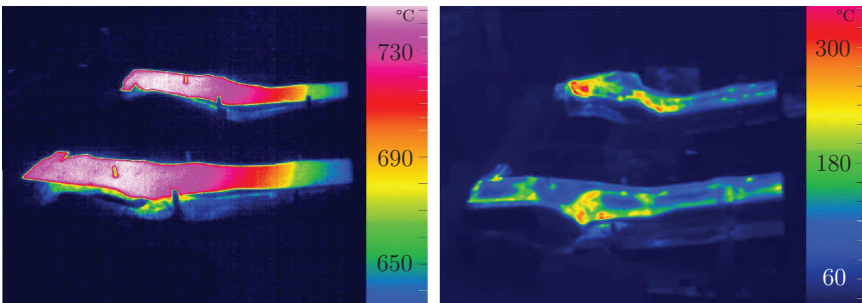
Figure 4.3.: Temperature distribution before (left) and after (right) forming. The hard / hot and soft / cold parts can be seen in the left picture.

blank temperature evolution. Furthermore, the blank temperatures are measured at several stages of the production process and the final geometry cannot be influenced significantly by the process variables. An illustration of the blank temperature evolution is presented in Fig. 4.10 on page 67.

The modeling of the abrupt faults focuses on complete production unit stops rather than smaller faults, since short stops of the whole production constitute one of the main causes for process variations in the hot stamping process. Furthermore, the consideration of complete production unit breakdowns results in a more complex coupling of the subsystems. This renders the MPC design more challenging and provides a more interesting test scenario.

4.2. Modeling of the Production Units

In this section, the modeling of the RHFs, the chamber furnace, the robot, and the pressing process according to Section 3.2 is described.

4.2.1. Roller Hearth Furnaces

The following assumption holds for the modeling of the RHFs:

Assumption 4.1. *The temperature of the left and the right blank are the same. The whole blank has the temperature $T_b(t)$ until partial hardening is performed.*

This assumption is well motivated by the fact that the left and right blank have the same geometry and are heated up in parallel. Furthermore, it has been shown in [130] that the reduction of the blank to a point mass is accurate enough for the modeling of the blank temperature dynamics. Based on the same reasoning, it is assumed that the temperature distribution of the blanks after the partial hardening can be approximated by two temperatures: a higher temperature $T_{b,h}$ in the hard part of the blank and a lower temperature $T_{b,s}$ in the soft part of the blank.

Both RHFs are identical in construction and consist of 9 temperature segments (cf. Fig. 4.4), that can be heated independently. The temperature distribution in the RHF is a function of the position s in transportation direction. The segment temperatures $T_{f,1}$ till $T_{f,9}$ are measured in the middle of each segment. The temperature profile $T_f(s)$ is approximated by a linear interpolation of the measured segment temperatures $T_{f,n}$. See Fig. 4.5 for an exemplary approximation of a temperature profile. As shown in [130] by the author, this approximation is accurate enough for modeling the blank temperature dynamics in the RHF.

A pair of blanks is entering the furnace every second cycle of the press. However, the necessary heat treatment time is considerably larger than the cycle time (about 360 seconds, i.e., ten times the cycle time of the RHF). To realize low cycle times, several blank pairs are processed at the same time, running through the furnace one after another with a fixed distance d_b. This distance cannot be changed during production and is considered as a constant in the following. This implies that the
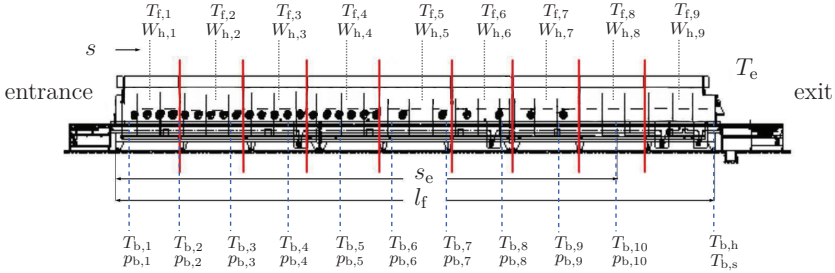
Figure 4.4.: Sketch of an RHF with 9 temperature segments indicated by the red lines. In addition, the location of the furnace temperatures $T_{f,m}$, heating powers $W_{h,m}$, and blank temperatures $T_{b,n}$ are shown.

number of blank pairs in the furnace is constant. At standard production setup, ten blank pairs are in the furnace at the same time. Furthermore, a variation of the transportation velocity v results into an inverse variation of the cycle time.

The RHF is sampled when the MPC is executed, i.e., when a blank pair reaches the exiting point s_e (see Fig. 4.4) and the partial hardening time $t_{ph,1}$ has to be determined. Due to the constant distance d_b, the positions of the blank pairs at the sampling incidents are constant and defined by:

$$s_{b,n} = s_e - (10 - n)d_b, \quad n \in \mathbb{N}_{10}. \tag{4.1}$$

If one or more slots are not occupied with a pair of blanks, the system is sampled when the blanks would have reached the defined positions.
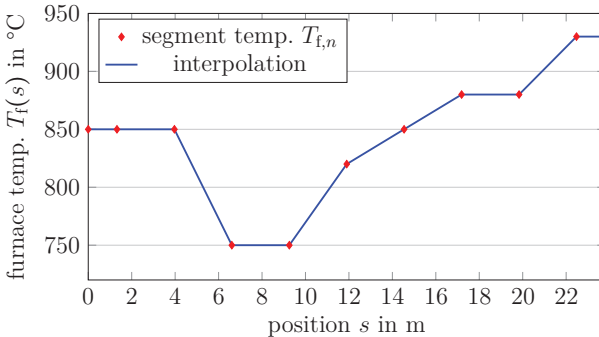
Figure 4.5.: Linearly interpolated exemplary temperature profile $T_f(s)$ [130].

Components of the Subsystem: The disturbances are given by the temperature of the blanks inserted into the furnace (which equals the ambient temperature T_a), by a disturbance influencing the transportation velocity, by the ambient temperature at the furnace exit T_e, and by the plant model mismatch of the furnace temperatures[1]:

$$w[k] := \begin{bmatrix} T_a[k] & w_v[k] & T_e[k] & \nu_{T,1:9}[k] \end{bmatrix}. \tag{4.2}$$

The output is defined by the blank temperatures in the soft and hard part, and a variable encoding whether a blank pair exits the RHF ($p_b[k] = 1$), or not ($p_b[k] = 0$):

$$y[k] := \begin{bmatrix} T_{b,h}[k] & T_{b,s}[k] & p_b[k] \end{bmatrix}. \tag{4.3}$$

The state is given by the furnace segment temperatures, the heat treatment time for nominal operation, the temperatures of all blank pairs processed in the furnace, the variables $p_{b,i}$ describing whether a blank pair is processed at the i-th position, and a state describing the plant model mismatch of the heat treatment time:

$$x[k] := \begin{bmatrix} T_{f,1:9}[k] & t_f[k] & T_{b,1:10}[k] & p_{b,1:10}[k] & \nu_t[k] \end{bmatrix}. \tag{4.4}$$

The heating power in the different furnace segments, the inverse of the transportation velocity, and the partial hardening time are the inputs of the systems:

$$u[k] := \begin{bmatrix} W_{h,1:9}[k] & v^{-1}[k] & t_{ph,1}[k] \end{bmatrix}. \tag{4.5}$$

Note that the temperature of the last blank pair in the furnace ($T_{b,10}[k]$) and the blank temperatures resulting from the partial hardening in $y[k]$ belong to the same blanks. This illustrates that the output $y[k]$ is a prediction of the part properties.

Modeling of the Markov Chain: Four different operation modes are considered:

1. **Normal operation:** The blanks pass through the furnace at the constant velocity $v[k]$ until they reach the exiting point s_e and partial hardening is initiated.

2. **Reversing operation:** If a set of blanks reaches the exiting point s_e but cannot exit the furnace (e.g., because the chamber furnace or the press stopped), all blanks are held at their current positions.

3. **Furnace emptying:** When the production is stopped intentionally, no new blanks are inserted to the furnace. All blanks leave the furnace one after another under standard production conditions.

4. **Standby:** The furnace is empty and waiting for new blanks.

---

[1]The plant model mismatch for the blank temperatures of the RHF are not considered explicitly, since these temperatures cannot be measured. These uncertainties propagate to the chamber furnace. Hence, they are considered in the model of the chamber furnace.
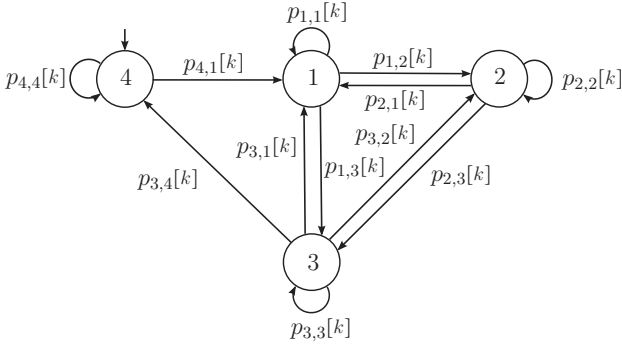
Figure 4.6.: Graph of the Markov chain $\mathcal{M}$ of the roller hearth furnaces.

These modes determine the $n_\theta = 4$ states of the Markov chain $\mathcal{M}$. The operation mode changes to normal operation as soon as a blank enters the process. If the furnace is in normal operation mode, it can jump into the furnace emptying mode until all blanks left the furnace, or to reversing mode if no blanks can exit the furnace. If the furnace is empty, it switches into the standby mode. The transition probabilities depend on the number of blanks in the furnace and, hence, vary with time. The graph of the Markov chain is depicted in Fig. 4.6.

The following subsections show how the nonlinear functions $f_i(\cdot)$ and $g_i(\cdot)$ can be determined for the different Markov states.

Modeling of the Furnace Temperatures[2]:   A continuous-time model of the furnace segment temperatures can be determined based on the heat flow rates into the segments (from neighboring segments and the heating) and the heat flow rates out of the segments (into neighboring segments, the blanks, and the environment). Due to the relatively low differences of the furnace segment temperatures and the good insulation of the furnace, thermal radiation is neglected. The resulting model has the following form for the $n \in \mathbb{N}_9$ segments:

$$
\begin{aligned}
\dot{T}_{\mathrm{f},n}(t) \;=\; c_n \big[& (1 - \alpha_{n,n-1} - \alpha_{n,n+1}) W_{\mathrm{h},n}(t) + \alpha_{n-1,n} W_{\mathrm{h},n-1}(t) + \alpha_{n+1,n} W_{\mathrm{h},n+1}(t) + \ldots \\
& \cdots + \lambda_{n+1,n}(T_{\mathrm{f},n+1}(t) - T_{\mathrm{f},n}(t)) + \lambda_{n-1,n}(T_{\mathrm{f},n-1}(t) - T_{\mathrm{f},n}(t)) + \ldots \\
& \cdots + \lambda_{\mathrm{a},n}(T_{\mathrm{a}}(t) - T_{\mathrm{f},n}(t)) - \dot{Q}_{\mathrm{b},n}(t) \big],
\end{aligned} \tag{4.6}
$$

where c_n is a constant depending on the mass, surface area, and thermal capacity of the n-th furnace segment. The continuous-time heating power $W_{\mathrm{h},n}(t)$ applied to the n-th segment also affects neighboring segments. The amount of these cross flows

---

[2]The formulation of the continuous-time furnace temperature model (4.6) and the determination of the model parameters is mainly based on the bachelor thesis [106] which was supervised by the author. The combination with the blank temperature models, the time-discretization, and the error determination (Table 4.1) are new contributions by the author.

Table 4.1.: Model errors of the discrete-time model of the furnace temperatures.

	$\nu_{T,1}$	$\nu_{T,2}$	$\nu_{T,3}$	$\nu_{T,4}$	$\nu_{T,5}$	$\nu_{T,6}$	$\nu_{T,7}$	$\nu_{T,8}$	$\nu_{T,9}$
mean abs. error in °C	1.6	1.9	0.9	0.8	0.9	1.1	0.7	1.2	2.5
max. abs. error in °C	8.9	8.1	3.8	6.4	5.6	4.2	3.8	8.3	8.6

is defined by the constants $\alpha_{\cdot,\cdot} \in [0,1)$. The constants $\lambda_{n,m} = \lambda_{m,n}$ are the thermal conductivities between the segments m and n, $\lambda_{a,n}$ is the thermal conductivity between the n-th segment and the environment, and $\dot{Q}_{b,n}(t)$ is a heat flow rate from the segment into the blanks. The heat flow $\dot{Q}_{b,n}(t)$ is a nonlinear function of the segment temperature $T_{f,n}(t)$ as well as the positions and temperatures of all blanks processed in the segment. To simplify the model, the heat flow is defined as:

$$\dot{Q}_{b,n}(t) = \lambda_{b,n}\big(T_{f,n}(t) - T'_{b,n}(t)\big), \tag{4.7}$$

where $T'_{b,n}(t)$ is the nominal blank temperature at the middle of the segment, which is assumed to be constant during one production cycle. By choosing the parameter $\lambda_{b,n}$ appropriately, the approximation error is neglectable. Since the heating power $W_{h,n}(t)$, the ambient temperature $T_a(t)$, and the differences $T_{f,n}(t) - T'_{b,n}(t)$ are nearly constant during one production cycle, a discrete-time furnace temperature model can be derived by ZOH discretization. The absolute discretization errors are below 0.03 °C (according to simulations) and, hence, can be neglected.

In the discrete-time model, the temperatures $T'_{b,n}[k]$ are determined by linear interpolation of the blank temperatures $T_{b,n}[k]$ at the sampling positions. The heat flows $\dot{Q}_{b,n}$ are only present if blanks are processed at the corresponding positions, i.e., $p_{b,n}[k] = 1$. Hence, the corresponding temperature differences are multiplied with $p_{b,n}[k]$, and a nonlinear discrete-time furnace temperature model results.

The plant model mismatch of the resulting discrete-time model is described by $\nu_{T,1:9}[k]$. It is considered as an additive disturbance in the model. The maximum and mean values are determined based on temperature recordings of the production line. The results are shown in Table 4.1. It can be seen that the mean absolute error is quite low considering the absolute values of the furnace temperatures. Commonly, the deviation of the furnace temperature from its reference should be bounded by ±30 °C. Hence, the accuracy of the model is considered high enough for an MPC application. In addition, the furnace temperatures are controlled by two point hysteresis controllers currently. The furnace temperature variations, resulting from the switching of the heating, cause the maximum errors shown in Table 4.1. If the furnace would be controlled by the MPC proposed in this theses, the heating power variations would be smaller. Hence, the errors in the temperature prediction will most likely decrease.

Modeling of the Heat Treatment Time: The nominal heat treatment time is determined by the quotient of furnace length l_f and transportation velocity $v[k]$. Due

to contamination effects of the transportation rollers, the effective transportation velocity may differ slightly from the specified velocity. The resulting plant model mismatch is described by $\nu_t[k]$, which gives the model for the heat treatment time:

$$t_f[k] = l_f v^{-1}[k] + \nu_t[k]. \tag{4.8}$$

The contamination effects are characterized by very large time constants. Hence, the process $\nu_t[k]$ is assumed to be constant during the prediction horizon, except for a noise process $w_v[k]$:

$$\nu_t[k+1] = \nu_t[k] + w_v[k]. \tag{4.9}$$

Modeling of the Blank Temperature: A model for the blank temperature dynamics as a function of the furnace temperature profile $T_f(s)$ and the transportation velocity $v[k]$ was developed in [130] by the author. In [130], a nonlinear differential equation system was proposed to describe the blank temperature $T_b(t)$ and the position $s(t)$:

$$
\begin{aligned}
\dot{T}_b(t) &= c(T_b(t)) \left[\varepsilon(T_b(t)) \, \sigma \left(T_f^4(s(t), t) - T_b^4(t) \right) + \lambda_{f,b} \left(T_f(s(t), t) - T_b(t) \right) \right], &(4.10\text{a}) \\
\dot{s}(t) &= v(t), &(4.10\text{b}) \\
T_b(t_0) &= T_{b,0}, \qquad s(t_0) = s_0. &(4.10\text{c})
\end{aligned}
$$

In (4.10), $c(T_b(t))$ is a parameter that depends on the thickness and the density of the blanks, their thermal capacity (which is a function of $T_b(t)$), and the geometry of the furnace. $\varepsilon(T_b(t))$ is the emissivity of the blanks' coating, and σ is the Stefan-Boltzmann constant. The blank cooling at ambient temperature during the partial hardening can be modeled by an adapted version of (4.10a):

$$
\begin{aligned}
\dot{T}_b(t) &= c(T_b(t)) \left[\varepsilon(T_b(t)) \, \sigma \left(T_a^4(t) - T_b^4(t) \right) + \lambda_{a,b} \left(T_a(t) - T_b(t) \right) \right] &(4.11\text{a}) \\
T_b(t_0) &= T_{b,0}. &(4.11\text{b})
\end{aligned}
$$

The parameters c and λ have been determined by drag measurements and model identification as described in [130].

Definition 4.1. *Let the blank temperature $T_b(t)$ with $t \in [t_0, t_0 + t']$ be the solution of the initial value problem (4.10) for a constant velocity $v(t) = v$ and constant furnace temperature profile $T_f(s)$. Furthermore, let the final blank temperature $T_b(t_0 + t')$ be given by a function $\mathcal{F}(T_{f,:}, T_{b,0}, v, s_0, t') := T_b(t_0 + t')$. Here, $T_{f,:}$ denotes all furnace temperatures $T_{f,n}$. The solution $\mathcal{F}_a(T_a, T_{b,0}, t') := T_b(t_0 + t')$ of the initial value problem (4.11) is defined analogously.*

The function $\mathcal{F}(\cdot)$ describes the temperature of the blank T_b after a certain time t' in the RHF for a constant furnace temperature profile, transportation velocity, and initial temperature as well as initial position. For a fixed t', this function is a discrete-time representation of the initial value problem (4.10) and can be used to formulate

the discrete-time dynamics of the blank temperature in the RHF. The function $\mathcal{F}_a(\cdot)$ can be used to define the discrete-time dynamics of the blank temperature during cooling at air. If no blanks are present, the corresponding temperatures are set to 0, i.e., $\mathcal{F}(\cdot, 0, \cdot, \cdot, \cdot) = 0$ and $\mathcal{F}_a(\cdot, 0, \cdot) = 0$.

In general, no algebraic expressions exist for $\mathcal{F}(\cdot)$ and $\mathcal{F}_a(\cdot)$. However, the resulting functions can be approximated locally by linear functions. To derive an accurate discrete-time model, the values of the functions $\mathcal{F}(\cdot)$ and $\mathcal{F}_a(\cdot)$ are determined by solving the initial value problems numerically offline at a grid of sample points for $T_{b,n}, T_a, v$, and $t_{ph,1}$. Then, the actual values of $\mathcal{F}(\cdot)$ and $\mathcal{F}_a(\cdot)$ for the current values of $T_{b,n}[k], T_a[k], v[k]$, and $t_{ph,1}[k]$ can be calculated by linear interpolation. The absolute error between linear interpolation and numerical solution is less than 1 °C. This is accurate enough for the formulation of the discrete-time model.

Employing Definition 4.1, the discrete-time nonlinear dynamics for the blank temperatures can be formulated for the different Markov states as follows:

- **Normal Operation Mode ($\theta_k = 1$):** The blank pairs 1 - 9 are transported from $s_{b,n}$ to $s_{b,n+1}$. A new blank pair enters the furnace and is transported to $s_{b,1}$, while the 10-th blank pair is partially hardened and leaves the furnace. In general, $s_{b,1} < d_b$, i.e., the blanks enter the furnace at a certain time during the sampling interval and are heated for the duration of $s_{b,1}/v[k]$ starting at ambient temperature $T_a[k]$. The blank pairs $n \in \mathbb{N}_9$ are heated during the transport from $s_{b,n}$ to $s_{b,n+1}$. All in all, the blank temperatures are defined by:

$$T_{b,n}[k+1] = \begin{cases} \mathcal{F}\left(T_{f,:}[k], T_a[k], v[k], 0, \frac{s_{b,1}}{v[k]}\right) & \mid n = 1 \\ \mathcal{F}\left(T_{f,:}[k], T_{b,n-1}[k], v[k], s_{b,n-1}, \Delta t_f\right) & \mid n \in \{2, \dots, 10\} \end{cases}. \quad (4.12)$$

- **Reversing Operation Mode ($\theta_k = 2$):** If the blanks cannot exit the furnace due to errors in the successive production steps, the blanks stay at their sampling positions and the blank temperatures evolve according to:

$$T_{b,n}[k+1] = \mathcal{F}\left(T_{f,:}[k], T_{b,n}[k], 0, s_{b,n}, \Delta t_f\right), \quad \forall n \in \mathbb{N}_{10}. \quad (4.13)$$

- **Furnace Emptying Mode ($\theta_k = 3$):** This mode is similar to the normal production, but no new blanks are inserted to the furnace. However, the blank temperatures $T_{b,n}$ are simulated as if there were blanks at all positions in order to minimize linearization errors. The blank temperatures are defined as in (4.12).

- **Standby Mode ($\theta_k = 4$):** In this mode, no blanks are in the furnace. However, the blank temperatures $T_{b,n}$ are also simulated by (4.12) as if there were blanks at all positions to minimize linearization errors at restart of the production.

The partial hardening process is started by transporting the blank pair at a high exiting velocity v_e until the end of the furnace is reached ($s = l_f$) and the soft part has left the furnace. Until that point, the temperature of the blank pair is homogeneous:

$$T_b'[k] = \mathcal{F}\left(T_{f,:}[k], T_{b,10}[k], v_e, s_{b,10}, \frac{l_f - s_{b,10}}{v_e}\right). \quad (4.14)$$

Then, the blank pair stops ($v = 0$) for the time $t_{\mathrm{ph},1}[k]$, and the soft part cools down at air while the hard part remains in the RHF. The two different temperatures are:

$$
\begin{aligned}
T_{\mathrm{b,h}}[k] &= \mathcal{F}\left(T_{\mathrm{f,:}}[k], T_{\mathrm{b}}'[k], 0, l_{\mathrm{f}}, t_{\mathrm{ph},1}[k]\right), &\quad (4.15\mathrm{a})\\
T_{\mathrm{b,s}}[k] &= \mathcal{F}_{\mathrm{a}}\left(T_{\mathrm{e}}[k], T_{\mathrm{b}}'[k], t_{\mathrm{ph},1}[k]\right). &\quad (4.15\mathrm{b})
\end{aligned}
$$

Modeling of the Blank Positions: The variables $p_{\mathrm{b},n}[k]$ encode whether a blank is at the n-th position ($p_{\mathrm{b},n}[k] = 1$) or not ($p_{\mathrm{b},n}[k] = 0$). Blanks only enter the furnace for $\theta_k = 1$. Furthermore, the blanks stay at their positions in reversing mode. They are transported to the next position in normal operation or if the furnace is emptied:

$$
p_{\mathrm{b},1}[k+1] = \begin{cases} 1 & \mid \theta_k = 1 \\ p_{\mathrm{b},1}[k] & \mid \theta_k = 2 \\ 0 & \mid \theta_k \in \{3,4\} \end{cases}
\qquad
p_{\mathrm{b},n}[k+1] = \begin{cases} p_{\mathrm{b},n-1}[k] & \mid \theta_k \in \{1,3\} \\ p_{\mathrm{b},n}[k] & \mid \theta_k = 2 \\ 0 & \mid \theta_k = 4 \end{cases}
\qquad (4.16)
$$

for all $n \in \{2, \dots, 10\}$. The blank exiting probability is given by the product of the probability that a blank is at the end of the furnace $p_{\mathrm{b},10}[k]$ and the probability that the furnace does not break down in the current cycle $(1 - \lambda_{\mathrm{f}})$. Hence, it follows:

$$
p_{\mathrm{b}}[k] = \begin{cases} p_{\mathrm{b},10}[k] \cdot (1 - \lambda_{\mathrm{f}}) & \mid \theta_k \in \{1,3\} \\ 0 & \mid \theta_k \in \{2,4\} \end{cases}.
\qquad (4.17)
$$

During the normal simulation of the RHF, the variables $p_{\mathrm{b},i}[k]$ are binary variables. However, for the prediction of the subsystem behavior based on the transition probabilities, the predicted values of these variables become probabilities that a blank pair is processed at the corresponding position or is exiting the furnace. Thus, the variables may take values in $[0, 1]$.

4.2.2. Chamber Furnace

The chamber furnace is a small RHF with one temperature T_{f}. A sketch of the furnace and the localization of the temperatures is shown in Fig. 4.7. In each production cycle, only one blank pair is processed by the furnace. While the blanks leave the furnace, the temperature of both blanks is measured thermographically. This is the first point where the temperature of the blanks is measured and differences between the right and left blank can be detected. Hence, the output $y[k]$ now describes the temperatures of the left and right blank separately as well as the probability $p_{\mathrm{b}}[k]$ that a blank pair exits the furnace:

$$
y[k] := \begin{bmatrix} T_{\mathrm{b,h},1}[k] & T_{\mathrm{b,s},1}[k] & T_{\mathrm{b,h},2}[k] & T_{\mathrm{b,s},2}[k] & p_{\mathrm{b}}[k]. \end{bmatrix},
\qquad (4.18)
$$

The index 1 / 2 indicates whether the temperature belongs to the left or the right blank (cf. Fig. 4.7). The disturbances of the subsystem are given by the temperatures
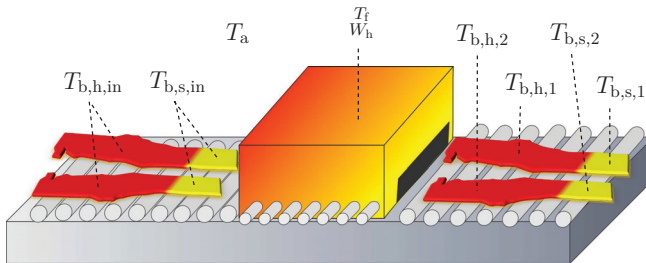
Figure 4.7.: Sketch of the chamber furnace including the localization of the different temperatures and the heating power.

of the blank pair entering the furnace, which equals the output $y[k]$ of the RHF, the ambient temperature T_a, the plant model mismatch process for the furnace temperature, and the plant model mismatch processes for the blank temperatures:

$$w[k] := \begin{bmatrix} T_{b,h,in}[k] & T_{b,s,in}[k] & T_a[k] & \nu_T[k] & \nu_{h,1}[k] & \nu_{s,1}[k] & \nu_{h,2}[k] & \nu_{s,2}[k] \end{bmatrix}. \quad (4.19)$$

The state of the subsystem is given by the furnace temperature:

$$x[k] := T_f[k]. \quad (4.20)$$

The inputs are the heating power, the furnace time, and the partial hardening time:

$$u[k] := \begin{bmatrix} W_h[k] & t_f[k] & t_{ph,2}[k] \end{bmatrix}. \quad (4.21)$$

Modeling of the Markov Chain: Two operation modes are considered:

1. The chamber furnace is processing a blank pair.

2. No blanks are received.

The graph of the Markov chain is depicted in Fig. 4.8. Since the transition probabilities depend also on the fault rates of all other subsystems and the number of blanks in the RHF, the transition probabilities are time-varying. It may also occur that the chamber furnace has an error and cannot eject the blanks to the robot. In this case, the processed blanks are scrap. The blank pair is removed from the process, and the robot and the press idle for one cycle. Hence, the corresponding failure probabilities have to be considered by the other subsystems.
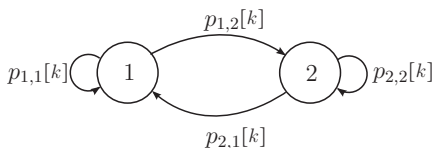
Figure 4.8.: Graph of the Markov chain $\mathcal{M}$ of the camber furnace.

Modeling of the Furnace Temperature: The furnace temperature can be modeled similar to that of the RHF, but with only one segment. For the derivation of the discrete-time model, the continuous-time dynamics of the furnace temperature $T_f(t)$ are considered first. To this end, the continuous-time representation of the heating power $W_h(t)$, the blank temperature $T_b(t)$, and the ambient temperature $T_a(t)$ are used. By evaluation of the heat flow in and out of the furnace, the following differential equation results for $\theta_k = 1$ (cf. [106]):

$$\dot{T}_f(t) \;=\; c\left[\lambda_{a,1}\big(T_a(t) - T_f(t)\big) + \lambda_b\big(T_b(t) - T_f(t)\big) + W_h(t)\right]. \qquad (4.22)$$

If no blanks are processed ($\theta_k = 2$), it holds:

$$\dot{T}_f(t) \;=\; c\left[\lambda_{a,2}\big(T_a(t) - T_f(t)\big) + W_h(t)\right]. \qquad (4.23)$$

The Markov state θ_k influences the thermal conductivity λ_a, since the furnace gates are opened for $\theta_k = 1$ only. The discrete-time furnace temperature equation is determined by ZOH discretization. The resulting discretization errors are neglectable. The temperature $T_{b,h,in}[k]$ of the hot part is used in the discrete-time model to describe the heat flow into the blanks, since the hot part covers the most area of the blank and remains longer in the furnace. The model error of the furnace temperature is described by the additive, zero-mean disturbance $\nu_T[k]$. The mean and the maximum value of the absolute model errors $|\nu_T[k]|$ are 0.8 °C and 2.5 °C, respectively. As for the RHF, this is accurate enough for an MPC application.

Modeling of the Outputs: The blank temperatures can be modeled based on the concepts introduced for the RHF. The blank temperature dynamics are (cf. [130]):

$$\dot{T}_b(t) \;=\; c\big(T_b(t)\big)\left[\varepsilon\big(T_b(t)\big)\,\sigma\,\big(T_f^4(t) - T_b^4(t)\big) + \lambda_{f,b}\,\big(T_f(t) - T_b(t)\big)\right]. \qquad (4.24)$$

These dynamics differ from those for cooling at air (cf. (4.11)) only in the use of the furnace temperature T_f instead of the ambient temperature and a different heat transfer coefficient $\lambda_{f,b}$. Hence, the dynamics can be discretized over time as shown for the cooling at air. Let the corresponding solution of the initial value problem be given by the function $\mathcal{F}_{CF}(T_f, T_{b,0}, t')$, which is defined analogously to $\mathcal{F}_a(T_a, T_{b,0}, t')$. Then, the blank temperatures at the end of the furnace are:

$$T'_{b,h}[k] = \mathcal{F}_{CF}\big(T_f[k], T_{b,h,in}[k], t_f\big), \qquad T'_{b,s}[k] = \mathcal{F}_{CF}\big(T_f[k], T_{b,s,in}[k], t_f\big). \qquad (4.25)$$

Theoretically, the left and the right blank should have the same temperature. In practice, the values may differ slightly in normal operation or significantly if faults occur in the furnace heating. Since these effects cannot be modeled in advance, disturbance models $\nu_y[k]$ are used to approximate these effects for the hard and soft part of the left and right blank. For $\theta_k = 1$ and $n \in \{1, 2\}$, one gets:

$$T_{b,h,n}[k] \;=\; \mathcal{F}_{CF}\big(T_f[k], T'_{b,h}[k], t_{ph,2}[k]\big) + \nu_{h,n}[k], \qquad (4.26a)$$
$$T_{b,s,n}[k] \;=\; \mathcal{F}_a\big(T_a[k], T'_{b,s}[k], t_{ph,2}[k]\big) + \nu_{s,n}[k]. \qquad (4.26b)$$

For $\theta_k = 2$, the temperatures are set to 0. The processes $\nu_{\mathrm{h},n}[k]$ and $\nu_{\mathrm{s},n}[k]$ are modeled by second order ARIMA models defined in Section 3.2.2. To account for faults and disturbances from the two different RHFs, separate models are used for the parts received from the first or the second RHF.

If the furnace is processing a blank pair, the probability that a blank pair exits the chamber furnace equals the probability that the furnace does not break down, i.e., $p_{\mathrm{b}}[k] = 1 - \lambda_{\mathrm{f}}$. Otherwise it is zero.

4.2.3. Transfer Robot

The robot transfers the blank pair from the exit of the chamber furnace to the press. During the transfer time t_{t}, the blanks cool at the air. The blank temperatures are measured directly after these have been inserted into the pressing tool.

Components of the Subsystem: The disturbances are defined by the blank temperatures of the hard and the soft part of both blanks, the ambient temperature, and the plant model mismatch processes:

$$w[k] := \begin{bmatrix} T_{\mathrm{b,h,in},1}[k] & \dots & T_{\mathrm{b,s,in},2}[k] & T_{\mathrm{a}}[k] & \nu_{\mathrm{h},1}[k] & \dots & \nu_{\mathrm{s},2}[k] \end{bmatrix}. \qquad (4.27)$$

The index 1 / 2 indicates whether the temperature belongs to the left or the right blank. The outputs are defined analogously to that of the chamber furnace:

$$y[k] := \begin{bmatrix} T_{\mathrm{b,h},1}[k] & T_{\mathrm{b,s},1}[k] & T_{\mathrm{b,h},2}[k] & T_{\mathrm{b,s},2}[k] & p_{\mathrm{b}}[k] \end{bmatrix}. \qquad (4.28)$$

The model does not require any states. The input is the transfer time:

$$u[k] := t_{\mathrm{t}}[k]. \qquad (4.29)$$

Markov Chain: Two operation modes are considered:

1. A blank pair is transferred to the press.

2. No blanks are transferred.

The transition probabilities may be time-varying due to the dependency on the Markov states in the other subsystems.

Modeling of the Outputs: The cooling of the hard and the soft parts of the blanks can be modeled independently due to the low sheet thickness for $\theta_k = 1$:

$$\begin{aligned}
T_{\mathrm{b,h},n}[k] &= \mathcal{F}_{\mathrm{a}}\left(T_{\mathrm{a}}[k], T_{\mathrm{b,h,in},n}[k], t_{\mathrm{t}}[k]\right) + \nu_{\mathrm{h},n}[k], & n \in \{1,2\}, & \qquad (4.30\mathrm{a}) \\
T_{\mathrm{b,s},n}[k] &= \mathcal{F}_{\mathrm{a}}\left(T_{\mathrm{a}}[k], T_{\mathrm{b,s,in},n}[k], t_{\mathrm{t}}[k]\right) + \nu_{\mathrm{s},n}[k], & n \in \{1,2\}. & \qquad (4.30\mathrm{b})
\end{aligned}$$

For $\theta_k = 2$, the temperatures are set to 0. The processes $\nu_{\mathrm{h},n}[k]$ and $\nu_{\mathrm{s},n}[k]$ are modeled analogously to that of the chamber furnace. If the robot is transferring a blank pair, the probability $p_{\mathrm{b}}[k]$ is given by the probability that the robot does not break down, i.e., $1 - \lambda_{\mathrm{f}}$. Otherwise it is zero.

4.2.4. Press

After the blank pair has been transferred into the pressing tool, the hydraulic press closes the tool, builds up the pressing force $F[k]$, and holds this force for a time t_p. During that time, the blanks are quenched by two water cooled pressing tools. The hot stamping process is finished when the tools open again. A sketch of the process and the localization of the corresponding quantities is shown in Fig. 4.9. The temperatures of the formed parts are measured by thermographic cameras (cf. Fig. 4.3 on page 52). The temperature evolution of the blanks is mainly influenced by the temperature of the pressing tools, the pressing time t_p, and the pressing force $F[k]$. The temperatures of the left and the right tool are influenced by the cooling agent flows $\dot{V}_\mathrm{l}[k]$ and $\dot{V}_\mathrm{r}[k]$, the inlet temperature of cooling agent $T_\mathrm{c}[k]$, and the pressing force $F[k]$. In [116], a JMS that models the tool and blank temperatures of a hot stamping test line was proposed. This modeling concept is adopted here.

Components of the Subsystem: The disturbances are defined by the temperatures of the blank pairs before pressing, the ambient temperature, the cooling agent temperature, and the model mismatch processes for all tool and blank temperatures:

$$w[k] \coloneqq \begin{bmatrix} T_{\mathrm{b,h,in,1}}[k] & \dots & T_{\mathrm{b,s,in,2}}[k] & T_\mathrm{a}[k] & T_\mathrm{c}[k] & \nu_{\mathrm{x,1:16}}[k] & \nu_{\mathrm{y,1:8}}[k] \end{bmatrix}. \quad (4.31)$$

The output of the pressing process is defined by the temperatures at different positions of the blanks after pressing. To have moderate system dimensions, the
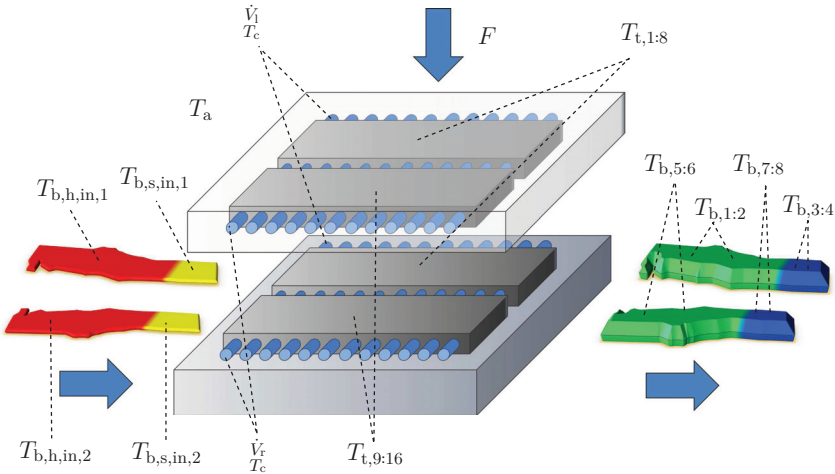
Figure 4.9.: Sketch of the press and the two tools including the localization of the different temperatures, the cooling agent flows, and the pressure force.

temperatures at a few significant points of the formed blanks, such as hot spots and cold spots, are considered. Here, eight temperatures are used:

$$y[k] := \begin{bmatrix} T_{b,1}[k] & \dots & T_{b,8}[k] \end{bmatrix}. \tag{4.32}$$

$T_{b,1}[k]$ and $T_{b,2}[k]$ are located in the hard part of the left blank, $T_{b,3}[k]$ and $T_{b,4}[k]$ in the soft part of the left blank, $T_{b,5}[k]$ and $T_{b,6}[k]$ in the hard part of the right blank, and $T_{b,7}[k]$ and $T_{b,8}[k]$ in the soft part of the right blank (cf. Fig. 4.9).

The cooling agent flows $\dot{V}_l[k]$ and $\dot{V}_r[k]$ as well as the pressing force $F[k]$ can be changed by the controller during production:

$$u[k] := \begin{bmatrix} \dot{V}_l[k] & \dot{V}_r[k] & F[k] \end{bmatrix}. \tag{4.33}$$

The change of the cooling agent flows is delayed by one time step due to the long pipes of the cooling system. This effect is modeled with two states $x_{Vl}[k]$ and $x_{Vr}[k]$. Furthermore, the temperature distribution of the tool is considered as a set of states of the production unit. The tool temperatures can be monitored at 16 different points in the tools $(T_{t,1}, \dots, T_{t,16})$ at most by the programmable logic controller (PLC) of the press. Hence, the state vector is defined by 16 tool temperatures (sampled at the moment when the pressing tool closes) and the delay states:

$$x[k] := \begin{bmatrix} T_{t,1}[k] & \dots & T_{t,16}[k] & x_{Vl}[k] & x_{Vr}[k] \end{bmatrix}. \tag{4.34}$$

The first eight tool temperatures are located in the left tool, and the last eight temperatures in the right tool.

Markov Chain: Analogous to the chamber furnace and the robot, two operation modes are considered (1. Normal operation; 2. No blanks are processed). The transition probabilities may be time-varying due to the dependency on the Markov states of the other subsystems.

Modeling of Tool and Blank Temperature: During the pressing operation, the heat transfer is caused mainly by thermal conduction from the blanks to the tool, i.e., radiation can be neglected [94]. Thermal conduction can be described by linear differential equations. However, the heat transfer coefficients between the blanks and the tool as well as between the tool and the cooling agent are nonlinear functions of the inputs. Considering the thermal decoupling of the left and right tool, the overall discrete-time model can be defined as follows:

$$\begin{bmatrix} T_{t,1:8}[k+1] \\ T_{t,9:16}[k+1] \\ T_{b,1:2}[k] \\ T_{b,3:4}[k] \\ T_{b,5:6}[k] \\ T_{b,7:8}[k] \end{bmatrix} = \begin{bmatrix} * & 0 & * & * & 0 & 0 & * & * \\ 0 & * & 0 & 0 & * & * & * & * \\ * & 0 & * & 0 & 0 & 0 & * & * \\ * & 0 & 0 & * & 0 & 0 & * & * \\ 0 & * & 0 & 0 & * & 0 & * & * \\ 0 & * & 0 & 0 & 0 & * & * & * \end{bmatrix} \begin{bmatrix} T_{t,1:8}[k] \\ T_{t,9:16}[k] \\ T_{b,h,in,1}[k] \\ T_{b,s,in,1}[k] \\ T_{b,h,in,2}[k] \\ T_{b,s,in,2}[k] \\ T_c[k] \\ T_a[k] \end{bmatrix} + \begin{bmatrix} \nu_{x,1:8}[k] \\ \nu_{x,9:16}[k] \\ \nu_{y,1:2}[k] \\ \nu_{y,3:4}[k] \\ \nu_{y,5:6}[k] \\ \nu_{y,7:8}[k] \end{bmatrix}. \tag{4.35}$$

The $*$ indicates parameter matrices that are nonlinear functions of the pressing force and the cooling agent flow. Hence, the overall system is a nonlinear JMS. The dashed lines separate the tool temperatures, blank temperatures, and disturbances. These functions can be determined based on measurements of the tool and blank temperatures (see [116] for details). The plant model mismatch processes $\nu_x[k]$ and $\nu_y[k]$ are modeled by 24 different second order ARIMA models. For details, see Section 3.2.2.

At the time this thesis was written, the thermocouples that measure the tool temperatures had not been implemented completely to the tool of the production line. To demonstrate and test the proposed MPC approaches with a complete model of a hot stamping line nevertheless, the measurements of a test tool are used to determine realistic model parameters. To this end, a test tool made of steel with a high heat transfer coefficient is used (cf. [112, 116]). This model can be used to evaluate the properties of the proposed MPC approaches as well because the dimensions and model structure are identical to that of the serial production line.

4.3. Modeling of the Overall Production Process

The model of the whole production line is determined by the coupling of the five subsystems. The product flow is determined by the alternating sampling of both RHFs with the double cycle time of all subsequent processes. The time-varying graph $\mathfrak{G}[k_b] = (\mathfrak{N}, \mathfrak{E}[k_b])$ that represents the product flow at time k_b is given by:

$$
\begin{align}
\mathfrak{N} &= \{1, \dots, 5\} \tag{4.36a}\\
\mathfrak{E}[k_b] &= \{((k_b + 1) \bmod 2) + 1, 3), (3, 4), (4, 5)\}. \tag{4.36b}
\end{align}
$$

The modulus is used to formalize that the two RHFs (subsystems 1 and 2) deliver blanks alternatingly to the chamber furnace (subsystem number 3). A detailed description of the timing of the interaction is provided in Chapter 13 together with the setup of the distributed MPC.

The continuous dynamics of the subsystems are coupled by the temperatures of the processed blanks. Hence, every time a set of blanks is transported to the subsequent subsystem (according to $\mathfrak{E}[k_b]$), the temperatures in the output vector $y[k]$ are passed to the temperatures in the disturbance vector of the subsequent subsystem. This concept is shown in Fig. 4.10, where the continuous temperature evolution of the blank temperatures in the hard and the soft part are shown. The sampling incidents of the corresponding quantities are indicated by dotted lines.

In addition to the dynamic couplings, the discrete events of the subsystems are coupled. A fault in the chamber furnace, robot, or press results in a halt of all other subsystems. If one of the RHFs breaks down, the processing will only take place in every second base cycle. This results in a cyclic switching of the Markov states of the chamber furnace, robot, and press. Obviously, this affects the continuous dynamics of all subsystems. A detailed description of the calculation of all transition
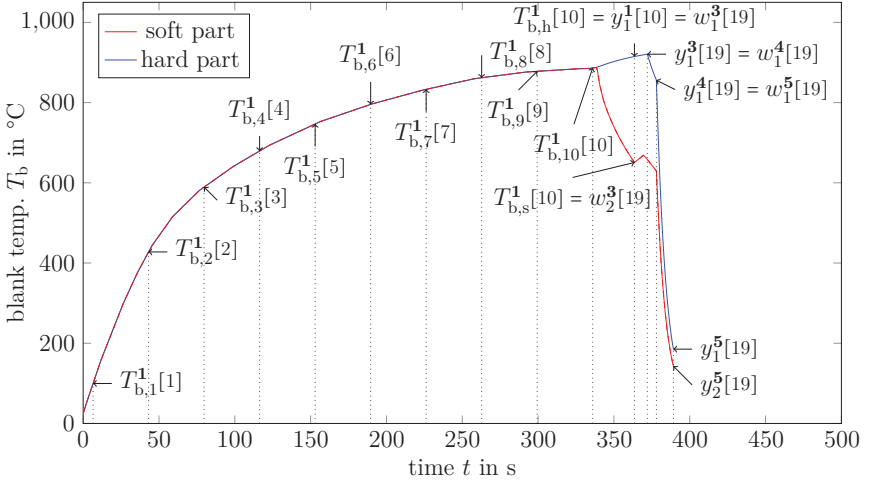
Figure 4.10.: Sketch of the temperature evolution of the hard part (blue) and the soft part (red) of the blanks during all production stages. The dotted lines indicate the different sampling instants. Subsystem 1 is the RHF, subsystem 3 the chamber furnace, subsystem 4 the transfer robot, and subsystem 5 the press.

probabilities can be found in Appendix C. In short: during normal operation, the transition probabilities of the RHFs are given by the mean fault and repair rates of the complete production line (for details on the calculation see Section 3.2.3). Once a fault occurs in one of the subsystems, the corresponding time-dependent repair rate is used instead. The prediction of $\mathrm{E}(p_b[j])$ in the RHFs provides the probabilities that blanks are transferred to the chamber furnace. These probabilities can be used directly as transition probabilities for the chamber furnace. Considering the local break down probability of the chamber furnace, the probabilities that blanks are handed over to the robot $(\mathrm{E}(p_b^3[j]))$ can be predicted. These again can be used for the transition probabilities of the Markov chain of the robot, and so forth. Hence, the overall probabilities of a production line stop are predicted mainly in the RHFs and then propagated through the subsystems based on the output predictions.

4.4. Conclusions

In this part, a modeling framework for multistage manufacturing processes, that are subject to uncertainties, has been presented. The production units are modeled by JMS where the transitions of the Markov chain model abrupt disturbances and faults. The plant model mismatch resulting from continuous disturbances, such as

Table 4.2.: Dimensions of the JMS and cycle times of the subsystems. The plant model mismatch processes are included in the disturbance dimensions.

subsystem	n_x	n_u	n_w	n_y	n_θ	t_c in s
1/2 - RHF	31	11	12	3	4	36
3 - chamber furnace	1	3	7	5	2	18
4 - robot	0	1	9	5	2	18
5 - press	18	3	30	8	2	18

wear, is modeled by auto regressive disturbance models. The subsystems are coupled by the properties of the processed products based on a coupling graph. A large class of manufacturing processes for piece goods can be modeled with this framework, since no restricting assumptions are made on the dynamics of the processes and very general disturbance and fault distributions may be considered. Only buffers or cyclic coupling graphs are not considered. In this chapter, the whole modeling procedure has been illustrated for a hot stamping production line. The JMS of all subsystems contain nonlinear state and output dynamics. The dimensions and the cycle times of the subsystems are summarized in Table 4.2. All in all, this illustrates the applicability of the modeling framework to real manufacturing processes.

Based on the modeling framework, the requirements for the control can be deduced. The main goal is to keep the product properties and the states of the whole manufacturing system, i.e., of all production units, robustly within predefined bounds considering the Markov state switching and the continuous disturbances. To this end, general linear constraints on inputs, states, and outputs have to be considered. If possible, PMSS should be guaranteed. However, particularly when faults occur or continuous disturbances affect the system, minimization of the costs (which might represent real economic costs) is preferred to PMSS as long as the constraints are satisfied. Due to effects, such as wear, time-varying system behavior needs to be considered. Last but not least, the controller has to be applicable to larger system dimensions and relatively short cycle times. All in all, a robust control for time-varying nonlinear JMS considering arbitrary reference signals and general linear constraints is needed. For the application to real systems, such as the hot stamping process, subsystems modeled by JMS with up to 75 states, 10 Markov states, and 20 inputs have to considered. The online computation time should not exceed one second for these dimensions. Based on these requirements, different MPC setups are proposed in the following parts.

Part III.

Model Predictive Control for Jump Markov Systems

5. Literature Review and Problem Setup

As stated in the previous chapter, the main challenge of controlling the manufacturing system is that state and input constraints have to be satisfied robustly. In addition, (economic) costs shall be minimized considering the time-variant dynamics of the subsystems. The most common control approach capable of handling constraints, general cost formulations, and time-variant dynamics is MPC, see [71, 88, 93, 107] and Section 2.5. Hence, an approach for the design of a distributed MPC scheme for coupled JMS is considered in this thesis. First, only a single production unit (and not the coupled production system) is considered in this part. Subsequently, concepts for a distributed MPC of the coupled manufacturing system are presented in the Chapters 10 and 11. In this chapter, the state of the art in MPC for JMS is reviewed, and the problem setup is deduced from the requirements for the control (cf. Section 4.4) and the literature review.

5.1. State of the Art in MPC for JMS

This section presents the state of the art in MPC for discrete-time JMS. The literature is reviewed separately for the linear and the nonlinear case. Approaches to MPC for distributed JMS are discussed in Section 10.1.

Jump Markov Linear Systems

MPC for JMLS has attracted increasing attention in the last decades. Approaches to finite horizon optimal quadratic control (which can be regarded as the most simple form of MPC) date back to the late 1980s [33]. Subsequently, finite horizon MPC approaches determining sets of linear state feedback control laws for the whole prediction horizon N have been proposed both for the case with known and unknown Markov state (cf., e.g., [40, 136]). It has been shown that this approach guarantees MSS, if the horizon N is large enough [37]. However, these approaches do not consider state or input constraints.

Recently, many different MPC approaches have been proposed that consider various constraint formulations and guarantee robustness to various uncertainties. First of all, these approaches can be categorized by the type of **uncertainties** considered:

1. First of all, some approaches do only consider the uncertainties induced by the Markov state θ_k (e.g. [15, 55, 56]).

2. The majority of approaches considers additive zero-mean disturbances [17, 18, 32, 83, 137, 138, 143]. However, to the best of the author's knowledge, there is no approach considering additive disturbances with a non-zero mean.

3. Multiplicative disturbances are considered in [144].

4. Unknown, uncertain, or time-variant parameters of the system matrices A_i and B_i can be described by matrix polytopes. Robust MPC approaches that consider this type of uncertain JMLS are proposed in, e.g., [29, 84, 103, 149, 150].

5. Finally, robustness against partially unknown transition probabilities (cf. [143, 149]) and uncertain transition probability matrices described by a matrix polytope (cf. [29, 84, 103]) are considered.

Classical deterministic MPC formulations, as described in Section 2.5, cannot be employed because of the stochastic nature of JMLS and the uncertainties. Stochastic formulations or approximations are needed to formulate the MPC problem. The common approach is to replace the deterministic **cost function** (cf. (2.36a) on page 26) by its expected value [137]:

$$\mathrm{E}\left(\sum_{j=1}^{N} \|x[j]\|_{Q_{\theta_k}}^2 + \sum_{j=0}^{N-1} \|u[j]\|_{R_{\theta_k}}^2\right). \tag{5.1}$$

An infinite horizon version of (5.1) can be minimized using a state feedback control law for pre-stabilization and a Lyapunov function for cost prediction [31, 32, 83]. To avoid the explicit prediction of the expected costs, approaches that approximate the costs have been proposed. In scenario based approaches, only a subset of all possible Markov state trajectories is used for prediction. The trajectories are selected by certain criteria, e.g., most likely Markov state trajectories (cf. [15, 56, 146]). In contrast, particle based approaches approximate the system behavior and the resulting costs based on n_p random Markov state and disturbance trajectories (cf. [17, 18]). Instead, many robust MPC approaches (e.g., [29, 84, 103, 149, 150]) aim at minimizing the worst-case infinite horizon cost by solving a min-max-optimization problem.

In addition to different cost functions, various **constraint** formulations are considered. These can be divided into three groups:

1. Different variants of hard constraint formulations are proposed, such as box-constraints only for the inputs (cf. [17, 18]), box-constraints for inputs and states (cf. [15, 84, 146, 149]), norm bounds for inputs and states (cf. [29, 150]), as well as polytopic input and state constraints (see [143]). General quadratic

input and state constraints are considered in [144]. However, hard constraint satisfaction is not guaranteed.

2. Second moment constraints for states and inputs [138] or constraints for the first and second moment of states and inputs [137] are considered.

3. State chance constraints of the form $\Pr(Hx[k] \leq h) \geq \beta$ are used. See [17, 18] for approaches considering hard input constraints and chance constraints for the state $x[k]$ based on the number of particles violating the constraints. In addition, chance constraints can be formulated by tightening the constraints for the disturbance free state predictions [31, 32, 83].

To the best of the author's knowledge, there is no robust MPC approach considering mixed input and state constraints[1]. Furthermore, no approach considers constraints for whole state and input trajectories, such as state or input change rate constraints.

Different **control and prediction strategies**, also known from common and robust MPC formulations for LTI systems (cf. Section 2.5), are proposed to address MPC for JMLS. For finite horizon cost formulations, two approaches are considered:

1. Open-loop prediction: The input trajectory $\boldsymbol{u}[k]$ is determined directly by the optimization problem (e.g. [15, 17, 18, 146]).

2. Closed-loop prediction: The input trajectory is determined indirectly by determining a set of state feedback controllers $K_i[j]$ for all $i \in \Theta$ and $j \in \mathbb{N}_{N-1}^0$ by the online optimization problem, such that $u[j] = K_{\theta_{k+j}}[j]x[k+j]$ minimizes the costs. This approach may reduce the conservatism of the prediction considerably but results in significantly more complex optimization problems (cf. [137, 138]).

In order to calculate the infinite horizon costs, only stable JMS can be considered. Hence, all approaches that consider infinite horizons make use of mean square stabilizing state feedback laws in different ways:

1. A feedback control law $u[k] = K_i[k], i \in \Theta$ is determined at each time step k such that it minimizes the infinite horizon costs [149].

2. As for deterministic systems, dual mode MPC is proposed: The input trajectory $\boldsymbol{u}[j]$ for a given horizon $0 \leq j < N$ and a state feedback controller $u[j] = K_{\theta_{k+j}}[k]x[k+j]$ for all subsequent inputs $j \geq N$ are determined at each time k [150]. For this setup, often one-step approaches ($N = 1$) are considered [84, 103].

3. Similar to tube based MPC formulations (cf. Section 2.5), a linear state feedback law is used in combination with an affine term $v[k]$ such that $u[k] =$

---

[1]Results for design of constrained state feedback controllers (e.g. [38]) might be adopted for the formulation of approaches considering mixed input and state constraints.

$K_{\theta_k} x[k] + v[k]$. Here, only the affine term is calculated online by the MPC (cf. [32, 83, 143]).

In all these problem formulations and control strategies, there exist three **concepts to ensure MSS, robust constraint satisfaction, and recursive feasibility**:

1. As in MPC for LTI systems, stability and recursive feasibility can be ensured by forcing the state into a terminal invariant set [55]. Therefore, all possible Markov state trajectories are simulated and the corresponding state trajectories are forced to end in the terminal set. The number of constraints of the resulting quadratic program (QP) are exponentially increasing with the prediction horizon N. Hence, this approach is only suitable for small scale JMLS and short prediction horizons.

2. LMI constraints are used to guarantee that the controllers $K_i[j]$, i.e., the inputs $u[k]$, lead to a robust decrease of a quadratic Lyapunov function (e.g. [29, 84, 103, 149]). This ensures, that the Lyapunov function is bounded and can be used to formulate ellipsoidal invariant sets. Furthermore, the formulations may consider time-variant system behavior by polytopic over-approximations. However, the semidefinite programming (SDP) formulations solved online exhibit a sharp increase in calculation time with increasing system dimensions. This renders these approaches impractical for an application to high dimensional systems as required for the control of manufacturing systems. The calculation times may be reduced in some cases by calculating state feedback controllers offline and linear combinations of these online. The weighting factors can be calculated by an SDP with a significantly reduced number of optimization variables, but still subject to high dimensional LMI constraints (see [29, 84]).

3. Quadratic Lyapunov functions and ellipsoidal robust control invariant sets (RCIS) are determined offline. Then, the states are forced to stay in these sets and a decrease of the Lyapunov function is guaranteed by quadratic constraints, resulting in quadratically constrained quadratic programs (QCQPs) [15, 83, 146]. This shifts most of the computational effort to the offline calculations and provides online optimization problems that can be solved for higher dimensional problems. However, most formulations approximate the cost functions by considering only a subset of all Markov trajectories [15, 146]. In addition, all formulations consider time-invariant systems [15, 83, 146].

Similar shortcomings characterize the MPC formulations without robustness guarantees. The majority of approaches makes use of computationally expensive optimization problems, such as SDP formulations [137, 138, 144], or mixed integer problems [17, 18]. The only approaches known to the author that employ a QP, or may be formulated as a QP, are [31, 32] and [56], respectively. However, the formulations are not capable of dealing with time-variant system behavior.

The topic of control to general reference trajectories is not discussed in literature. The same holds for the explicit consideration of time-variant but known system matrices. Only in standard finite horizon quadratic control, time-variant system matrices are considered directly (cf. [33, 39]).

Jump Markov Nonlinear Systems

In contrast to the large interest in MPC for JMLS, nonlinear JMS have not drawn much attention. Often, only offline controller design for special formulations of nonlinearities is considered, cf. [86, 127, 152]. Approaches based on online optimization that are similar to MPC are proposed employing T-S fuzzy models [148] and neuronal net techniques [158]. However, none of these approaches considers input and state constraints[2]. In general, there are three approaches that consider state and input constraints:

1. Solve a dynamic programming problem which considers all admissible Markov state trajectories (cf. [104]). The exponential growth of admissible Markov state trajectories with horizon N renders this approach feasible only for JMS with few Markov states and short prediction horizons. By considering only a subset of all Markov state trajectories the effort may by reduced [104]. However, solving the resulting nonlinear program remains challenging.

2. Analog to particle based MPC for JMLS, the nonlinear system behavior can be approximated by simulating a number of random trajectories of all uncertainties [9, 18]. In this framework, the input trajectory $\boldsymbol{u}[k]$ is determined by solving a mixed-integer nonlinear program. The computational effort to solve these problems is very high if larger numbers of particles are used. However, large numbers of particles are needed to guarantee a high level of robustness and an accurate state prediction.

3. The nonlinearities can be considered as a system uncertainty and over-approximated by a differential inclusion. Hence, the system can be regarded as a JMLS with polytopic uncertainties describing the differential inclusion. Then, MPC approaches for uncertain JMLS can be employed [82]. The resulting formulation is quite conservative due to the over-approximations. Furthermore, the computational effort is quite high, since all MPC formulations that consider polytopic system uncertainties solve SDPs online (cf. Section 5.1).

In addition to approaches specifically designed for JMS, NMPC approaches for general nonlinear stochastic systems are proposed [89, 142]. These approaches approximate the system behavior by Gaussian mixture models that describe the probability density function of the predicted states [142] or by Markov chain Monte-Carlo methods [89]. However, according to the authors, both approaches are only suitable for small scale systems or may result in high computation times.

---

[2]Actuator saturation can be considered by appropriately designed input nonlinearities in [86].

Conclusions

For JMLS, a MPC formulation that considers general linear state, input, and output constraints as well as general reference signals is not existing. For the application to large systems, an efficient formulation for the online optimization approach is necessary. However, most of the existing approaches make use of SDP formulations that exhibit a prohibitively large computational effort. A more simple but general open-loop prediction strategy with low computational effort is still an open issue. For the design of robust MPC for JMLS, the two-step approach proposed by [15, 83, 146] (described in point 3. on the bottom of page 74) seems to be the only approach that is applicable to higher dimensional JMLS. However, the approaches employing QPs or QCQPs do either not consider time-variant system behavior or do not guarantee robust constraint satisfaction.

All MPC approaches for nonlinear JMS are either not considering input and state constraints or lead to high computational effort. These approaches are not suitable for most applications. For deterministic nonlinear systems, the problem of computation times is approached by combining MPC with online linearization procedures (see, e.g., [78, 139]). This approach seems to be the most promising for the design of a robust MPC that is fast enough for the system dimensions considered. However, no MPC based on linearizations has been proposed for JMS.

All in all, no existing MPC approach for JMLS or JMS is capable of satisfying all requirements. The main issue with most approaches is the high computational burden, the incapability of dealing with time-variant behavior, or the quite specific constraint formulation. Hence, this part addresses the following goals:

1. Development of a general open-loop prediction strategy for time-variant JMLS, such that the MPC can be posed as a standard QP. A special emphasis is on the computation time of the MPC formulation. Furthermore, general linear constraints and general references have to be considered.

2. A generalization of the two-step concept (of determining RCIS offline and implementing the MPC as a QCQP online using the RCIS) to the JMLS under consideration. The resulting formulation should retain the low computation times of the prediction schemes in 1. and guarantee PMSS, recursive feasibility, and robust constraint satisfaction.

3. Development of an MPC for JMS that applies the developed robust MPC for JMLS (2.) to online linearizations of the JMS. The approach shall retain the low computation times and the guarantees concerning stability and robustness of the approaches for JMLS.

5.2. Problem Setup

Based on the aforementioned goals, the problem setup is described. To this end, this part considers the control of a general isolated JMS:

$$\mathcal{S} := \begin{cases} x[k{+}1] &= f_{\theta_k}\big(x[k], u[k], w[k]\big) \\ y[k] &= g_{\theta_k}\big(x[k], u[k], w[k]\big) \\ \mathcal{M} &= \big(\Theta, P[k], \mu[k]\big) \end{cases} . \tag{5.2}$$

For the sake of a brief notation, the plant model mismatch processes $\nu_{\mathrm{x}}[k]$ and $\nu_{\mathrm{y}}[k]$ are included in the disturbance vector $w[k]$ as additional elements.

Assumption 5.1. *It is assumed that the Markov state θ_k and the continuous state are measurable.*

This assumption is motivated by the fact that the state of a production system and its components is monitored in most cases. For example, the states of the hot stamping line are directly determined by the PLC. Furthermore, large scale production lines are often equipped with sophisticated sensors, so the state $x[k]$ often is measurable. If this is not the case, state estimation has to be employed. The combination of MPC and state estimation will be discussed in Chapter 9.

For this setup, an MPC that considers the following optimal control problem at time $k \in \mathbb{N}$ with references defined by $x_{\mathrm{r}}[j]$, $u_{\mathrm{r}}[j]$, and $y_{\mathrm{r}}[j]$ as well as constraints defined by $H_{\mathrm{x}}, H_{\mathrm{y}}, H_{\mathrm{u}}$, and $h[k+j]$ is developed in this thesis:

$$\min_{\boldsymbol{u}[k]} \quad \sum_{j=1}^{N} \|x[j] - x_{\mathrm{r}}[j]\|^2_{Q_{\theta_{k+j}}} + \sum_{j=0}^{N-1} \left(\|y[j] - y_{\mathrm{r}}[j]\|^2_{S_{\theta_{k+j}}} + \|u[j] - u_{\mathrm{r},\theta_{k+j}}[j]\|^2_{R_{\theta_{k+j}}} \right) \tag{5.3}$$

$$\mathrm{s.\,t.} \quad H_{\mathrm{x}}x[j] + H_{\mathrm{y}}y[j] + H_{\mathrm{u}}u[j] \leq h[k+j] \qquad \forall j \in \mathbb{N}^0_{N-1}. \tag{5.4}$$

Here, $Q_i \geq 0, S_i \geq 0$, and $R_i > 0$ for all $i \in \Theta$, and $N \geq 1$.

Remark 5.1. *Note that the bounds $h[k]$ but not the constraint structure, defined by $H_{(\cdot)}$, are time-variant. This is motivated by the fact that, for most applications, the bounds may vary with ambient conditions or even with the Markov states, but not the general structure of the constraints. In addition, this eases the notation and the extension to $H_{(\cdot)}[k]$ is straightforward.*

Construction of an Augmented JMS: To further ease notation, the system states and outputs are combined in an augmented system state and a corresponding state equation:

$$\hat{x}[k] := \begin{bmatrix} x[k] \\ y[k-1] \end{bmatrix}, \qquad \hat{f}_{\theta_k}(\hat{x}[k], u[k], w[k]) := \begin{bmatrix} f_{\theta_k}\left([I \ \mathbf{0}]\hat{x}[k], u[k], w[k]\right) \\ g_{\theta_k}\left([I \ \mathbf{0}]\hat{x}[k], u[k], w[k]\right) \end{bmatrix}. \tag{5.5}$$

From this definition, the JMS is now defined as a JMS without outputs (as used in almost all work on MPC for JMS). This results in an MPC problem for the augmented system of the following form:

$$\min_{\boldsymbol{u}[k]} \sum_{j=1}^{N} \|\hat{x}[j] - \hat{x}_{\mathrm{r}}[j]\|_{\hat{Q}(j,\boldsymbol{\theta}_{k,N})}^2 + \sum_{j=0}^{N-1} \|u[j] - u_{\mathrm{r},\theta_{k+j}}[j]\|_{R_{\theta_{k+j}}}^2 \tag{5.6a}$$

$$\text{s.\,t.} \quad H_{\hat{x}}\hat{x}[j] + H_{\mathrm{u}}u[j] \le h[k+j] \qquad \forall j \in \mathbb{N}_{N-1}^0, \tag{5.6b}$$

with the extended cost matrix

$$\hat{Q}(j,\boldsymbol{\theta}_{k,N}) = \begin{bmatrix} Q_{\theta_{k+j}} & \mathbf{0} \\ \mathbf{0} & S_{\theta_{k+j-1}} \end{bmatrix}. \tag{5.7}$$

In addition, the constraint matrix $H_{\hat{x}}$ is comprised of block matrix columns from H_{x} and H_{y}. The reference vector $\hat{x}_{\mathrm{r}}$ is constructed analogously to $\hat{x}$.

This merging is possible without restrictions, as a discrete-time setting is considered. However, the different time indices of state and output vector have to be considered carefully when evaluating time or Markov state dependent cost statements and constraints. In the following, the $\hat{\cdot}$ symbols are left out and the augmented system is considered as a standard JMS to improve readability.

Probabilistic Reformulation of the Cost Functions: Since the future Markov states and the disturbances are unknown in advance, the costs in problem (5.6) cannot be calculated and the problem is not solvable in this exact formulation. Hence, two probabilistic versions of the cost function are considered instead:

- **The costs of the expected value of the predicted states**:

$$J_1 := \sum_{j=1}^{N} \|\bar{x}[j] - x_{\mathrm{r}}[j]\|_Q^2 + \sum_{j=0}^{N-1} \|u[j] - \bar{u}_{\mathrm{r},\theta_{k+j}}[j]\|_R^2, \tag{5.8}$$

where $Q \ge 0$ and $R > 0$. In this formulation (referred to by CoE – costs of expected value), the costs are assumed to be independent of the Markov state. Thus, Q and R are constant matrices in this setup. There is no difference whether the JMS is an augmented JMS according to (5.5) or a JMS without outputs.

- **The expected value of the costs**:

$$J_2 := \mathrm{E}\left(\sum_{j=1}^{N} \|x[j] - x_{\mathrm{r}}[j]\|_{Q_{\theta_{k+j}}}^2 + \sum_{j=0}^{N-1} \|u[j] - u_{\mathrm{r},\theta_{k+j}}[j]\|_{R_{\theta_{k+j}}}^2 \right), \tag{5.9}$$

where $Q_i \ge 0$ and $R_i > 0$ for all $i \in \Theta$. This formulation (referred to by EoC – expected value of costs) is the common formulation for JMS (cf. Section 5.1). If the costs depend on the Markov state, it matters whether the JMS is an

augmented JMS (5.5) or a simple JMS without outputs. For the sake of notation, the simple JMS without outputs is considered in the derivations. Adjustments necessary for the consideration of augmented systems are explained in respective remarks.

The first cost function formulation is considered since it is a straightforward approach to deal with the uncertainties in the cost function. In addition, it can be used to formulate the MPC in a very efficient way, as can be seen in the following chapters. The second approach is considered as it is the common approaches to deal with uncertain cost functions in the context of JMS [137].

If the cost matrices are independent of the Markov state θ_k, both cost functions can be compared, cf. [132]:

Theorem 5.1. *For the state cost terms, it holds that:*

$$\mathrm{E}\left(\|x[j] - x_\mathrm{r}[j]\|_Q^2\right) = \|\bar{x}[j] - x_\mathrm{r}[j]\|_Q^2 + \mathrm{tr}\left(Q\mathrm{Var}(x[j] - x_\mathrm{r}[j])\right). \tag{5.10}$$

Proof. For the step costs of the state, it holds:

$$\begin{aligned}
\mathrm{E}\left(\|x[j] - x_\mathrm{r}[j]\|_Q^2\right) &= \mathrm{E}\left(\mathrm{tr}\left(\|x[j] - x_\mathrm{r}[j]\|_Q^2\right)\right) \\
&= \mathrm{E}\left(\mathrm{tr}\left(Q\left(x[j] - x_\mathrm{r}[j]\right)\left(x[j] - x_\mathrm{r}[j]\right)^\top\right)\right) \\
&= \mathrm{tr}\left(Q\mathrm{E}\left(\left(x[j] - x_\mathrm{r}[j]\right)\left(x[j] - x_\mathrm{r}[j]\right)^\top\right)\right).
\end{aligned} \tag{5.11}$$

From the definition of the covariance (see Definition 2.3), it follows:

$$\mathrm{E}\left(\left(x[j] - x_\mathrm{r}[j]\right)\left(x[j] - x_\mathrm{r}[j]\right)^\top\right) = \mathrm{Var}(x[j] - x_\mathrm{r}[j]) + \left(\bar{x}[j] - \bar{x}_\mathrm{r}[j]\right)\left(\bar{x}[j] - \bar{x}_\mathrm{r}[j]\right)^\top. \tag{5.12}$$

Inserting (5.12) into (5.11), reorganizing the resulting term, and using the fact that $x_\mathrm{r}[j] = \bar{x}_\mathrm{r}[j]$ leads to Equation (5.10) as a result. □

A similar relation can be established for the input cost terms. The EoC cost function (5.9) penalizes the expected value of the difference of the state to its reference and the corresponding covariance. Instead, CoE tries to minimize only the expectancy of the difference of the state to its reference. Hence, it is expected that EoC will lead to a better performance and is capable of stabilizing a broader class of JMLS, but is computationally more demanding. Similar cost function formulations have been proposed in the context of fault tolerant MPC for a conventional nonlinear system [90].

Remark 5.2. *The consideration of time-variant costs matrices ($Q[k]$ and $R[k]$) is a straightforward extension of the proposed approaches. For the sake of notation, this is not considered in this thesis.*

Organization of this Part

The following chapters are organized according to the goals specified in Section 5.1. First, efficient prediction approaches for both cost functions (CoE and EoC) are derived for JMLS in Chapter 6. In addition, the formulation of the MPC problem as a standard QP is shown. The focus is on the efficient formulation of the prediction equations in terms of computational effort.

In Chapter 7, a general two-step design approach for a robust MPC for JMLS is presented. First, time-invariant JMLS are considered in Section 7.1. For this case, the design of ellipsoidal RCIS and Lyapunov functions is presented. Then, the formulation of the robust MPC as a quadratically constrained QP is demonstrated. The effects of the additional constraints are investigated with simulations. Finally, the extension to time-variant JMLS is considered in Section 7.2.

A robust MPC for nonlinear JMS is considered in Chapter 8. The whole procedure is based on the idea to apply the robust MPC to a linearization of the JMS considering the linearization errors. To this end, the system linearization, the error set characterization, the design of RCIS, and the MPC formulation are presented in Section 8.1. The properties of the linearization-based MPC are illustrated with a simulation study in Section 8.2.

Finally, Chapter 9 provides a brief overview of state estimation methods for JMS and its combination with the MPC approaches.

6. Efficient MPC Formulation for Jump Markov Linear Systems

In this chapter, efficient MPC formulations for time-variant JMLS are presented employing both cost functions (CoE (5.8) and EoC (5.9)). Motivated by the definition of the augmented system in (5.5), only the state equation is considered:

$$\mathcal{S}_\mathrm{l} := \begin{cases} x[k+1] &= A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[k] + G_{\theta_k}[k]\,w[k] \\ \mathcal{M} &= \big(\Theta, P[k], \mu[k]\big) \end{cases}. \tag{6.1}$$

In this chapter, Assumption 5.1 is relaxed:

Assumption 6.1. *It is assumed that the probability distribution $\mu[k]$ of the Markov state and the continuous state x_k are measurable. If the Markov state is known, the corresponding element of $\mu[k]$ is set to one.*

In addition, a probabilistic version of the constraints is considered for the non-robust MPC formulations presented in this chapter:

$$H_\mathrm{x}\bar{x}[\![j]\!] + H_\mathrm{u}u[\![j]\!] \le h[k+j]. \tag{6.2}$$

This formulation is motivated by the fact that failure rates, i.e., transition probabilities, are relatively low for production systems. Thus, the expected value is a good approximation for the real system behavior.

To formulate the MPC optimization problem for both cost functions (CoE (5.8) and EoC (5.9)), the expectancy of the states or the expectancy of the costs have to be predicted for the entire prediction horizon. Both could be determined directly by calculating the value of the states or costs for each possible Markov trajectory and the probability of the corresponding trajectory. However, this approach would result in an exponential computational complexity $\mathcal{O}(n_\theta^{N+1})$. This is intractable if larger prediction horizons are used. Hence, approaches for the efficient formulation of the MPC problem with polynomial complexity are presented for both cost functions in the following sections. To this end, only condensed MPC formulations (cf. Section 2.5) are considered, since the formulation of a sparse MPC is not applicable for JMLS. It is not possible to consider only the state trajectory as additional optimization variables and add the system dynamics as constraints, due to the different dynamics for each Markov state. It would be possible to add state trajectories for all possible Markov state trajectories as optimization variables. However, this would result in a severe increase in the number of optimization variables, constraints, as well as the computation time.

6.1. Costs of the Expected Value of the States

This section presents efficient prediction equations for the expected value of the states. Based on these equations, the formulation of the MPC (employing the CoE cost function) as a QP is presented. This section is based on results for time-invariant JMLS and zero references ($x_r[k] = \mathbf{0}$ and $u_r[k] = \mathbf{0}$) that have been published in [131].

Prediction Equations for the Expected Value of the States

In order to formulate the MPC problem as a QP, the expected value of the predicted states $\bar{x}[j]$ has to be formulated as a linear function of the current state $x[k]$, the inputs $u[l]$ with $0 \le l < j$, and the expected disturbances $\bar{w}[l]$ with $0 \le l < j$. The prediction equation can be formulated as follows:

$$\bar{x}[j] = \bar{A}[j]\,x[k] + \sum_{l=0}^{j-1} \left(\bar{B}[j,l]\,u[l] + \bar{G}[j,l]\,\bar{w}[l] \right). \tag{6.3}$$

To determine the matrices $\bar{A}[j]$, $\bar{B}[j,l]$, and $\bar{G}[j,l]$, the following conditional expectancy of the predicted states is introduced:

$$\bar{x}_i[j] := \mathrm{E}\left(x[k+j|k, \theta_{k+j-1} = i] \right) = \mathrm{E}\left(x[j] \cdot \mathbb{1}_{\{\theta_{k+j-1}=i\}} \right). \tag{6.4}$$

It describes the expectancy of the predicted states for the case that the Markov state θ_{k+j-1} equals $i \in \Theta$. Note that the Markov state of the previous time step $(k + j - 1)$ is considered. In contrast, the expectancy is commonly conditioned on the Markov state of the same time step $(k + j)$ in literature (see, e.g., [39]). The different conditioning in this thesis is motivated by the fact that the system state $x[k+j]$ is determined by $f_{\theta_{k+j-1}}(\cdot)$. In addition, it simplifies the following formulas.

Let the prediction of the conditional expectancies $\bar{x}_i[j]$ be defined by the following linear prediction equation:

$$\bar{x}_i[j] = \tilde{A}_i[j]\,x[k] + \sum_{l=0}^{j-1} \left(\tilde{B}_i[j,l]\,u[l] + \tilde{G}_i[j,l]\,\bar{w}[l] \right), \tag{6.5}$$

where the matrices $\tilde{A}_i[j]$, $\tilde{B}[j,l]$, and $\tilde{G}[j,l]$ describe the influence of $x[k]$, $u[l]$, and $\bar{w}[l]$ on $\bar{x}_i[j]$, respectively. The prediction matrices can be determined recursively according to the following theorem:

Theorem 6.1. *Let the prediction of the expected value of the states $\bar{x}[j]$ and the prediction of the conditional expectancy $\bar{x}_i[j]$ be defined as in (6.3) and (6.5). The operator $\mathcal{V}_i(\cdot, \cdot)$ and $\mu[j]$ are defined according to (2.53) and (2.48). Then, the prediction matrices $\bar{A}[j]$, $\bar{B}[j,l]$, and $\bar{G}[j,l]$ as well as $\tilde{A}_i[j]$, $\tilde{B}_i[j,l]$, and $\tilde{G}_i[j,l]$ can be determined by the following algorithm:*

1. **Initialization:**

$$\tilde{A}_i[1] = \mu_i[k]\, A_i[k] \qquad\qquad\qquad\qquad \forall i \in \Theta, \qquad\qquad (6.6)$$

$$\tilde{B}_i[j, j-1] = \mu_i[j-1]\, B_i[j-1], \quad \tilde{G}_i[j, j-1] = \mu_i[j-1]\, G_i[j-1] \quad \forall i \in \Theta, j \in \mathbb{N}_N. \quad (6.7)$$

2. **Recursion for all $j \in \mathbb{N}_{N-1}$:**

$$\tilde{A}_i[j+1] = A_i[j]\, \mathcal{V}_i\left(\tilde{A}[j], j-1\right) \qquad\qquad \forall i \in \Theta, \qquad\qquad (6.8)$$

$$\tilde{B}_i[j+1, l] = A_i[j]\, \mathcal{V}_i\left(\tilde{B}[j, l], j-1\right) \qquad\qquad \forall i \in \Theta, l \in \mathbb{N}_{j-1}^0, \qquad (6.9)$$

$$\tilde{G}_i[j+1, l] = A_i[j]\, \mathcal{V}_i\left(\tilde{G}[j, l], j-1\right) \qquad\qquad \forall i \in \Theta, l \in \mathbb{N}_{j-1}^0. \qquad (6.10)$$

3. **Matrix Calculation:**

$$\bar{A}[j] = \sum_{i=1}^{n_\theta} \tilde{A}_i[j], \qquad \bar{B}[j, l] = \sum_{i=1}^{n_\theta} \tilde{B}_i[j, l], \qquad \bar{G}[j, l] = \sum_{i=1}^{n_\theta} \tilde{G}_i[j, l], \qquad \forall j \in \mathbb{N}_N. \qquad (6.11)$$

Proof. The first two steps, i.e., the calculation of the matrices $\tilde{A}_i[j]$, $\tilde{B}_i[j, l]$, and $\tilde{G}_i[j, l]$, are proven by induction. The induction starts with the prediction of the expectancy for $j = 1$. The first prediction step follows from the state dynamics:

$$\bar{x}_i[1] = \mathrm{E}\left((A_i[k]\, x[k] + B_i[k]\, u[0] + G_i[k]\, w[0]) \cdot \mathbb{1}_{\{\theta_k = i\}}\right). \qquad (6.12)$$

According to Assumption 2.2, the Markov state θ_k and $w[0] = w[k]$ are stochastically independent. All system matrices, $x[k]$, and $u[0]$ are deterministic. Hence, the expectancy can be calculated by:

$$\bar{x}_i[1] = \left(A_i[k]\, x[k] + B_i[k]\, u[0] + G_i[k]\, \bar{w}[0]\right) \mu_i[k]$$
$$= \mu_i[k]\, A_i[k]\, x[k] + \mu_i[k]\, B_i[k]\, u[0] + \mu_i[k]\, G_i[k]\, \bar{w}[0]. \qquad (6.13)$$

The expressions $\mu_i[k]\, A_i[k]$, $\mu_i[k]\, B_i[k]$, and $\mu_i[k]\, G_i[k]$ coincide with the initializations in (6.6) and (6.7) for $j = 1$. The general induction step is given by the calculation of the expectancy for $j + 1$:

$$\bar{x}_i[j+1] = \mathrm{E}\left(x[j+1] \cdot \mathbb{1}_{\{\theta_{k+j} = i\}}\right) = \mathrm{E}\left((A_i[j]\, x[j] + B_i[j]\, u[j] + G_i[j]\, w[j]) \cdot \mathbb{1}_{\{\theta_{k+j} = i\}}\right)$$
$$= \mathrm{E}\left(A_i[j]\, x[j] \cdot \mathbb{1}_{\{\theta_{k+j} = i\}}\right) + \underbrace{\mu_i[j]\, B_i[j]}_{= \tilde{B}_i[j+1, j]}\, u[j] + \underbrace{\mu_i[j]\, G_i[j]}_{= \tilde{G}_i[j+1, j]}\, \bar{w}[j]. \qquad (6.14)$$

Note that the labeled terms coincide with the initializations in (6.6) and (6.7). The first term in equation (6.14) can be conditioned on the previous time step (see Lemma 2.7 on page 32):

$$\mathrm{E}\left(A_i[j]\, x[j] \cdot \mathbb{1}_{\{\theta_{k+j} = i\}}\right) = \mathrm{E}\left(A_i[j]\, x[j] \sum_{m=1}^{n_\theta} p_{m,i}[k+j-1] \cdot \mathbb{1}_{\{\theta_{k+j-1} = m\}}\right)$$
$$= A_i[j] \sum_{m=1}^{n_\theta} p_{m,i}[k+j-1]\, \mathrm{E}\left(x[j] \cdot \mathbb{1}_{\{\theta_{k+j-1} = m\}}\right). \qquad (6.15)$$

The expectancy coincides with the definition of the conditioned expectancy in (6.4). Hence, the prediction equation (6.5) can be inserted in (6.15) to obtain:

$$\mathrm{E}\left(A_i[j]\, x[j] \cdot \mathbb{1}_{\{\theta_{k+j}=i\}}\right) \tag{6.16}$$

$$= A_i[j] \sum_{m=1}^{n_\theta} p_{m,i}[k+j-1] \left(\tilde{A}_m[j]\, x[k] + \sum_{l=0}^{j-1}\left(\tilde{B}_m[j,l]\, u[l] + \tilde{G}_m[j,l]\, \bar{w}[l]\right)\right)$$

$$= \underbrace{A_i[j]\, \mathcal{V}_i\left(\tilde{A}[j], j-1\right)}_{=:\ \tilde{A}_i[j+1]} x[k] + \sum_{l=0}^{j-1}\left(\underbrace{A_i[j]\, \mathcal{V}_i\left(\tilde{B}[j,l], j-1\right)}_{=:\tilde{B}_i[j+1,l]} u[l] + \underbrace{A_i[j]\, \mathcal{V}_i\left(\tilde{G}[j,l], j-1\right)}_{=:\tilde{G}_i[j+1,l]} \bar{w}[l]\right).$$

Note that the labeled terms coincide with the recursions (6.8) - (6.10). Thus, the induction holds if the prediction matrices are chosen as stated in Theorem 6.1.

From the definition of the conditional expectancy it follows that (cf. [39]):

$$\bar{x}[j] = \sum_{i=1}^{n_\theta} \bar{x}_i[j]. \tag{6.17}$$

Hence, summing Equation (6.5) over all Markov states and comparing the result with the general prediction equation (6.3) results in the equations in (6.11). □

In each recursion, all transitions that lead to Markov state i are aggregated in $\bar{x}_i[j]$. This is possible due to the proposed definition of the conditional expectancies $\bar{x}_i[j]$. The recursive formulation reduces the computational complexity of the calculation of the prediction matrices to $\mathcal{O}\left(n_\theta^2 \cdot N^2\right)$. This enables the application of this prediction scheme to larger system dimensions and prediction horizons compared to the direct calculation of all trajectories, which incurs exponential complexity.

Remark 6.1. *The prediction scheme can be considered as an extension of the recursions commonly used for the prediction of the first moment of the states in [40, 137]. In contrast to these formulations, the prediction scheme proposed also considers the dependencies on the inputs u[j] and the expected disturbances $\bar{w}[j]$. These dependencies are crucial for the formulation of the MPC problem.*

Formulation of the Optimization Problem

The linear prediction equation (6.3) for the expected value of the states can be used to formulate the MPC problem with cost function (5.8) as a common QP. To this end, the following trajectories are introduced: $\bar{\boldsymbol{x}}[k] := \begin{bmatrix} \bar{x}[0] & \cdots & \bar{x}[N-1] \end{bmatrix}$, $\boldsymbol{u}[k] := \begin{bmatrix} u[0] & \cdots & u[N-1] \end{bmatrix}$, $\bar{\boldsymbol{w}}[k] := \begin{bmatrix} \bar{w}[0] & \cdots & \bar{w}[N-1] \end{bmatrix}$, and $\boldsymbol{x}_\mathrm{r}[k] := \begin{bmatrix} x_\mathrm{r}[0] & \cdots & x_\mathrm{r}[N-1] \end{bmatrix}$. The expected value of the input reference can be calculated with the predicted probability distribution $\mu[j]$:

$$\bar{\boldsymbol{u}}_\mathrm{r}[k] := \begin{bmatrix} \sum_{i=1}^{n_\theta} \mu_i[0]\, u_{\mathrm{r},i}[0] & \cdots & \sum_{i=1}^{n_\theta} \mu_i[N-1]\, u_{\mathrm{r},i}[N-1] \end{bmatrix}. \tag{6.18}$$

Employing these definitions, the cost function (5.8) can be formulated as follows:

$$J[k] = \|\bar{\boldsymbol{x}}[1] - \boldsymbol{x}_{\mathrm{r}}[1]\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{u}[k] - \bar{\boldsymbol{u}}_{\mathrm{r}}[k]\|_{\boldsymbol{R}}^2, \tag{6.19}$$

with block diagonal cost matrices $\boldsymbol{Q} = \mathrm{diag}\,(Q, \cdots, Q)$ and $\boldsymbol{R} = \mathrm{diag}\,(R, \cdots, R)$.

Remark 6.2. *Costs caused by combinations of states or inputs of different prediction steps, e.g., costs for the change rate of states or inputs, can be formulated by adding off-diagonal elements to $\boldsymbol{Q}$ and $\boldsymbol{R}$. This is a straightforward extension of the presented approaches and is not elaborated in this thesis.*

Furthermore, the prediction of the expected values of the states can be formulated in the following condensed form:

$$\bar{\boldsymbol{x}}[1] = \boldsymbol{A}[k]\,x[k] + \boldsymbol{B}[k]\,\boldsymbol{u}[k] + \boldsymbol{G}[k]\,\bar{\boldsymbol{w}}[k], \tag{6.20}$$

where the condensed prediction matrices are:

$$\boldsymbol{A}[k] = \begin{bmatrix} \bar{A}[1] \\ \vdots \\ \bar{A}[N] \end{bmatrix}, \quad \boldsymbol{B}[k] = \begin{bmatrix} \bar{B}[1,0] & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \ddots & \boldsymbol{0} \\ \bar{B}[N,0] & \cdots & \bar{B}[N,N-1] \end{bmatrix}, \quad \boldsymbol{G}[k] = \begin{bmatrix} \bar{G}[1,0] & \boldsymbol{0} & \boldsymbol{0} \\ \vdots & \ddots & \boldsymbol{0} \\ \bar{G}[N,0] & \cdots & \bar{G}[N,N-1] \end{bmatrix}. \tag{6.21}$$

In addition, let the matrices $\boldsymbol{A}_{\mathrm{c}}[k]$, $\boldsymbol{B}_{\mathrm{c}}[k]$, and $\boldsymbol{G}_{\mathrm{c}}[k]$ be defined analogously, such that $\bar{\boldsymbol{x}}[k] = \boldsymbol{A}_{\mathrm{c}}[k]\,x[k] + \boldsymbol{B}_{\mathrm{c}}[k]\,\boldsymbol{u}[k] + \boldsymbol{G}_{\mathrm{c}}[k]\,\bar{\boldsymbol{w}}[k]$. In this vector, the current state $x[k]$ is contained and $\bar{x}[N]$ is omitted, since no mixed input and state constraints can be formulated for the last prediction step (due to the lack of the value $u[N]$). Simple linear constraints on $\bar{x}[N]$ could be added, if necessary.

Based on these definitions, the main result for the MPC formulation is stated:

Theorem 6.2. *The solution of the MPC optimization problem with cost function (5.8) and constraints (6.2) is equivalent to that of the following QP:*

$$\min_{\boldsymbol{u}[k]} \quad \|\boldsymbol{u}[k]\|_{W[k]}^2 + q[k]\,\boldsymbol{u}[k] \tag{6.22a}$$

$$\mathrm{s.\,t.} \quad (\boldsymbol{H}_{\mathrm{x}}\,\boldsymbol{B}_{\mathrm{c}}[k] + \boldsymbol{H}_{\mathrm{u}})\,\boldsymbol{u}[k] \le \boldsymbol{h}[k] - \boldsymbol{H}_{\mathrm{x}}\,(\boldsymbol{A}_{\mathrm{c}}[k]\,x[k] + \boldsymbol{G}_{\mathrm{c}}[k]\,\bar{\boldsymbol{w}}[k]), \tag{6.22b}$$

where the matrices $W[k]$ and $q[k]$ are selected as follows:

$$W[k] = \boldsymbol{B}^{\top}[k]\,\boldsymbol{Q}\,\boldsymbol{B}[k] + \boldsymbol{R}, \tag{6.23}$$

$$q[k] = 2\,(\boldsymbol{A}[k]\,x[k] + \boldsymbol{G}[k]\,\bar{\boldsymbol{w}}[k] - \boldsymbol{x}_{\mathrm{r}}[1])^{\top}\,\boldsymbol{Q}\,\boldsymbol{B}[k] - 2\bar{\boldsymbol{u}}_{\mathrm{r}}^{\top}[k]\boldsymbol{R}, \tag{6.24}$$

and the constraints are defined by:

$$\boldsymbol{H}_{\mathrm{x}} = I_N \otimes H_{\mathrm{x}}, \qquad \boldsymbol{H}_{\mathrm{u}} = I_N \otimes H_{\mathrm{u}}, \qquad \boldsymbol{h}[k] = \begin{bmatrix} h^{\top}[k] & \cdots & h^{\top}[k+N-1] \end{bmatrix}^{\top}. \tag{6.25}$$

Proof. Inserting prediction equation (6.20) in cost function (6.19) results in:

$$J[k] = \left\| \boldsymbol{A}[k] x[k] + \boldsymbol{B}[k] \boldsymbol{u}[k] + \boldsymbol{G}[k] \bar{\boldsymbol{w}}[k] - \boldsymbol{x}_{\mathrm{r}}[1] \right\|_{\boldsymbol{Q}}^2 + \left\| \boldsymbol{u}[k] - \bar{\boldsymbol{u}}_{\mathrm{r}}[k] \right\|_{\boldsymbol{R}}^2. \qquad (6.26)$$

By expanding the quadratic expression, neglecting all terms independent of the optimization variables $\boldsymbol{u}[k]$, and some straightforward manipulations, the parameterization of the cost function by (6.23) and (6.24) can be derived.

The constraints (6.2) can be formulated in the condensed formulation:

$$\boldsymbol{H}_{\mathrm{x}} \, \bar{\boldsymbol{x}}[k] + \boldsymbol{H}_{\mathrm{u}} \, \boldsymbol{u}[k] \le \boldsymbol{h}[k]. \qquad (6.27)$$

Inserting the prediction equation $\bar{\boldsymbol{x}}[k] = \boldsymbol{A}_{\mathrm{c}}[k] \, x[k] + \boldsymbol{B}_{\mathrm{c}}[k] \, \boldsymbol{u}[k] + \boldsymbol{G}_{\mathrm{c}}[k] \, \bar{\boldsymbol{w}}[k]$ into (6.27) provides (6.22b). $\qquad \square$

For the QP (6.22), efficient solvers exist. Hence, the QP formulation in Theorem 6.2 and the recursive calculation of the prediction matrices according to Theorem 6.1 provide a very efficient approach to formulate and solve the MPC problem (6.24). The control performance and the computational properties are investigated in Section 6.3.

Remark 6.3. *If the Markov state θ_k is measurable, the one-step prediction $\bar{x}[1]$ is deterministic except for the disturbance $w[k]$. Since the disturbance $w[k]$ is bounded, hard constraints on $x[k+1]$ can be formulated by constraint tightening (cf. Section 2.5). If the optimization problem is feasible at each time step, the constraints are satisfied robustly. However, recursive feasibility is not guaranteed for the approach proposed. The formulation of an MPC scheme that is guaranteed to be recursively feasible is presented in Chapter 7.*

Remark 6.4. *If elements off the block diagonal of $\boldsymbol{H}_{\mathrm{u}}$ and $\boldsymbol{H}_{\mathrm{x}}$ are considered, constraints on the whole state and input trajectory can be stated, e.g., change rate constraints for the states or inputs. This is not possible for the majority of the MPC approaches for JMLS proposed in literature.*

6.2. Expected Value of the Costs

This section presents the efficient formulation of the EoC variant of the MPC

$$\min_{\boldsymbol{u}[k]} \quad \mathrm{E}\left(\sum_{j=1}^{N} \left\| x[\![j]\!] - x_{\mathrm{r}}[\![j]\!] \right\|_{Q_{\theta_{k+j}}}^2 + \sum_{j=0}^{N-1} \left\| u[\![j]\!] - u_{\mathrm{r},\theta_{k+j}}[\![j]\!] \right\|_{R_{\theta_{k+j}}}^2 \right) \qquad (6.28a)$$

$$\text{s.\,t.} \quad H_{\mathrm{x}} \bar{x}[\![j]\!] + H_{\mathrm{u}} u[\![j]\!] \le h[k+j] \qquad \forall j \in \mathbb{N}_{N-1}^0, \qquad (6.28b)$$

in the form of a QP. This MPC problem only differs from that considered in the previous section in the formulation of the cost function. Thus, the constraint formulation can be adopted from the previous section. The reformulation of the cost function and the resulting QP are presented in the following sections. The results are based on less general results for time-invariant JMLS and zero references $x_{\mathrm{r}}[k] = \boldsymbol{0}$ as well as $u_{\mathrm{r}}[k] = \boldsymbol{0}$ published in [132] by the author.

Reformulation of the Cost Function

The cost function in (6.28a) can be written as follows:

$$J_2[k] = \sum_{j=1}^{N} \mathrm{E} \left(\|x[j]\|_{Q_{\theta_{k+j}}}^2 - 2x_\mathrm{r}^\top[j]\, Q_{\theta_{k+j}}\, x[j] + \|x_\mathrm{r}[j]\|_{Q_{\theta_{k+j}}}^2 \right) + \ldots \tag{6.29a}$$

$$\cdots + \sum_{j=0}^{N-1} \mathrm{E} \left(\|u[j]\|_{R_{\theta_{k+j}}}^2 - 2u_{\mathrm{r},\theta_{k+j}}^\top[j]\, R_{\theta_{k+j}}\, u[j] + \|u_{\mathrm{r},\theta_{k+j}}[j]\|_{R_{\theta_{k+j}}}^2 \right). \tag{6.29b}$$

The cost function can then be transformed into a function that is quadratic in $\boldsymbol{u}[k]$. This is done by transforming all terms separately into the required form and aggregating the resulting formulations into one. Cost terms independent of $\boldsymbol{u}[k]$ do not affect the solution of the resulting optimization problem. Hence, all these terms can be neglected during the reformulation. However, the dependency of the costs on the expected disturbances $\bar{\boldsymbol{w}}[k]$ are crucial for the design of the distributed MPC, and are needed later. Hence, only terms independent of $\boldsymbol{u}[k]$ and $\bar{\boldsymbol{w}}[k]$ are neglected. The summands $\|x_\mathrm{r}[j]\|_{Q_{\theta_{k+j}}}^2$ and $\|u_{\mathrm{r},\theta_{k+j}}[j]\|_{R_{\theta_{k+j}}}^2$ are such negligible terms.

Input Costs: The remaining input and input reference costs in (6.29b) can be formulated as follows:

Lemma 6.1. *Let $\mu[j]$ be defined according to* (2.48). *Then, the following equation holds:*

$$\sum_{j=0}^{N-1} \mathrm{E} \left(\|u[j]\|_{R_{\theta_{k+j}}}^2 - 2u_{\mathrm{r},\theta_{k+j}}^\top[j]\, R_{\theta_{k+j}}\, u[j] \right) = \|\boldsymbol{u}[k]\|_{\boldsymbol{R}[k]}^2 - 2\Phi_{\mathrm{u_r u}}[k]\, \boldsymbol{u}[k] \tag{6.30}$$

if the matrices $\boldsymbol{R}[k]$ and $\Phi_{\mathrm{u_r u}}[k]$ are defined as follows:

$$\boldsymbol{R}[k] = \mathrm{diag}\left(\sum_{i=1}^{n_\theta} \mu_i[0]\, R_i, \quad \cdots, \quad \sum_{i=1}^{n_\theta} \mu_i[N-1]\, R_i \right) \tag{6.31}$$

$$\Phi_{\mathrm{u_r u}}[k] = \left[\sum_{i=1}^{n_\theta} \mu_i[0]\, u_{\mathrm{r},i}^\top[0]\, R_i \quad \cdots \quad \sum_{i=1}^{n_\theta} \mu_i[N-1]\, u_{\mathrm{r},i}^\top[N-1]\, R_i \right]. \tag{6.32}$$

Proof. The inputs $u[j]$ are deterministic quantities determined by the MPC. The cost matrices R_i and references $u_{\mathrm{r},i}[j]$ depend only on the Markov state. Hence, the expected values can be determined with the probability distribution $\mu[j]$:

$$\sum_{j=0}^{N-1} \mathrm{E} \left(\|u[j]\|_{R_{\theta_{k+j}}}^2 - 2u_{\mathrm{r},\theta_{k+j}}^\top[j]\, R_{\theta_{k+j}}\, u[j] \right)$$

$$= \sum_{j=0}^{N-1} \left(u^\top[j]\, \mathrm{E}\left(R_{\theta_{k+j}} \right) u[j] - 2\mathrm{E}\left(u_{\mathrm{r},\theta_{k+j}}^\top[j]\, R_{\theta_{k+j}} \right) u[j] \right)$$

$$= \sum_{j=0}^{N-1} \left(u^\top[j] \left(\sum_{i=1}^{n_\theta} \mu_i[j]\, R_i \right) u[j] - 2 \left(\sum_{i=1}^{n_\theta} \mu_i[j]\, u_{\mathrm{r},i}^\top[j]\, R_i \right) u[j] \right). \tag{6.33}$$

The equations (6.30) - (6.32) follow directly from transforming the sum over all prediction steps in (6.33) into a matrix form with $\boldsymbol{u}[k]$. $\qquad\square$

State Reference Costs: The step costs caused by the state reference in (6.29a) can be reformulated as follows:

Lemma 6.2. *Let the operator* $\mathcal{V}_i(\cdot,\cdot)$ *be defined according to (2.53). Then, the following equation holds:*

$$\mathrm{E}\left(x_\mathrm{r}[\![j]\!] \, Q_{\theta_{k+j}} \, x[\![j]\!]\right) = \Phi_\mathrm{xru}[\![j]\!] \, \boldsymbol{u}[k] + \Phi_\mathrm{xrw}[\![j]\!] \, \bar{\boldsymbol{w}}[k] + \Psi, \qquad (6.34)$$

where Ψ *collects terms independent of* $\boldsymbol{u}[k]$ *and* $\bar{\boldsymbol{w}}[k]$, *and:*

$$\Phi_\mathrm{xru}[\![j]\!] = x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} Q_i \left[\mathcal{V}_i\left(\tilde{B}[\![j,0]\!], j-1\right) \;\; \cdots \;\; \mathcal{V}_i\left(\tilde{B}[\![j,j-1]\!], j-1\right) \;\; \boldsymbol{0}_{n_\mathrm{x} \times (N-j)n_\mathrm{u}} \right], \quad (6.35)$$

$$\Phi_\mathrm{xrw}[\![j]\!] = x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} Q_i \left[\mathcal{V}_i\left(\tilde{G}[\![j,0]\!], j-1\right) \;\; \cdots \;\; \mathcal{V}_i\left(\tilde{G}[\![j,j-1]\!], j-1\right) \;\; \boldsymbol{0}_{n_\mathrm{x} \times (N-j)n_\mathrm{w}} \right]. \quad (6.36)$$

Proof. Since the reference $x_\mathrm{r}[\![j]\!]$ is deterministic, it follows:

$$\mathrm{E}\left(x_\mathrm{r}^\top[\![j]\!] \, Q_{\theta_{k+j}} \, x[\![j]\!]\right) = x_\mathrm{r}^\top[\![j]\!] \, \mathrm{E}\left(Q_{\theta_{k+j}} \, x[\![j]\!]\right)$$

$$= x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} Q_i \sum_{m=1}^{n_\theta} p_{m,i}[k+j-1] \, \mathrm{E}\left(x[\![j]\!] \cdot \mathbb{1}_{\{\theta_{k+j-1}=m\}}\right)$$

$$= x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} Q_i \sum_{m=1}^{n_\theta} p_{m,i}[k+j-1] \, \bar{x}_m[\![j]\!]. \qquad (6.37)$$

The last equation follows from the definition of the conditional expectancy in (6.4). The conditional expectancy can be determined by the prediction equation (6.5). By collecting the terms independent of $\boldsymbol{u}[k]$ and $\bar{\boldsymbol{w}}[k]$ in Ψ, it follows:

$$x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} Q_i \sum_{m=1}^{n_\theta} p_{m,i}[k+j-1] \left(\tilde{A}_m[\![j]\!] \, x[k] + \sum_{l=0}^{j-1} \left(\tilde{B}_m[\![j,l]\!] \, u[l] + \tilde{G}_m[\![j,l]\!] \, \bar{w}[l] \right) \right)$$

$$= x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} Q_i \sum_{l=0}^{j-1} \left[\mathcal{V}_i\left(\tilde{B}[\![j,l]\!], j-1\right) u[l] + \mathcal{V}_i\left(\tilde{G}[\![j,l]\!], j-1\right) \bar{w}[l] \right] + \Psi$$

$$= \Phi_\mathrm{xru}[\![j]\!] \, \boldsymbol{u}[k] + \Phi_\mathrm{xrw}[\![j]\!] \, \bar{\boldsymbol{w}}[k] + \Psi. \qquad (6.38)$$

The last equation results from formulating the sum over l in matrix form. $\qquad\square$

The prediction matrices $\tilde{B}_m[\![j,l]\!]$ and $\tilde{G}_m[\![j,l]\!]$ can be determined according to Theorem 6.1. Finally, the costs for the whole prediction horizon can be determined by summing up (6.34) for all $j \in \mathbb{N}_N$.

Remark 6.5. *If the JMLS under consideration is an augmented system according to (5.5), the determination of the matrices* $\Phi_\mathrm{xru}[\![j]\!]$ *and* $\Phi_\mathrm{xrw}[\![j]\!]$ *has to be adapted. In this case, the cost matrix depends on two consecutive Markov states, and it follows:*

$$\mathrm{E}\left(x_\mathrm{r}^\top[\![j]\!] \, Q_{\theta_{k+j}} \, x[\![j]\!]\right) = x_\mathrm{r}^\top[\![j]\!] \sum_{i=1}^{n_\theta} \sum_{m=1}^{n_\theta} \begin{bmatrix} Q_i & \boldsymbol{0} \\ \boldsymbol{0} & S_m \end{bmatrix} p_{m,i}[k+j-1] \, \bar{x}_m[\![j]\!]. \qquad (6.39)$$

Inserting the prediction equation (6.5) provides the adapted formulation.

State Costs: Finally, the state costs $\mathrm{E}\!\left(\|x[j]\|_{Q_{\theta_j}}^2\right)$ have to be reformulated. Since none of the factors is deterministic, the calculation of this term becomes more involved. Parts of the calculation have been published in [133].

Lemma 6.3. *Let the operator $\mathcal{T}_i(\cdot,\cdot)$ be defined according to (2.54). The expected value of the step costs of the states for a given k and $j \geq 1$ can be transformed into the following quadratic expression:*

$$\mathrm{E}\!\left(\|x[j]\|_{Q_{\theta_j}}^2\right) = \boldsymbol{u}^\top[k]\,\Phi_{\mathrm{uu}}[j]\,\boldsymbol{u}[k] + 2x^\top[k]\,\Phi_{\mathrm{xu}}[j]\,\boldsymbol{u}[k] + 2x^\top[k]\,\Phi_{\mathrm{xw}}[j]\,\bar{\boldsymbol{w}}[k] \quad (6.40)$$
$$+ 2\bar{\boldsymbol{w}}^\top[k]\,\Phi_{\mathrm{wu}}[j]\,\boldsymbol{u}[k] + \bar{\boldsymbol{w}}^\top[k]\,\Phi_{\mathrm{ww}}[j]\,\bar{\boldsymbol{w}}[k] + \Psi.$$

The cost prediction matrices $\Phi_{(\cdot,\cdot)}$ can be calculated recursively. To this end, the following recursion matrices are introduced for all $l \in \mathbb{N}_j$ and $i \in \Theta$:

$$Q_i(j) \coloneqq \mathcal{T}_i(Q, j-1), \qquad\qquad \boldsymbol{Q}_{\mathrm{wu},i}(j) \coloneqq \begin{bmatrix} \mathbf{1}_j \otimes Q_i(j) & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(N-j)\cdot n_{\mathrm{w}} \times (N-j)n_{\mathrm{u}}} \end{bmatrix},$$

$$\boldsymbol{Q}_{\mathrm{xu},i}(j) \coloneqq \begin{bmatrix} \mathbf{1}_{1\times j} \otimes Q_i(j) & \mathbf{0}_{n_{\mathrm{x}} \times (N-j)n_{\mathrm{u}}} \end{bmatrix}, \quad \boldsymbol{Q}_{\mathrm{xw},i}(j) \coloneqq \begin{bmatrix} \mathbf{1}_{1\times j} \otimes Q_i(j) & \mathbf{0}_{n_{\mathrm{x}} \times (N-j)n_{\mathrm{w}}} \end{bmatrix},$$

$$\boldsymbol{Q}_{\mathrm{uu},i}(j) \coloneqq \begin{bmatrix} \mathbf{1}_j \otimes Q_i(j) & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(N-j)n_{\mathrm{u}}} \end{bmatrix}, \qquad \boldsymbol{Q}_{\mathrm{ww},i}(j) \coloneqq \begin{bmatrix} \mathbf{1}_j \otimes Q_i(j) & \mathbf{0} \\ \mathbf{0} & \mathbf{0}_{(N-j)n_{\mathrm{w}}} \end{bmatrix},$$

$$\boldsymbol{B}'_i[l] \coloneqq \mathrm{diag}\Big(\underbrace{A_i[l-1], \dots, A_i[l-1]}_{l-1 \text{ times}}, B_i[l-1], I_{(N-l)\cdot n_{\mathrm{u}}}\Big), \qquad\qquad (6.41)$$

$$\boldsymbol{G}'_i[l] \coloneqq \mathrm{diag}\Big(\underbrace{A_i[l-1], \dots, A_i[l-1]}_{l-1 \text{ times}}, G_i[l-1], I_{(N-l)\cdot n_{\mathrm{w}}}\Big).$$

Then, the prediction matrices are given by the following algorithm:

1. **Initialization**:

   Given j, set $m = 1$ and calculate:

   $$\phi_{\mathrm{uu},i}^{(1)} \coloneqq \big(\boldsymbol{B}'_i[j]\big)^\top \boldsymbol{Q}_{\mathrm{uu},i}(j)\, \boldsymbol{B}'_i[j], \quad \phi_{\mathrm{xu},i}^{(1)} \coloneqq A_i^\top[j-1]\,\boldsymbol{Q}_{\mathrm{xu},i}(j)\, \boldsymbol{B}'_i[j], \quad \forall i \in \Theta \quad (6.42)$$

   $$\phi_{\mathrm{xw},i}^{(1)} \coloneqq A_i^\top[j-1]\,\boldsymbol{Q}_{\mathrm{xw},i}(j)\, \boldsymbol{G}'_i[j], \quad \phi_{\mathrm{wu},i}^{(1)} \coloneqq \big(\boldsymbol{G}'_i[j]\big)^\top \boldsymbol{Q}_{\mathrm{wu},i}(j)\, \boldsymbol{B}'_i[j], \quad \forall i \in \Theta \quad (6.43)$$

   $$\phi_{\mathrm{ww},i}^{(1)} \coloneqq \big(\boldsymbol{G}'_i[j]\big)^\top \boldsymbol{Q}_{\mathrm{ww},i}(j)\, \boldsymbol{G}'_i[j] \qquad\qquad\qquad\qquad\qquad \forall i \in \Theta. \quad (6.44)$$

2. **Recursion**:

   a) *If $m < j$:*

   $$\phi_{\mathrm{uu},i}^{(m+1)} \coloneqq \big(\boldsymbol{B}'_i[j-m]\big)^\top \mathcal{T}_i\!\left(\phi_{\mathrm{uu}}^{(m)}, j-m-1\right) \boldsymbol{B}'_i[j-m] \quad \forall i \in \Theta, \qquad (6.45\mathrm{a})$$

   $$\phi_{\mathrm{xu},i}^{(m+1)} \coloneqq A_i^\top[j-m-1]\,\mathcal{T}_i\!\left(\phi_{\mathrm{xu}}^{(m)}, j-m-1\right) \boldsymbol{B}'_i[j-m] \quad \forall i \in \Theta, \qquad (6.45\mathrm{b})$$

   $$\phi_{\mathrm{xw},i}^{(m+1)} \coloneqq A_i^\top[j-m-1]\,\mathcal{T}_i\!\left(\phi_{\mathrm{xw}}^{(m)}, j-m-1\right) \boldsymbol{G}'_i[j-m] \quad \forall i \in \Theta, \qquad (6.45\mathrm{c})$$

   $$\phi_{\mathrm{wu},i}^{(m+1)} \coloneqq \big(\boldsymbol{G}'_i[j-m]\big)^\top \mathcal{T}_i\!\left(\phi_{\mathrm{wu}}^{(m)}, j-m-1\right) \boldsymbol{B}'_i[j-m] \quad \forall i \in \Theta, \qquad (6.45\mathrm{d})$$

   $$\phi_{\mathrm{ww},i}^{(m+1)} \coloneqq \big(\boldsymbol{G}'_i[j-m]\big)^\top \mathcal{T}_i\!\left(\phi_{\mathrm{ww}}^{(m)}, j-m-1\right) \boldsymbol{G}'_i[j-m] \quad \forall i \in \Theta, \qquad (6.45\mathrm{e})$$

   else: go to 3.

  b) *Set $m := m + 1$ and go to a).*

3. **Calculation of the prediction matrices**:

$$\Phi_{\text{uu}}[j] = \sum_{i=1}^{n_\theta} \mu_i[k] \, \phi_{\text{uu},i}^{(j)}, \quad \Phi_{\text{xu}}[j] = \sum_{i=1}^{n_\theta} \mu_i[k] \, \phi_{\text{xu},i}^{(j)}, \quad \Phi_{\text{xw}}[j] = \sum_{i=1}^{n_\theta} \mu_i[k] \, \phi_{\text{xw},i}^{(j)}, \quad (6.46)$$

$$\Phi_{\text{wu}}[j] = \sum_{i=1}^{n_\theta} \mu_i[k] \, \phi_{\text{wu},i}^{(j)}, \quad \Phi_{\text{ww}}[j] = \sum_{i=1}^{n_\theta} \mu_i[k] \, \phi_{\text{ww},i}^{(j)}. \quad (6.47)$$

Proof. The proof can be found in Appendix A.1. $\qquad\qquad\qquad\qquad\qquad$ □

From the recursion equations (6.45) and the deductions in the proof, it can be seen that the matrices $\phi_{(\cdot,\cdot)}$ are constructed recursively from the inside to the outside. The recursion begins with matrices related to the prediction step $k + j$ and iterates back to the present time k. This guarantees that left and right matrix multiplications are performed for matrices for the same Markov state. Finally, the information about the probability distribution $\mu[k]$ are employed in *3.* to calculate the cost prediction matrices. All in all, *1.* and *2.* describe the evolution of the costs over the prediction horizon and *3.* connects the evolution with the current state and Markov state.

The prediction equation (6.40) shows that only the expectancy of the disturbance influences cost terms that can be affected by the inputs. Thus, zero-mean disturbances do not affect the optimal input trajectory but the residual costs Ψ.

Remark 6.6. *If a JMLS that is an augmented system (including outputs) is considered, the auxiliary matrix $Q_i(j)$ has to be defined as follows:*

$$Q_i(j) := \begin{bmatrix} \mathcal{T}_i(Q, j-1) & \mathbf{0} \\ \mathbf{0} & S_i \end{bmatrix}. \quad (6.48)$$

All other definitions and equations still hold.

Computational Effort of the Recursion: The presented recursions reduce the exponential computational complexity to a polynomial one. The computational effort to calculate the matrices $\Phi_{\text{uu}}[j]$, ..., $\Phi_{\text{ww}}[j]$ is larger than the effort for the calculation of the prediction matrices for the expectancy of the states according to Theorem 6.1. This is due to the large dimensions of the recursion matrices as well as the more involved recursions. Hence, steps *1.* and *2.* produce the majority of the computational effort. Since step *1.* and *2.* are independent of the current Markov state and the continuous state, these steps can be performed offline if the JMLS is time-invariant. Then, only the final calculation of the cost prediction matrices in *3.* needs to be performed online. Unfortunately, for the production systems considered, time-variant behavior is often encountered and the matrices have to be determined online. Due to this, the algorithm would only be applicable to medium scale JMLS in the presented form.

However, it is possible to mitigate this issue significantly, if the special structure of the prediction matrices is considered. Most recursion matrices are sparse, since the step costs at $k + j$ can be influenced only by the inputs $u[l]$ with $l < j$. The matrices have the following structure:

$$\boldsymbol{Q}_{\mathrm{uu},i}(1) = \begin{bmatrix} * & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0}_{(N-1)n_{\mathrm{u}}} \end{bmatrix}, \quad \boldsymbol{Q}_{\mathrm{uu},i}(2) = \begin{bmatrix} * & * & \boldsymbol{0} \\ * & * & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}_{(N-2)n_{\mathrm{u}}} \end{bmatrix}, \quad \cdots \quad \boldsymbol{Q}_{\mathrm{uu},i}(N) = \begin{bmatrix} * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \end{bmatrix},$$

$$\boldsymbol{Q}_{\mathrm{xu},i}(1) = \begin{bmatrix} * & \boldsymbol{0}_{1\times(N-1)n_{\mathrm{u}}} \end{bmatrix}, \quad \boldsymbol{Q}_{\mathrm{xu},i}(2) = \begin{bmatrix} * & * & \boldsymbol{0}_{1\times(N-2)n_{\mathrm{u}}} \end{bmatrix}, \quad \cdots \quad \boldsymbol{Q}_{\mathrm{xu},i}(N) = \begin{bmatrix} * & \cdots & * \end{bmatrix}.$$

The $*$ denotes matrices different from $\boldsymbol{0}$. Furthermore, the matrices $\boldsymbol{B}'_i[l]$ and $\boldsymbol{G}'_i[l]$ are block diagonal and contain identity matrices to a large extend. Hence, a substantial part of the matrix multiplication in the recursions (6.45) are multiplications with zero and identity matrices. By avoiding these multiplications, the computation time can be reduced considerably. These calculations can be avoided if the matrices $\Phi_{\mathrm{uu}}[j]$, ..., $\Phi_{\mathrm{ww}}[j]$ are calculated for all combinations of $u[l], \bar{w}[l]$, and $x[k]$ separately. With the resulting block matrices, the overall matrices $\Phi_{\mathrm{uu}}[j]$, ..., $\Phi_{\mathrm{ww}}[j]$ for the vectors $\boldsymbol{u}[k], \bar{\boldsymbol{w}}[k]$, and $x[k]$ can be constructed. This approach results in the same prediction matrices, but improves the computation time for larger dimensions significantly. For example, the overall MPC computation time for a JMLS with $n_{\mathrm{x}} = 100, n_{\mathrm{u}} = 30, N = 10$, and $n_\theta = 10$ is reduced from 64 s to 0.15 s if the matrix calculation is optimized this way. A detailed investigation of the resulting computation times for the MPC with optimized matrix calculation is presented in Section 6.3. A detailed presentation of the overall algorithm is omitted, since it is quite lengthy and does not provide much insight into the problem at hand.

Formulation of the Optimization Problem

The MPC optimization problem can be formulated as a QP by combining the cost reformulation procedure presented in this section and the constraint formulation presented in Section 6.1:

Theorem 6.3. *Let the constraints be defined as in Theorem 6.2, and the matrices* $\Phi_{(\cdot,\cdot)}$ *and* $\boldsymbol{R}[k]$ *be defined according to Lemmata 6.1, 6.2, and 6.3. Then, the solution of the MPC optimization problem* (6.28) *is equivalent to that of the following QP:*

$$\min_{\boldsymbol{u}[k]} \quad \|\boldsymbol{u}[k]\|_{W[k]}^2 + q[k]\,\boldsymbol{u}[k] \tag{6.49a}$$

$$\mathrm{s.\,t.} \quad (\boldsymbol{H}_{\mathrm{x}}\,\boldsymbol{B}_{\mathrm{c}}[k] + \boldsymbol{H}_{\mathrm{u}})\,\boldsymbol{u}[k] \le \boldsymbol{h}[k] - \boldsymbol{H}_{\mathrm{x}}\left(\boldsymbol{A}_{\mathrm{c}}[k]\,x[k] + \boldsymbol{G}_{\mathrm{c}}[k]\,\bar{\boldsymbol{w}}[k]\right), \tag{6.49b}$$

with:

$$W[k] = \sum_{j=1}^{N} \Phi_{\mathrm{uu}}[j] + \boldsymbol{R}[k], \tag{6.50}$$

$$q[k] = \sum_{j=1}^{N} \left(2x^{\top}[k]\Phi_{\mathrm{xu}}[j] + 2\bar{\boldsymbol{w}}^{\top}[k]\Phi_{\mathrm{wu}}[j] - 2\Phi_{\mathrm{x_r u}}[j]\right) - 2\Phi_{\mathrm{u_r u}}[k]. \tag{6.51}$$

Proof. The equality of the cost functions (6.49a) and (6.28a) follows from the cost decomposition in (6.29b) and Lemmata 6.1, 6.2, and 6.3. The constraint formulation is adopted from Theorem 6.2. □

The dimension of the QP is the same as for the QP presented in the previous section (considering the CoE cost function). Hence, the computation time for solving the QP is nearly the same and the complexity of the approaches differs only in the determination of the prediction matrices. The prediction matrices $\boldsymbol{A}_{\mathrm{c}}[k]$, $\boldsymbol{B}_{\mathrm{c}}[k]$, and $\boldsymbol{G}_{\mathrm{c}}[k]$ have to be determined for both approaches. The cost prediction matrices $\Phi_{(\cdot,\cdot)}$ are only needed for the approach presented in this section (considering the EoC). Due to this, the computation time of the CoE approach is inherently smaller than for the EoC approach. However, the absolute computation time is still relatively small. This is particularly true, if the computational effort is compared to SDP or mixed integer linear programming (MILP) formulations commonly used for MPC for JMLS. An investigation of the computation time is presented in Section 6.3.

Due to the same constraint formulation as for the CoE approach, the properties concerning input and state trajectory constraint formulation and hard constraint formulations (explained in Remarks 6.3 and 6.4) also apply for the formulation presented in this section.

6.3. Evaluation and Comparison of the Approaches

This section presents a comparison of the MPC approaches proposed in Sections 6.1 and 6.2 concerning stability properties, control performance, computation time, and reference control. Except for the evaluation of the computation time, the disturbance free case ($w[k] = \boldsymbol{0}$) is considered. Robust approaches are considered in Chapter 7.

Stability Properties

Both MPC approaches only differ in the cost function formulation. The main difference is that minimizing the EoC includes a minimization of the CoE and the variance of the predicted state trajectory (see Theorem 5.1 on page 79). This has a direct impact on the stability properties and the control performance. In terms of stability, two questions arise: First, does a convergence of the cost function to zero imply stability? Second, is the convergence guaranteed? The results presented in this section are mainly based on [132].

Convergence of the Cost Function and Stability Whether a convergence of the cost function implies stability depends on the measurability of the Markov state. If the Markov state is measurable, the one-step prediction $x[\![1]\!]$ and the costs for the first prediction step are deterministic. If the corresponding costs equal zero and $Q > 0$ or $Q_i > 0$, the state is converged to the reference. Thus, a convergence of the predicted costs implies MSS for both approaches. If the Markov state is not measurable, this reasoning does not apply. For the CoE approach, a convergence

of the costs does not imply stability anymore, since the expectancy of the states may be equal to the reference while this is not the case for all realizations of the Markov state trajectory. This will be shown by an illustrative example later. For the approach that minimizes the EoC, the convergence of the costs implies stability:

Lemma 6.4. *If the costs (5.9) converge to zero, i.e.,* $\lim\limits_{k\to\infty} J_2[k] = 0$, *and* $Q_i > 0$ *for all* $i \in \Theta$, *MSS is implied for the controlled JMLS.*

Proof. $J_2[k] \to 0$ implies $\mathrm{E}\big(\|x[j] - x_\mathrm{r}[j]\|^2_{Q_{\theta_{k+j}}}\big) \to 0$ for all $j \in \mathbb{N}_N$. The expectancy can be calculated by considering all possible Markov state trajectories:

$$\mathrm{E}\left(\|x[j] - x_\mathrm{r}[j]\|^2_{Q_{\theta_{k+j}}}\right) = \sum_{\Theta_{k,j}} \Pr(\boldsymbol{\theta}_{k,j})\|x[j] - x_\mathrm{r}[j]\|^2_{Q_{\theta_{k+j}}}, \tag{6.52}$$

where $\Theta_{k,j}$ is the set of all Markov state sequences $\boldsymbol{\theta}_{k,j}$ from time k to $k + j$. Since $Q_{\theta_{k+j}} > 0$, the costs $\|x[j] - x_\mathrm{r}[j]\|^2_{Q_{\theta_{k+j}}}$ are equal to zero for all $\boldsymbol{\theta}_{k,j}$ and it holds:

$$\|x[1] - x_\mathrm{r}[1]\|^2_{Q_{\theta_{k+j}}} = 0 \quad \Rightarrow \quad \mathrm{E}\left(\|x[k + 1] - x_\mathrm{r}[k + 1]\|^2\right) = 0. \tag{6.53}$$

This implication coincides with the definition of MSS. $\qquad\square$

Example 6.1. To illustrate the difference between both MPC approaches, the following JMLS is considered:

$$A_1 = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 0 & 1.1 \\ 1.1 & 0 \end{bmatrix}, \quad B_1 = I, \quad B_2 = 20I, \quad P = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}. \tag{6.54}$$

This JMLS is simulated with both MPC approaches for 1,000 Markov trajectories, initial conditions $\mu[0] = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^\top$ and $x[0] = \begin{bmatrix} 1 & -1 \end{bmatrix}^\top$, $N = 4$, and cost matrices $Q = 10I$ as well as $R = I$. The envelopes of all state trajectories are shown in Fig. 6.1. The results show that the CoE approach does not stabilize the system. In fact,
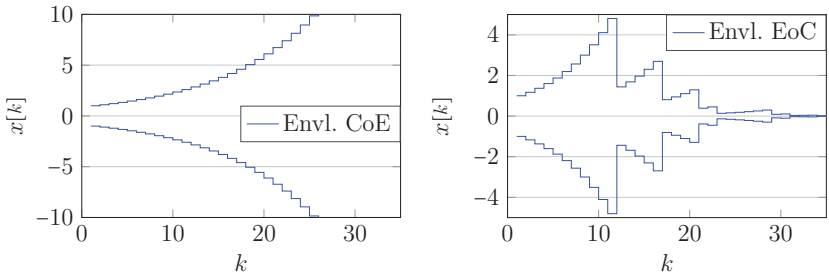
Figure 6.1.: Envelopes of the state trajectories for the JMLS in (6.54). The jumps in the plot on the right are due to transitions of the Markov state.

it does not control the system at all, i.e., all inputs are set to zero. This is caused by the special structure of the system. It causes the predicted expectancy of the states and the costs to be zero at all time. In contrast, the variance of the states is not zero and the EoC approach is able to stabilize the system in the MSS sense. This illustrates that the EoC approach may stabilize a broader class of systems.   △

Convergence Properties   No general statement concerning the convergence of the costs can be made for both approaches. Whether the MPC stabilizes the JMLS depends on the JMLS, the horizon length N, and the cost matrices. On the one hand, an increase of N leads to additional information about the system and a better controller performance. On the other hand, the prediction quality decreases with each prediction step due to the uncertainties of the open-loop prediction. This is particularly a problem for the EoC approach, since the variance of the predicted states increases with the horizon length and it becomes the dominating part of the cost function. This may result in a degraded control performance. Hence, in contrast to deterministic systems, stability cannot be guaranteed by means of a large horizon in general.

All in all, stability is not guaranteed for both approaches. However, a suitable selection of the cost function and horizon length often results in a stable closed-loop behavior. To guarantee stability, two approaches are possible:

1. Formulation of a closed-loop prediction with input trajectories as a function of the future Markov states (for details see Appendix B) in combination with large or infinite horizons.

2. Design of additional constraints that ensure a convergence of the states.

The second approach is presented in Chapter 7. The first approach is not considered in this thesis, since this approach results in significantly larger computation times and is less suitable for time-variant JMLS.

Control Performance

This section investigates the control performance of both MPC approaches in terms of resulting input and state costs. The difference in the two cost function formulations, i.e., the consideration of the state variance in the EoC approach, has a direct influence on the resulting control performance. Due to the combined minimization of the expected value and the variance of the states, it is expected that the control performance of the EoC approach is better than for the CoE approach.

Example 6.2. This expectation is confirmed by a comparison of both approaches as published in [132]. In this comparison, 10 randomly generated, time-invariant JMLS with $n_x = 10$, $n_u = 4$, $n_\theta = 4$ are considered. For all systems, only unstable system matrices A_i are used. The resulting systems are controlled by both MPC approaches

Table 6.1.: Comparison of the average costs for both MPC approaches for 10 randomly generated JMLS [132].

No.	1	2	3	4	5	6	7	8	9	10
CoE	117	178	87	312	181	$1.2 \cdot 10^5$	143	$2 \cdot 10^9$	365	111
EoC	98	170	81	349	166	$1.1 \cdot 10^3$	139	946	224	114
$\frac{J_{\mathrm{EoC}} - J_{\mathrm{CoE}}}{J_{\mathrm{CoE}}}$ in %	-16.7	-4.2	-7.0	11.6	-8.6	-	-2.5	-	-38.5	3.3

with $N = 5$, $Q_i = 0.5I$, and $R_i = I$ for all $i \in \Theta$. For details on the simulation setup, see [132]. The resulting average costs for 200 simulations are shown in Table 6.1.

The results show that the EoC approach outperforms the CoE approach in 8 out of 10 cases. In addition, systems 6 and 8 are only stabilized by the EoC approach and not by the CoE approach. However, the results also show that the EoC approach does not always perform better (systems 4 and 10). This is due to the fact that the minimization of the variance of the state trajectories may slow down the convergence. This issue is illustrated in Example 6.3. Furthermore, the relation between both approaches depends on the horizon length N. To illustrate this, a JMLS with the specified dimensions is simulated with both MPC formulations for all $N \in \{2, \ldots, 11\}$. The resulting average costs are shown in Fig. 6.2 as functions of the horizon length N for $Q_i = 0.5I$ and $R_i = I$ as well as $Q_i = 5I$ and $R_i = I$. The results for the CoE approach show a convergence of the resulting costs with increasing prediction horizon length. In contrast, the costs for the EoC approach show a minimum for a certain horizon length and increase significantly for larger horizon lengths. This illustrates the effects caused by the minimization of the variance (as described in the previous section). △
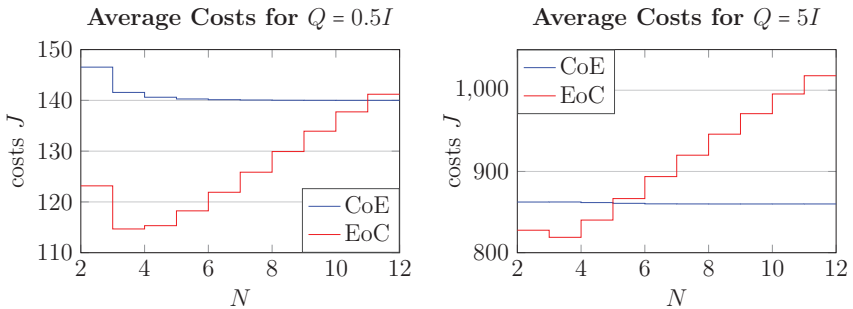
Figure 6.2.: Average cost values of the controlled JMLS as a function of the prediction horizon length N for both MPC approaches.

Example 6.3. To illustrate the effects of the different cost functions in more detail, a low-dimensional JMLS that models an economic system is considered. The following simulations are an extension of simulations presented in [131] for the CoE approach. The three Markov states represent a "normal", "booming", or "slumping" economy. The corresponding dynamics are defined by [19, 38]:

$$A_1 = \begin{bmatrix} 0 & 1 \\ -2.5 & 3.2 \end{bmatrix}, \; A_2 = \begin{bmatrix} 0 & 1 \\ -4.3 & 4.5 \end{bmatrix}, \; A_3 = \begin{bmatrix} 0 & 1 \\ 5.3 & -5.2 \end{bmatrix}, B_1 = B_2 = B_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad (6.55)$$

$$P = \begin{bmatrix} 0.67 & 0.17 & 0.16 \\ 0.30 & 0.47 & 0.23 \\ 0.26 & 0.10 & 0.64 \end{bmatrix}, \quad x[0] = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mu[0] = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}.$$

This setup results in a significant variance of the predicted states. The JMLS is controlled with both MPC formulations with $N = 6$, $R = 1$, and $Q = 0.185I$. The states and the input are constrained to the interval $[-1, 1]$. The average values for the input and the states as well as the corresponding envelopes of 10,000 simulation runs are shown in Fig. 6.3. The plots show that both MPC formulations stabilize the JMLS asymptotically, but the control behavior is different. The CoE approach controls the system more aggressively. This results in larger input values and a faster convergence of the states. Hence, the additional minimization of the state variance slows down the control. Due to this, the costs of the EoC approach are
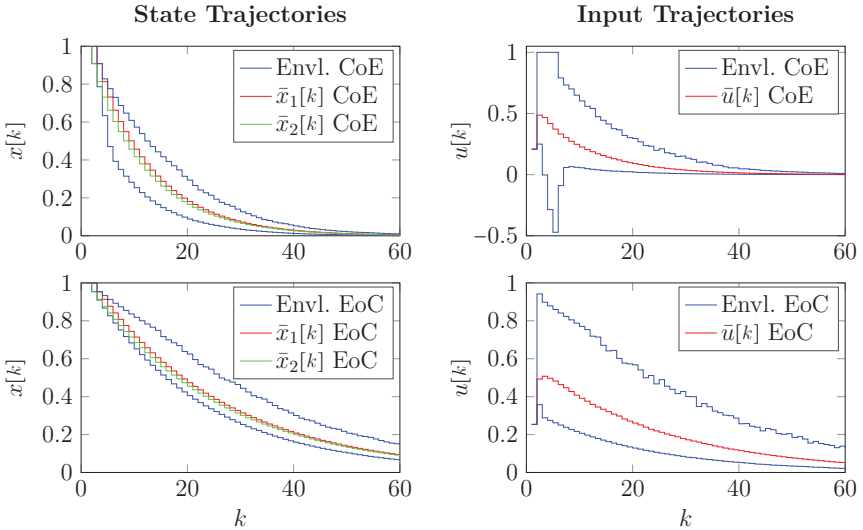
Figure 6.3.: Comparison of state and input trajectories with the corresponding envelopes for the JMLS (6.55) controlled by both MPC approaches.

8.74 in average compared to 3.83 for the CoE approach. This issue can be mitigated by reducing the horizon length but it cannot be solved completely. However, it can be solved by considering input trajectories that depend on the Markov states (see Appendix B). With this formulation, the variance of the state prediction is decreased considerably and the EoC approach produces even better control results than the CoE approach (average cost of 2.78). While the EoC approach produces higher average costs, it shows favorable properties in terms stability and robustness: If $Q = 5I$ is considered instead of $Q = 0.185I$, the CoE approach does not stabilize the JMLS anymore while the EoC formulation still does. The same holds if $N = 3$ is considered. $\triangle$

All in all, the results show that no general statement regarding control performance of both approaches is possible. In most cases, there is a suitable horizon length for which the EoC approach performers better than the CoE approach. However, this is not the case for all JMLS and cost configurations. In addition, the EoC approach shows favorable properties in terms of closed-loop stability. Since rather extreme examples are presented to demonstrate the properties of the MPC approaches, the difference between both approaches is likely to be much smaller for real systems (cf. the simulations of the hot stamping line in Chapter 12 and 13).

Computation Time

To investigate the scalability of the proposed MPC approaches, the computation times for time-invariant JMLS with different system dimension have been determined. The simulations were performed with Matlab 2016b on a Linux system with an i7-6700K processor. The average time for computation of the prediction matrices and solving the QP at each time step k are shown in Table 6.2 in ms.

The results show that the computation time for the EoC approach is by a factor

Table 6.2.: Average computation times per time step k in ms for both MPC approaches (CoE and EoC) and different system dimensions ($n_{\mathrm{w}} = n_{\mathrm{u}}$).

N	n_θ	\(n_{\mathrm{x}}, n_{\mathrm{u}}\) (5,5) CoE	EoC	(50,15) CoE	EoC	(75,20) CoE	EoC	(100,30) CoE	EoC
	5	1	3	5	10	6	12	11	21
5	10	3	7	7	17	11	25	22	49
	15	4	12	12	29	18	41	33	77
	5	4	12	13	29	21	40	38	68
10	10	10	27	25	60	38	87	67	148
	15	17	47	41	103	61	147	115	267
	5	9	27	28	62	44	84	85	145
15	10	21	62	54	136	83	191	159	336
	15	37	109	91	239	139	344	254	578

of 2 to 3 higher compared to the CoE approach. This is due to the more complex calculation of the cost prediction matrices for the EoC approach. However, the absolute calculation times demonstrate the effectiveness of both approaches. Even for large system dimensions, the resulting computation times are significantly below one second. Hence, both approaches are applicable to most system models of production systems. Moreover, the approaches proposed outperform SDP and MILP formulations used in MPC approaches for JMLS by far. For example, the time for solving a common SDP formulation for a JMLS with $n_\mathrm{w} = n_\mathrm{u} = 15, n_\mathrm{x} = 50$, and $n_\theta = 5$ once is about three minutes [132].

Reference Control

This section describes different possibilities to consider references $x_\mathrm{r}[k]$ and $u_{\mathrm{r},i}[k]$ different from zero. In general, two possibilities exist to consider references:

1. The references are considered directly in the cost function (as shown in Sections 6.1 and 6.2).

2. A difference system, denoted by "delta system", with $x_\delta[k] := x[k] - x_\mathrm{r}[k]$, $u_\delta[k] := u[k] - u_{\mathrm{r},\theta_k}[k]$, and $w_\delta[k] := w[k] - \bar{w}[k]$ is controlled to the origin.

A schematic comparison of both approaches is depicted in Fig. 6.4. For JMLS, both approaches behave differently, if the input reference depends on the Markov state $(u_{\mathrm{r},\theta_k}[k])$. Since the predicted inputs $u[j]$ are independent of the Markov state, it is not possible to consider the correct reference $u_{\mathrm{r},\theta_k}[j]$ in the cost function. Hence, expected values of the input reference are considered in both cost functions. In general, the minimization of these expected values does not coincide with the steady state of the JMLS and an offset from the reference results. For the EoC approach, this effect is stronger, since the minimization of $\mathrm{Var}(x[j] - x_\mathrm{r}[j])$ drags the system to a set point with an input reference independent of the Markov state. This is in many cases the origin. The problem does not occur if the delta system is controlled to the origin, since the dependency of the input reference is considered directly in
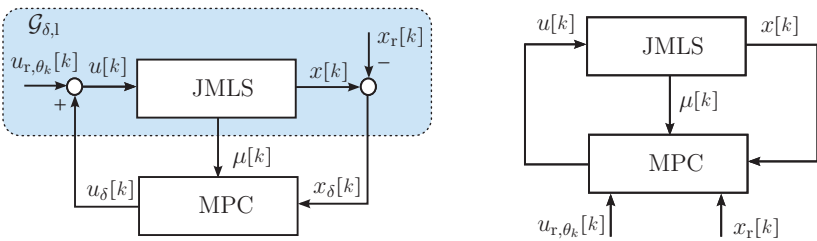
Figure 6.4.: Block diagrams for an MPC controlling the delta system (left) and an MPC considering the reference in the cost function (right).
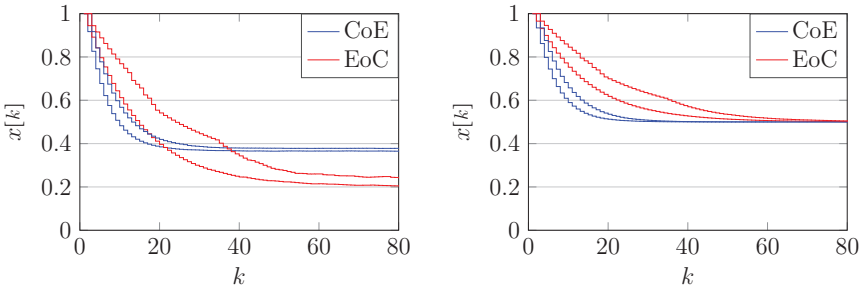
Figure 6.5.: State trajectory envelopes for both MPCs considering the reference in the cost function (left) and employing the delta system (right).

the definition of $u_\delta[k]$ and not in the cost function. If the delta system converges to the origin, the original system converges to the defined reference. However, the formulation of the constraints has to be adapted as follows:

$$H_x \bar{x}_\delta[j] + H_u u_\delta[k] \le h[k+j] - H_x x_r[j] - H_u u_{r,\theta_{k+j}}[j]. \tag{6.56}$$

The Markov state θ_{k+j}, however, is not known at time k and these constraints can only be formulated for the current time step. All future constraints have to be approximated by expected values, worst case formulations, or left out completely. Hence, the advantage concerning stationary accuracy comes at the expense of approximated constraints for all future prediction steps.

Example 6.4. The effects described concerning stationary accuracy of both methods can be illustrated by simulations. To this end, the economic example JMLS defined in (6.55) is employed. The goal is to stabilize the system at the steady state $x_r = [0.5\ 0.5]^\top$ and $U_r = \{0.15; 0.4; 0.45\}$. Both the CoE and the EoC approach are tested with $N = 3$, $Q = 0.5I$, and $R = 1$. The state trajectory envelopes of 1,000 simulations for both approaches to consider the references are shown in Fig. 6.5. The plots illustrate the problems discussed: The states do not converge to the reference if the reference is considered in the cost function. The issue gets worse if the prediction horizon is increased. For the delta system, both approaches converge to the reference. $\triangle$

The example shows that a direct consideration of the reference in the cost function (as presented in this chapter) is not applicable, if the input references vary significantly for the different Markov states. In this case, a delta system could be employed to guarantee stationary accuracy. However, this method causes problems for the constraint formulation. For the application to real processes, both problems should be avoided. Two possibilities exist to solve the issue concerning stationary accuracy employing the original system:

- Employing an extended MPC formulation that determines inputs that depend on the Markov state ($u_{\theta_{k+j}}[j]$ instead of $u[j]$). Then, $u_{\theta_{k+j}}[j] - u_{\mathrm{r},\theta_k}[j]$ can be minimized and the convergence issue disappears (see Appendix B).

- Force the states to converge to the reference by contracting constraints.

Both approaches increase the computational effort of the resulting optimization problem. For the second approach, the increase in computation time is smaller and PMSS is guaranteed. Hence, approach two is employed in this thesis and explained in the following chapter in detail.

7. Robust Model Predictive Control for Jump Markov Linear Systems

The MPC approaches presented in the previous section neither guarantee PMSS nor recursive feasibility. In this chapter, it is demonstrated how the MPC approaches can be extended such that these properties are guaranteed. For the sake of notation, separate linear state and input constraints:

$$H_{\mathrm{u}}\, u[k] \leq h_{\mathrm{u}}[k], \qquad\qquad H_{\mathrm{x}}\, x[k] \leq h_{\mathrm{x}}[k] \qquad\qquad \forall\ k \qquad (7.1)$$

are considered instead of mixed input and state constraints as in (6.2). However, the adaptions that are necessary to consider mixed input and state constraints are presented briefly. Note that the actual value of the states is used instead of its expected value, i.e., hard state constraints are considered. This is only possible if the disturbance set $\mathbb{W}$ is bounded (cf. Assumption 2.2 on page 31). This requires the disturbance models $\check{\nu}$ (introduced in Section 3.2.2) to be bounded, since these are elements of the considered disturbances $w[k]$. For the following chapters, it is assumed that the disturbance $w[k]$ is bounded by the ellipsoid $\mathcal{E}_{\mathrm{w}}$ centered at the expected value of the disturbance[1]:

$$\mathbb{W}: \qquad \mathcal{E}_{\mathrm{w}} = \mathcal{E}(\bar{w}, \Lambda_{\mathrm{w}}). \qquad (7.2)$$

This formulation is motivated by the fact that the iso-density locus of a normal distribution is an ellipsoid (see Section 2.3) and that set operations, such as Minkowski differences and sums, can be approximated efficiently for ellipsoids.

General Idea   The general idea to formulate the robust MPC is based on concepts from [15] and has been published by the author in [133]. For its presentation, the concept of robust control invariant sets[2] (RCIS) is employed:

Definition 7.1. *For any $i \in \Theta$, let $\mathbb{X}'_i \subset \mathbb{R}^{n_{\mathrm{x}}}$ be a compact set. The sets $\mathbb{X}'_i$ are called RCIS if for all $(\theta_k, x[k])$ with $x[k] \in \mathbb{X}'_{\theta_k}$ there exists an input $u[k] \in \mathbb{U}$ such that $x[k+1] \in \mathbb{X}'_{\theta_{k+1}}$ for all disturbances $w[k] \in \mathbb{W}$ where $\mathbb{W}$ is bounded.*

This definition implies that $x[k+j]$ with $j \in \mathbb{N}$ can be kept in the union of the sets $\mathbb{X}'_i$ if $x[k] \in \mathbb{X}'_{\theta_k}$. The union $\bigcup_{i \in \Theta} \mathbb{X}'_i$ is a compact set, i.e., the states are bounded.

The MPC design is divided into an offline part and an online part. First, a stabilizing auxiliary feedback law, a switched Lyapunov function for the closed-loop

---

[1]The expected value $\bar{w}$ is assumed to be time-invariant. A consideration of a time-variant expected values is possible but would further complicate the formulations and explanations.

[2]The reader is referred to [67] for invariant set definitions for classical nonlinear systems.

system according to (2.63), and ellipsoidal RCIS are determined offline by solving an SDP problem. The controller is designed to satisfy the input and state constraints, to guarantee PMSS, and to keep the state in the RCIS, if $x[0]$ is in the RCIS. Then, the MPC problem solved online is adapted in three ways:

1. The state constraints are tightened by the error set $\mathbb{W}$ in order to guarantee robust constraint satisfaction (cf. Section 2.5).

2. Constraints that force the subsequent state $x[k+1]$ to be in the determined RCIS are added.

3. A constraint that forces the expected value of the Lyapunov function to decrease at each time step is added.

The second point ensures recursive feasibility, since the feedback law always provides a feasible solution for the problem. The third point guarantees PMSS. This approach is presented for time-invariant JMLS in the following section and is extended to time-variant JMLS in Section 7.2.

7.1. Robust MPC for Time-Invariant JMLS

In this section, the following time-invariant JMLS is considered:

$$\mathcal{S}_{\mathrm{LTI}} := \left\{ \begin{array}{rcl} x[k+1] &=& A_{\theta_k}\,x[k] + B_{\theta_k}\,u[k] + G_{\theta_k}\,w[k] \\ \mathcal{M} &=& (\Theta, P, \mu[k]) \end{array} \right. . \tag{7.3}$$

The robust MPC presented in this section is based on results published in [133]. These results are generalized such that non-zero references can be considered. The reference (x_{r}, $u_{\mathrm{r},i}$, and $\bar{w}$) is assumed to be a steady state of the JMLS according to Definition 2.11, in order to be able to guarantee PMSS.

7.1.1. Design of Invariant Sets and Controllers

This section presents an SDP-based design approach for ellipsoidal RCIS $\mathcal{E}_i := \mathcal{E}(\mathcal{Q}_i)$ with $\mathcal{Q}_i \in \mathbb{S}_{n_{\mathrm{x}}}^{>0}$ for all $i \in \Theta$ and a switched linear control law

$$u_{\delta}[k] = K_{\theta_k} x_{\delta}[k]. \tag{7.4}$$

To this end, the 'delta system' denoted by $\mathcal{S}_{\mathrm{l},\delta}$ is employed with:

$$x_{\delta}[k] := x[k] - x_{\mathrm{r}}, \qquad u_{\delta}[k] := u[k] - u_{\mathrm{r},\theta_k}, \qquad \tilde{w}[k] := w[k] - \bar{w}. \tag{7.5}$$

Then, the closed-loop dynamics with control law (7.4) are given by:

$$x_{\delta}[k+1] = \left(A_{\theta_k} + B_{\theta_k}K_{\theta_k}\right)x_{\delta}[k] + G_{\theta_k}\,\tilde{w}[k], \tag{7.6}$$

where $\tilde{w}[k] \in \tilde{\mathcal{E}}_{\mathrm{w}} := \mathcal{E}(0, \Lambda_{\mathrm{w}})$. The ellipsoid containing the complete additive influence of the disturbance $G_i\,\tilde{w}[k]$ is denoted by $\mathcal{E}_{\mathrm{w},i} := \mathcal{E}\left(E_i\right)$ with $E_i = G_i\Lambda_{\mathrm{w}}G_i^{\mathsf{T}}$. It is assumed that these ellipsoids are non-degenerate, i.e., the shape matrices E_i are invertible. This assumption is made to be able to present an SDP formulation that

can be used also in the following chapters. For the case that this assumption does not hold, alternative formulations are presented in this section. In the next sections, LMI and bi-linear matrix inequality (BMI) constraints are derived that can be used to determine RCIS and controllers K_i for this setup.

Robust Control Invariant Sets and Stability

The ellipsoids $\mathcal{E}_i$ define the RCIS for the closed-loop system if

$$x_\delta[k] \in \mathcal{E}_{\theta_k} \implies x_\delta[k+1] \in \mathcal{E}_{\theta_{k+1}} \quad \forall \, \tilde{w}[k] \in \tilde{\mathcal{E}}_w \tag{7.7}$$

holds for all k. This condition can be formulated by matrix inequalities as follows[3]:

Lemma 7.1. *The ellipsoids $\mathcal{E}_i$ define RCIS for the closed-loop system, if parameters $\varrho_{i,m} \in (0,1)$ exist for all $i, m \in \Theta$ with $p_{i,m} > 0$, such that:*

$$\mathcal{Q}_m - \varrho_{i,m}^{-1}(A_i + B_i K_i)\mathcal{Q}_i(A_i + B_i K_i)^\top - (1 - \varrho_{i,m})^{-1}E_i \geq 0. \tag{7.8}$$

Proof. Assume that i is the current Markov state and $x_\delta[k] \in \mathcal{E}_i$. Then, one has: $(A_i + B_i K_i) x_\delta[k] \in \mathcal{E}\big((A_i + B_i K_i)\mathcal{Q}_i(A_i + B_i K_i)^\top\big)$. In addition, it holds: $G_i \tilde{w}[k] \in \mathcal{E}(E_i)$. From the closed-loop dynamics (7.6) and the outer approximation of the Minkowski sum of these ellipsoids (Lemma 2.1), it follows that:

$$x_\delta[k+1] \in \mathcal{E}\left(\varrho_{i,m}^{-1}(A_i + B_i K_i)\mathcal{Q}_i(A_i + B_i K_i)^\top + (1 - \varrho_{i,m})^{-1}E_i\right). \tag{7.9}$$

The matrix inequality (7.8) requires that this ellipsoid is contained completely in $\mathcal{E}_m = \mathcal{E}(\mathcal{Q}_m)$ and, in consequence, $x_\delta[k+1] \in \mathcal{E}_m$. If this inequality holds for all possible transitions of the Markov chain ($p_{i,m} > 0$), implication (7.7) is satisfied and the ellipsoids $\mathcal{E}_i$ define RCIS for the closed-loop system. $\qquad\square$

The matrix inequalities (7.8) can be transformed into tractable BMI constraints. To this end, a controller parameterization with $\mathcal{G}_i \in \mathbb{R}^{n_x \times n_x}$ and $\mathcal{Y}_i \in \mathbb{R}^{n_u \times n_x}$, commonly used in SDP formulations for LTI systems (cf. [102]), is introduced:

$$K_i = \mathcal{Y}_i \mathcal{G}_i^{-1}, \quad i \in \Theta. \tag{7.10}$$

By inserting the controller parameterization (7.10) into (7.8), one obtains:

$$\mathcal{Q}_m - \varrho_{i,m}^{-1}\left(A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1}\right)\mathcal{Q}_i\left(A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1}\right)^\top - (1 - \varrho_{i,m})^{-1}E_i \geq 0. \tag{7.11}$$

According to Lemmata 2.3 - 2.5, the application of the Schur complement, a congruence transformation with $\mathcal{T} = \text{diag}(I, \mathcal{G}_i, E_i)$, and the approximation $\mathcal{G}_i^\top \mathcal{Q}_i^{-1} \mathcal{G}_i \geq \mathcal{G}_i^\top + \mathcal{G}_i - \mathcal{Q}_i$ lead to:

$$\begin{bmatrix} \mathcal{Q}_m & A_i\mathcal{G}_i + B_i\mathcal{Y}_i & E_i \\ \star & \varrho_{i,m}\left(\mathcal{G}_i^\top + \mathcal{G}_i - \mathcal{Q}_i\right) & \mathbf{0} \\ \star & \star & (1 - \varrho_{i,m})E_i \end{bmatrix} > 0. \tag{7.12}$$

The "$\star$" abbreviates transposed block matrices to define a symmetric matrix. This inequality is a BMI, if $\varrho_{i,m}$ is a variable. Otherwise, it is an LMI.

---

[3]A similar result for an RCIS independent of the Markov state is proposed in [83].

Remark 7.1. *This transformation only holds if $E_i = G_i \Lambda_w G_i^\top$ is invertible. If this is not the case, the following BMI formulation can be employed:*

$$\begin{bmatrix} \mathcal{Q}_m & A_i \mathcal{G}_i + B_i \mathcal{Y}_i & G_i \Lambda_w \\ \star & \varrho_{i,m} \left(\mathcal{G}_i^\top + \mathcal{G}_i - \mathcal{Q}_i \right) & 0 \\ \star & \star & (1 - \varrho_{i,m}) \Lambda_w \end{bmatrix} > 0. \tag{7.13}$$

Lemma 7.2. *The feasibility of (7.8) implies PMSS, if $E_i = G_i \Lambda_w G_i^\top > 0$ for all $i \in \Theta$.*

Proof. Consider a state $x_\delta[k]$ on the surface of the ellipsoid ($\|x_\delta[k]\|_{\mathcal{P}_{\theta_k}}^2 = 1$), where $\mathcal{P}_i = \mathcal{Q}_i^{-1}$ for all $i \in \Theta$. Due to the invariance condition (7.8), the following inequality holds for these states:

$$\left\| \left(A_{\theta_k} + B_{\theta_k} K_{\theta_k} \right) x_\delta[k] + G_{\theta_k} \tilde{w}[k] \right\|_{\mathcal{P}_{\theta_{k+1}}}^2 \le 1 = \|x_\delta[k]\|_{\mathcal{P}_{\theta_k}}^2 \qquad \forall \; \tilde{w}[k] \in \tilde{\mathcal{E}}_w. \tag{7.14}$$

Furthermore, the ellipsoids $\mathcal{E}_{w,i}$ are non-degenerate, since $E_i > 0$. In consequence, the ellipsoids $\mathcal{E}_{w,i}$ have a non-zero expansion in the directions of all state space components, and the dynamics without disturbances transfers the state $x_\delta[k]$ to the interior of the RCIS. Thus, it holds for all Markov states $i \in \Theta$:

$$\|(A_i + B_i K_i) x_\delta[k]\|_{\mathcal{P}_m}^2 < \|x_\delta[k]\|_{\mathcal{P}_i}^2 = 1 \qquad\qquad \forall \; p_{i,m} > 0 \tag{7.15}$$

$$\Rightarrow \quad \sum_{m=1}^{n_\theta} p_{i,m} \|(A_i + B_i K_i) x_\delta[k]\|_{\mathcal{P}_m}^2 < \sum_{m=1}^{n_\theta} p_{i,m} \|x_\delta[k]\|_{\mathcal{P}_i}^2 \tag{7.16}$$

$$\Leftrightarrow \quad \|(A_i + B_i K_i) x_\delta[k]\|_{\mathcal{T}_i(\mathcal{P})}^2 < \|x_\delta[k]\|_{\mathcal{P}_i}^2. \tag{7.17}$$

According to Lemma 2.9, this inequality also holds for all $x_\delta[k] \ne 0$ in the interior of the RCIS ($\|x_\delta[k]\|_{\mathcal{P}_{\theta_k}}^2 < 1$), since the closed-loop dynamics is linear. Obviously, there is a matrix $L \in \mathbb{S}^{>0}$ such that inequality (7.17) can be transformed into:

$$\|(A_i + B_i K_i) x_\delta[k]\|_{\mathcal{T}_i(\mathcal{P})}^2 - \|x_\delta[k]\|_{\mathcal{P}_i}^2 < -\|x_\delta[k]\|_L^2 \tag{7.18}$$

for all $x_\delta[k] \ne 0$. This inequality implies that the PMSS condition (2.67) holds, employing that $\bar{w}[k] = \mathbf{0}$ and $x_r = \mathbf{0}$ for the delta system. $\qquad\square$

The matrices $E_i = G_i \Lambda_w G_i^\top$ are not invertible for all values of G_i, i.e., PMSS is not guaranteed for all values of G_i. In addition, the selection of L is not obvious. Hence, an additional LMI condition is presented that can be used to guarantee PMSS for all stabilizable JMLS: Inserting the closed-loop dynamics (7.6) into the stability condition (2.67) results in the following matrix inequality:

$$(A_i + B_i K_i)^\top \mathcal{T}_i(\mathcal{P})(A_i + B_i K_i) - \mathcal{P}_i \le -L \quad \forall \; i \in \Theta. \tag{7.19}$$

The following matrix inequality follows by inserting the controller parameterization, $\mathcal{P}_i = \mathcal{Q}_i^{-1}$, and a new variable $\mathcal{Z} := L^{-1} \in \mathbb{S}_{n_x}^{>0}$ into (7.19):

$$(A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1})^\top \mathcal{T}_i(\mathcal{Q}^{-1})(A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1}) - \mathcal{Q}_i^{-1} \le -\mathcal{Z}^{-1}$$

$$\Rightarrow \mathcal{Q}_i^{-1} - \mathcal{X}^{-1} - \sum_{l=1}^{n_\theta} \sqrt{p_{i,l}} \left(A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1} \right)^\top \mathcal{Q}_l^{-1} \sqrt{p_{i,l}} \left(A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1} \right) \geq 0. \quad (7.20)$$

Applying the Schur complement (Lemma 2.3) results in:

$$\begin{bmatrix} \mathcal{Q}_i^{-1} & I & \begin{bmatrix} \sqrt{p_{i,1}} & \cdots & \sqrt{p_{i,n_\theta}} \end{bmatrix} \otimes (A_i + B_i \mathcal{Y}_i \mathcal{G}_i^{-1})^\top \\ \star & \mathcal{X} & \mathbf{0} \\ \star & \star & \mathrm{diag}(\mathcal{Q}_1, \ldots, \mathcal{Q}_{n_\theta}) \end{bmatrix} \geq 0, \quad (7.21)$$

where $\otimes$ denotes the Kronecker product. According to Lemmata 2.4 and 2.5, the congruence transformation of (7.21) with $\mathcal{T} = \mathrm{diag}(\mathcal{G}_i, I, I)$ and the application of the approximation $\mathcal{G}_i^\top \mathcal{Q}_i^{-1} \mathcal{G}_i \geq \mathcal{G}_i^\top + \mathcal{G}_i - \mathcal{Q}_i$ lead to:

$$\begin{bmatrix} \mathcal{G}_i^\top + \mathcal{G}_i - \mathcal{Q}_i & \mathcal{G}_i^\top & \begin{bmatrix} \sqrt{p_{i,1}} & \cdots & \sqrt{p_{i,n_\theta}} \end{bmatrix} \otimes (A_i \mathcal{G}_i + B_i \mathcal{Y}_i)^\top \\ \star & \mathcal{X} & \mathbf{0} \\ \star & \star & \mathrm{diag}(\mathcal{Q}_1, \ldots, \mathcal{Q}_{n_\theta}) \end{bmatrix} \geq 0. \quad (7.22)$$

The LMI (7.22) establishes a PMSS condition for the controller and the design of RCIS. In addition, the quadratic function $\|x_\delta[k]\|_{\mathcal{P}_{\theta_k}}^2$ defines a switched Lyapunov function for the closed-loop system.

Input and State Constraints

In this section, LMI conditions are presented guaranteeing that the controllers K_i satisfy the input and state constraints, if the current state $x_\delta[k]$ is in the RCIS. Since the controller and RCIS design are performed offline, the worst cases defined by $h_\mathrm{u} := \min_k(h_\mathrm{u}[k])$ and $h_\mathrm{x} := \min_k(h_\mathrm{x}[k])$ have to be considered.

The input is given by $u[k] = K_{\theta_k} x_\delta[k] + u_{\mathrm{r},\theta_k}$. Thus, the constraints are satisfied if:

$$H_\mathrm{u} \left(\mathcal{Y}_i \mathcal{G}_i^{-1} x_\delta[k] + u_{\mathrm{r},i} \right) \leq h_\mathrm{u}, \quad \forall \; x_\delta[k] \in \mathcal{E}_i, \quad i \in \Theta. \quad (7.23)$$

This inequality requires that the resulting input set (an ellipsoid with center $u_{\mathrm{r},i}$ and shape matrix $\mathcal{Y}_i \mathcal{G}_i^{-1} \mathcal{Q}_i (\mathcal{Y}_i \mathcal{G}_i^{-1})^\top$) is within the polytope defined by H_u and h_u. This is the case, if the maximum expansion of the input ellipsoid in all directions $H_{\mathrm{u},m}$ of the boundary planes is smaller than $h_{\mathrm{u},m}$. This condition can be formalized by the support function of the input ellipsoid (cf. Equation (2.6) on page 20):

$$\rho \left(H_{\mathrm{u},m}^\top \,|\, \mathcal{E} \left(u_{\mathrm{r},i}, \; \mathcal{Y}_i \mathcal{G}_i^{-1} \mathcal{Q}_i (\mathcal{Y}_i \mathcal{G}_i^{-1})^\top \right) \right) = H_{\mathrm{u},m} u_{\mathrm{r},i} + \left\| (\mathcal{Y}_i \mathcal{G}_i^{-1})^\top H_{\mathrm{u},m}^\top \right\|_{\mathcal{Q}_i} \leq h_{\mathrm{u},m}$$

$$\Leftrightarrow \; H_{\mathrm{u},m} \mathcal{Y}_i \mathcal{G}_i^{-1} \mathcal{Q}_i (\mathcal{Y}_i \mathcal{G}_i^{-1})^\top H_{\mathrm{u},m}^\top \leq \left(h_{\mathrm{u},m} - H_{\mathrm{u},m} u_{\mathrm{r},i} \right)^2 \quad (7.24)$$

for all $m \in \mathbb{N}_{n_{\mathrm{h,u}}}$ and $i \in \Theta$. The equivalence only holds if $h_{\mathrm{u},m} - H_{\mathrm{u},m} u_{\mathrm{r},i} \geq 0$, i.e., the input reference must be feasible according to the input constraints. Applying the Schur complement to inequality (7.24) results in the following matrix inequalities:

$$\begin{bmatrix} (h_{\mathrm{u},m} - H_{\mathrm{u},m} u_{\mathrm{r},i})^2 & H_{\mathrm{u},m} \mathcal{Y}_i \mathcal{G}_i^{-1} \\ \star & \mathcal{Q}_i^{-1} \end{bmatrix} \geq 0 \qquad \forall m \in \mathbb{N}_{n_{\mathrm{h,u}}}, \; i \in \Theta. \quad (7.25)$$

These matrix inequalities can be transformed into LMIs by a congruence transformation with $\mathscr{T} = \text{diag}(I, \mathscr{G}_i)$ and employing Lemma 2.5:

$$\begin{bmatrix} (h_{\text{u},m} - H_{\text{u},m}\,u_{\text{r},i})^2 & H_{\text{u},m}\mathscr{Y}_i \\ \star & \mathscr{G}_i^\top + \mathscr{G}_i - \mathscr{Q}_i \end{bmatrix} \geq 0 \quad \forall m \in \mathbb{N}_{n_{\text{h,u}}},\ i \in \Theta. \tag{7.26}$$

If these LMIs are satisfied, the controllers K_i satisfy the input constraints.

A very similar procedure can be used to consider the state constraints. Inserting the dynamics in the constraints results in:

$$H_{\text{x}}\left(\left(A_i + B_i\mathscr{Y}_i\mathscr{G}_i^{-1}\right)x_\delta[k] + G_i\,\tilde{w}[k] + x_{\text{r}}\right) \leq h_{\text{x}}, \quad \forall\,x_\delta[k] \in \mathcal{E}_i,\ \tilde{w}[k] \in \tilde{\mathcal{E}}_{\text{w}},\ i \in \Theta. \tag{7.27}$$

This condition requires that the sum of two ellipsoids (the one-step prediction of the states and disturbances) and the reference x_{r} is within the polytope defined by H_{x} and h_{x}. Analogously to the input constraints, this condition can be reformulated by considering all rows separately and using the support function of both ellipsoids:

$$\rho\left(H_{\text{x},m}^\top \,|\, \mathcal{E}(x_{\text{r}}, (A_i + B_i\mathscr{Y}_i\mathscr{G}_i^{-1})\mathscr{Q}_i(A_i + B_i\mathscr{Y}_i\mathscr{G}_i^{-1})^\top)\right) + \rho\left(H_{\text{x},m}^\top \,|\, \mathcal{E}(E_i)\right) \leq h_{\text{x},m}$$

$$\Leftrightarrow\ H_{\text{x},m}\,x_{\text{r}} + \left\|(A_i + B_i\mathscr{Y}_i\mathscr{G}_i^{-1})^\top H_{\text{x},m}^\top\right\|_{\mathscr{Q}_i} + \left\|H_{\text{x},m}^\top\right\|_{E_i} \leq h_{\text{x},m} \tag{7.28}$$

$$\Leftrightarrow\ H_{\text{x},m}(A_i + B_i\mathscr{Y}_i\mathscr{G}_i^{-1})\mathscr{Q}_i(A_i + B_i\mathscr{Y}_i\mathscr{G}_i^{-1})^\top H_{\text{x},m}^\top \leq \left(h_{\text{x},m} - H_{\text{x},m}\,x_{\text{r}} - \left\|H_{\text{x},m}^\top\right\|_{E_i}\right)^2$$

for all $m \in \mathbb{N}_{n_{\text{h,x}}}$ and $i \in \Theta$. The last equivalence only holds if:

$$h_{\text{x},m} - H_{\text{x},m}\,x_{\text{r}} - \left\|H_{\text{x},m}^\top\right\|_{E_i} \geq 0 \qquad \forall\,m \in \mathbb{N}_{n_{\text{h,x}}},\ i \in \Theta. \tag{7.29}$$

This condition is fulfilled, if the reference x_{r} satisfies the state constraints and the disturbances are not too large. Analogously to the input constraints, the inequalities (7.28) can be transformed into the following LMIs:

$$\begin{bmatrix} \left(h_{\text{x},m} - H_{\text{x},m}\,x_{\text{r}} - \left\|H_{\text{x},m}^\top\right\|_{E_i}\right)^2 & H_{\text{x},m}(A_i\mathscr{G}_i + B_i\mathscr{Y}_i) \\ \star & \mathscr{G}_i^\top + \mathscr{G}_i - \mathscr{Q}_i \end{bmatrix} \geq 0 \quad \forall\,m \in \mathbb{N}_{n_{\text{h,x}}},\ i \in \Theta. \tag{7.30}$$

If these LMIs are satisfied, the controllers K_i satisfy the state constraints. The consideration of the disturbance in (7.30) can be interpreted as constraint tightening (cf. Section 2.5) with the support function of the disturbance ellipsoids.

Remark 7.2. *The approach presented can be used to consider mixed input and state constraints* $H_{\text{x}}\,x[k] + H_{\text{u}}\,u[k] \leq h$. *Analogously to the procedures shown, it follows that:*

$$H_{\text{x}}\left(x_\delta[k] + x_{\text{r}}\right) + H_{\text{u}}\left(\mathscr{Y}_i\mathscr{G}_i^{-1}x_\delta[k] + u_{\text{r},i}\right) \leq h \qquad \forall\,x_\delta[k] \in \mathcal{E}_i,\ i \in \Theta \tag{7.31}$$

$$\Leftrightarrow\ \left(H_{\text{x}} + H_{\text{u}}\mathscr{Y}_i\mathscr{G}_i^{-1}\right)x_\delta[k] \leq h - H_{\text{u}}\,u_{\text{r},i} - H_{\text{x}}\,x_{\text{r}}, \qquad \forall\,x_\delta[k] \in \mathcal{E}_i,\ i \in \Theta. \tag{7.32}$$

Analogously to the input and state constraints, these inequalities can be transformed into LMIs:

$$\begin{bmatrix} (h_m - H_{\text{u},m}\,u_{\text{r},i} - H_{\text{x},m}\,x_{\text{r}})^2 & H_{\text{x},m}\mathscr{G}_i + H_{\text{u},m}\mathscr{Y}_i \\ \star & \mathscr{G}_i^\top + \mathscr{G}_i - \mathscr{Q}_i \end{bmatrix} \geq 0 \quad \forall\,m \in \mathbb{N}_{n_{\text{h}}},\ i \in \Theta \tag{7.33}$$

if $h - H_{\text{u}}\,u_{\text{r},i} - H_{\text{x}}\,x_{\text{r}} \geq \mathbf{0}$ *for all* $i \in \Theta$.

SDP Formulation

The constraints presented in the previous sections can be used to determine the controllers, RCIS, and a Lyapunov function for the closed-loop system offline:

Theorem 7.1. *Let the matrix sets $\mathscr{Q}, \mathscr{G}, \mathscr{Y}$, the matrix $\mathscr{Z}$, and the parameters $\varrho = \{\varrho_{i,m} \in (0,1): i, m \in \Theta\}$ be a solution of the following SDP:*

$$\max_{\mathscr{Q}, \mathscr{G}, \mathscr{Y}, \mathscr{Z}, \varrho} \sum_{i=1}^{n_\theta} \log \det(\mathscr{Q}_i) \tag{7.34a}$$

$$\text{s.t. } (7.12), (7.22), (7.26), (7.29), (7.30). \tag{7.34b}$$

Then, the controllers $K_i = \mathscr{Y}_i \mathscr{G}_i^{-1}$ with $i \in \Theta$ stabilize the JMLS in the PMSS sense, satisfy the constraints (7.1), and hold the closed-loop system in the RCIS, if $x[0] \in \mathcal{E}_{\theta_0}$. The closed-loop system guarantees a decrease of the Lyapunov function according to (2.67) with the Lyapunov matrices $\mathscr{P}_i = \mathscr{Q}_i^{-1}$ and a decrease rate defined by $L = \mathscr{Z}^{-1}$.

Proof. The robust invariance of the ellipsoids $\mathcal{E}_i$ and PMSS are guaranteed by (7.12) and (7.22), respectively. The satisfaction of the input constraints is ensured by (7.26). The state constraints are satisfied due to (7.29) and (7.30). The formal proof follows directly from the deductions presented in the previous sections. □

The cost function (7.34a) is used in order to maximize the overall volume of the RCIS (cf. [15, 74]). If ϱ is a variable, the SDP becomes a BMI problem. Otherwise, it is an LMI problem. Both formulations can be solved by commercial solvers, such as PENBMI or Mosek. In the SDP formulation (7.34), an auxiliary variable $\mathscr{G}_i$ that is independent from the Lyapunov matrices $\mathscr{P}_i$ is used to parameterize the controller (7.10). This approach is based on the ideas presented in [102] and is often used in SDP formulations for LTI systems. This formulation reduces the conservatism of the design approach significantly. However, similar formulations are rarely used for JMLS (see, e.g., [84]).

Remark 7.3. *A minimal convergence rate can be guaranteed by adding the LMI constraint $\mathscr{Z} \leq L_{\min}^{-1}$ with $L_{\min} \in \mathbb{S}_{n_x}^{>0}$ to the SDP (7.34).*

Remark 7.4. *If the disturbance set is too large, the SDP (7.34) has no solution. In this case, it is useful to find the largest disturbance sets for which the SDP (7.34) can be solved, i.e., for which RCIS exist. These sets can be approximated by introducing scaling factors $\alpha_{i,m} \in (0,1]$ for the error ellipsoids, such that its new shape matrices are $\alpha_{i,m}^2 E_i$ for all $i, m \in \Theta$, and maximizing the sum of all $\alpha_{i,m}$:*

$$\max_{\mathscr{Q}, \mathscr{G}, \mathscr{Y}, \mathscr{Z}, \varrho, \alpha} \sum_{i=1}^{n_\theta} \sum_{m=1}^{n_\theta} \alpha_{i,m}, \tag{7.35a}$$

$$\text{s.t. } 0 < \varrho_{i,m} < 1, \ 0 < \alpha_{i,m} \leq 1, \ (7.22), (7.26), (7.29), (7.30), \tag{7.35b}$$

$$\begin{bmatrix} \mathscr{Q}_m & A_i \mathscr{G}_i + B_i \mathscr{Y}_i & \alpha_{i,m} E_i \\ \star & \varrho_{i,m}\left(\mathscr{G}_i^\top + \mathscr{G}_i - \mathscr{Q}_i\right) & \mathbf{0} \\ \star & \star & (1 - \varrho_{i,m})E_i \end{bmatrix} \geq 0 \qquad \forall \ p_{i,m} > 0. \tag{7.35c}$$

The constraint tightening is not relaxed, since nonlinear constraints would result.

7.1.2. Formulation of the MPC Problem

Once the RCIS have been determined offline by solving the SDP (7.34), the MPC problems (6.22) and (6.49) presented for the CoE and EoC approach can be "robustified". The corresponding adaptions are presented in this section. For the MPC, the original system and not the delta system is considered.

State constraints: Since the MPC problem is solved at each time step, it is sufficient to formulate the state constraints only for the first prediction step:

$$H_{\mathrm{x}} x[k+1] = H_{\mathrm{x}} \left(A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} (\bar{w} + \tilde{w}[k]) \right) \leq h_{\mathrm{x}}[k], \qquad \forall \, \tilde{w}[k] \in \tilde{\mathcal{E}}_{\mathrm{w}}. \quad (7.36)$$

Since $\tilde{w}[k]$ is not known, this constraint cannot be implemented directly and a robust formulation must be employed. Analogously to the controller design, robust state constraints can be formulated by tightening the original constraints with the support functions of the error ellipsoid $\mathcal{E}_{\mathrm{w},\theta_k}$. To this end, the rows of the state constraints are considered separately. Then, the tightened linear state constraints are given by:

$$H_{\mathrm{x},m} \left(A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} \bar{w} \right) \leq h_{\mathrm{x},m}[k] - \left\| H_{\mathrm{x},m}^{\mathsf{T}} \right\|_{E_{\theta_k}} \qquad \forall \, m \in \mathbb{N}_{n_{\mathrm{h,x}}}. \quad (7.37)$$

Note that in this formulation, the time-dependent constraint $h_{\mathrm{x},m}[k]$ and not the worst-case is used. This formulation is a standard linear constraint that can be implemented directly to guarantee robust satisfaction of the original state constraints.

Invariance Constraints: Recursive feasibility of the MPC can be guaranteed, if the state $x_\delta[k]$ is kept within the ellipsoids $\mathcal{E}_i$ that define RCIS. Hence, it has to hold:

$$x_\delta[k+1] = A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} (\bar{w} + \tilde{w}[k]) - x_{\mathrm{r}} \in \mathcal{E}_{\theta_{k+1}} \qquad \forall \, \tilde{w}[k] \in \tilde{\mathcal{E}}_{\mathrm{w}} \quad (7.38)$$

at each time step k. This condition is non-deterministic since $\tilde{w}[k]$ and θ_{k+1} are unknown. It can be transformed into deterministic, robust constraints by formulating the condition for all possible Markov states θ_{k+1} and reducing the corresponding ellipsoids $\mathcal{E}_{\theta_{k+1}}$ using the disturbance ellipsoid $\mathcal{E}_{\mathrm{w},\theta_k}$:

$$A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} \bar{w} - x_{\mathrm{r}} \in \mathcal{E}_m \ominus \mathcal{E}_{\mathrm{w},\theta_k} \qquad \forall m \in \Theta \mid p_{\theta_k,m} > 0. \quad (7.39)$$

The resulting set is given by the Minkowski difference of $\mathcal{E}_m$ and $\mathcal{E}_{\mathrm{w},\theta_k}$. As shown in Section 2.2, this set may be a general nonlinear set. In order to avoid general nonlinear constraints, an ellipsoidal inner approximation $\mathcal{E}(\breve{\mathcal{Q}}_{\theta_k,m})$ of $\mathcal{E}_m \ominus \mathcal{E}_{\mathrm{w},\theta_k}$ with maximum volume is determined according to Lemma 2.2. A graphical illustration of these sets is shown in Fig. 2.1 on page 21. Then, the invariance condition (7.38) can be stated as a set of quadratic constraints:

$$\left\| A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} \bar{w} - x_{\mathrm{r}} \right\|_{\breve{\mathcal{Q}}_{\theta_k,m}^{-1}}^2 \leq 1 \qquad \forall \, m \in \Theta \mid p_{\theta_k,m} > 0. \quad (7.40)$$

The shape matrices $\breve{\mathcal{Q}}_{i,m}$ can be calculated offline for all $i, m \in \Theta$.

Stability Constraints: The stability condition (2.67) presented in Lemma 2.10 is a quadratic inequality. It can be used directly as a constraint for the MPC:

$$\left\| A_{\theta_k} x[k] + B_{\theta_k} u[k] + G_{\theta_k} \bar{w} - x_{\mathrm{r}} \right\|^2_{\mathcal{T}_{\theta_k}(\mathscr{P})} - \left\| x[k] - x_{\mathrm{r}} \right\|^2_{\mathscr{P}_{\theta_k}} \leq - \left\| x[k] - x_{\mathrm{r}} \right\|^2_L \qquad (7.41)$$

where the matrices $\mathscr{P}_i = \mathcal{Q}_i^{-1}$ and $L = \mathscr{Z}^{-1}$ are a solution of the SDP (7.34). Note that this constraint defines a contracting ellipsoidal set by which the states are forced to converge towards the reference x_{r}.

Optimization Problem: The robust MPC can be formulated by employing the robust state constraints (7.37) instead of expectation constraints and adding the invariance and stability constraints (7.40) and (7.41):

Theorem 7.2. *Let the matrices $L = \mathscr{Z}^{-1}$ and $\mathscr{P}_i = \mathcal{Q}_i^{-1}$ be a solution of* (7.34), *the matrices $\breve{\mathcal{Q}}_{i,m}$ be determined as described above, $\boldsymbol{H}_{\mathrm{u}} = I_N \otimes H_{\mathrm{u}}$, and $\boldsymbol{h}_{\mathrm{u}}[k] = \left[h_{\mathrm{u}}^{\top}[k] \cdots h_{\mathrm{u}}^{\top}[k+N-1] \right]^{\top}$. Then, the following optimization problem:*

$$\min_{\boldsymbol{u}[k]} \ \left\| \boldsymbol{u}[k] \right\|^2_{W[k]} + q[k]\,\boldsymbol{u}[k] \qquad (7.42\text{a})$$

$$\text{s. t. } \boldsymbol{H}_{\mathrm{u}} \boldsymbol{u}[k] \leq \boldsymbol{h}_{\mathrm{u}}[k], \qquad (7.42\text{b})$$

$$H_{\mathrm{x},m} \left(A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} \bar{w} \right) \leq h_{\mathrm{x},m}[k] - \left\| H_{\mathrm{x},m}^{\top} \right\|_{E_{\theta_k}} \qquad \forall \ m \in \mathbb{N}_{n_{\mathrm{h,x}}}, \ (7.42\text{c})$$

$$\left\| A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} \bar{w} - x_{\mathrm{r}} \right\|^2_{\breve{\mathcal{Q}}_{\theta_k,m}^{-1}} \leq 1 \qquad \forall \ p_{\theta_k,m} > 0, \ (7.42\text{d})$$

$$\left\| A_{\theta_k} x[k] + B_{\theta_k} u[\![0]\!] + G_{\theta_k} \bar{w} - x_{\mathrm{r}} \right\|^2_{\mathcal{T}_{\theta_k}(\mathscr{P})} \leq \left\| x[k] - x_{\mathrm{r}} \right\|^2_{\mathscr{P}_{\theta_k}-L}, \qquad (7.42\text{e})$$

which is solved in all times k, establishes a scheme of robust MPC that is recursively feasible, satisfies the constraints (7.1), *and stabilizes $\mathcal{S}_{\mathrm{LTI}}$ in the PMSS sense.*

Proof. The consideration of (7.42e) directly implies PMSS according to Lemma 2.10. The input constraints in (7.1) are equivalent to (7.42b). The state constraints in (7.1) are robustly satisfied due to the constraint tightening in (7.42c) with the error ellipsoid. By design of the controllers K_i according to Theorem 7.1, the input $u[k] = K_{\theta_k} x_\delta[k] + u_{\mathrm{r},\theta_k}$ is a feasible solution of the optimization problem (7.42), if $x[k] - x_r \in \mathcal{E}_{\theta_k}$. Since the constraints (7.42d) guarantee exactly this, the MPC is recursively feasible. □

The properties concerning robustness, stability, and recursive feasibility do not depend on the cost function formulation. Hence, the problem (7.42) can be used for both MPC approaches. If $W[k]$ and $q[k]$ are selected according to Theorem 6.2, the CoE is minimized; if $W[k]$ and $q[k]$ are selected according to Theorem 6.3, the EoC is minimized. Since the MPC does not establish a linear control law, the invariance constraint (7.42d) does not guarantee PMSS and the constraint (7.42e) is needed. The optimization problem (7.42) is a quadratically constrained QP (QCQP) and

can be solved efficiently with solvers like CPLEX. Hence, (7.42) establishes a robust MPC with relatively low computational effort.

Since the controllers K_i are linear, only state and input constraints that are symmetric with respect to the references $u_{r,i}$ and x_r are considered by the controllers (non-symmetric constraints are made symmetric by (7.26) and (7.30)). In the MPC problem (7.42), in contrast, asymmetric input and state constraints can be considered. In addition, in most cases, the controllers K_i transfer the states into sets that are significantly smaller than the feasible set of the MPC. Thus, the MPC is less conservative. This issue is illustrated by the following example:

Example 7.1. The difference of the MPC and the linear controller K_i are illustrated with a JMLS taken from [146] with:

$$A_1 = \begin{bmatrix} -0.8 & 1 \\ 0 & 0.8 \end{bmatrix}, A_2 = \begin{bmatrix} -0.8 & 1 \\ 0 & 1.2 \end{bmatrix}, A_3 = \begin{bmatrix} -0.8 & 1 \\ 0 & -0.4 \end{bmatrix}, B_1 = B_2 = B_3 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \quad (7.43)$$

The JMLS is considered without disturbances $w[k]$. The state and input constraints are defined by:

$$-2 \le u[k] \le 2 \qquad \text{and} \qquad \begin{bmatrix} -2.6 \\ -2.5 \end{bmatrix} \le x[k] \le \begin{bmatrix} 3 \\ 3 \end{bmatrix}. \quad (7.44)$$

The reference for the control is the origin. Note that the state constraints are asymmetric in this case. In Fig. 7.1, the state constraints, the ellipsoids $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ defining the RCIS, and the set to which the states $x[k] \in \mathcal{E}_{\theta_k}$ are transferred by the closed-loop dynamics $(A_{\theta_k} + B_{\theta_k} K_{\theta_k}) x[k]$ for $\theta_k = 3$ are shown.
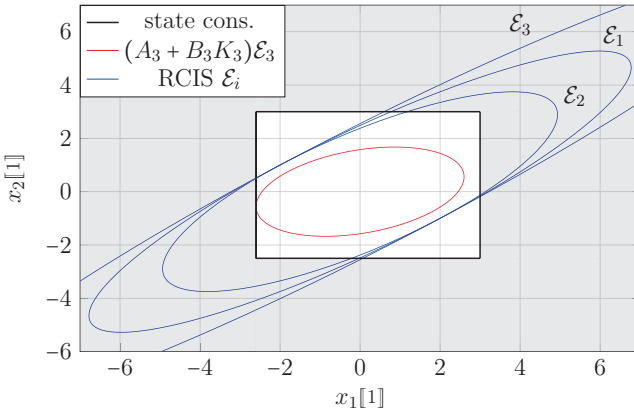
Figure 7.1.: Plot of the state constraints, the ellipsoids $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ defining the RCIS, and the set in which the state $x[k+1]$ is transferred to by K_3 for the JMLS (7.43).

The feasible set of the MPC is given by the union set of the state constraints and the RCIS $\mathcal{E}_i$. Obviously, this set is significantly larger than the set in which the controller K_3 transfers the states. To some extend, the conservatism is caused by the asymmetry of the state constraints. Due to linearity, the controllers require that $|x_1[1]| \leq 2.6$ and $|x_2[1]| \leq 2.5$. In contrast, the original constraints (7.44) are considered by the MPC. Depending on the initial state and the cost function, this may increase the control performance significantly. $\triangle$

It can be of interest to use references $u_{r,i}$ and x_r, that represent contradicting economic goals, such as minimal energy consumption and best reference tracking of product properties. Often, these references do not establish a steady state and the MPC is supposed to find an economically optimal steady state of the system implicitly by minimizing the cost function (*economic MPC*, cf. [93]). A similar behavior is desired for a distributed MPC setup, where one subsystem may deviate from its reference in order to minimize the overall costs. In these cases, an asymptotic convergence to x_r is not desired. To this end, the robust MPC can be used without the stability constraint (7.42e). In this case, the MPC remains recursively feasible, but may converge to any feasible state within the RCIS. This is illustrated in Chapter 12 with an economic MPC for a roller hearth furnace.

7.1.3. Simulation Results

This section presents two simulation studies that illustrate the properties of the robust MPC. The first simulation is supposed to demonstrate the stabilizing effects of the additional quadratic constraints. It is based on the setup published in [133]. The second example considers the stationary accuracy of the robust MPC.

Example 7.2. To demonstrate the properties of the robust MPC, the economic example JMLS presented in Example 6.3 on page 96 is considered. The dynamics is defined by (6.55). In addition, disturbances $w[k]$ are considered with

$$w[k] \sim \mathcal{N}\left(0, 10^{-3} \cdot \begin{bmatrix} 0.1 & -0.01 \\ -0.01 & 0.1 \end{bmatrix}\right) \quad \text{and} \quad G_i = I \;\; \forall \; i \in \Theta. \quad (7.45)$$

In order to obtain a bounded disturbance (set), only disturbances $w[k]$ contained in the confidence-ellipsoid $\mathcal{E}_\beta$ with a significance level of $\beta = 0.97$ are considered.

For this setup, the RCIS have been determined by solving (7.34) with the BMI solver PENBMI considering a minimum convergence rate of $L_{\min} = 0.1I$. The resulting Lyapunov matrices and L are shown below:

$$\mathscr{P}_1 = \begin{bmatrix} 5.2 & -5.2 \\ -5.2 & 6.7 \end{bmatrix}, \mathscr{P}_2 = \begin{bmatrix} 28.1 & -29.3 \\ -29.3 & 32.0 \end{bmatrix}, \mathscr{P}_3 = \begin{bmatrix} 25.8 & -27.6 \\ -27.6 & 30.9 \end{bmatrix}, L = 0.1I. \quad (7.46)$$

To demonstrate the properties of the additional constraints presented in this section, the JMLS controlled with the CoE approach presented in the previous
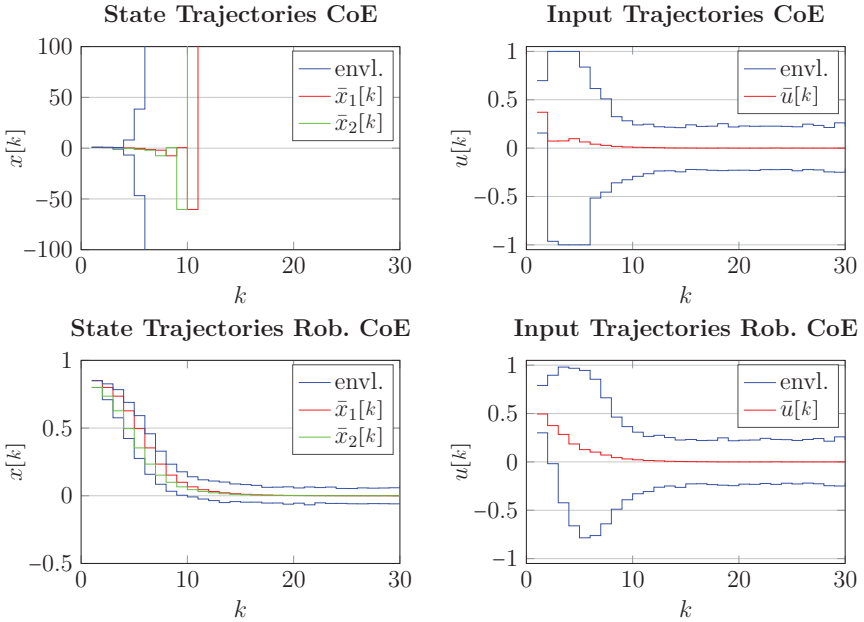
Figure 7.2.: Mean values and envelopes of states (left) and inputs (right) for 10,000 simulations of the JMLS controlled by the simple CoE MPC (top) and the robust CoE MPC (bottom).

Chapter and controlled by the robust MPC according to Theorem 7.2 (also minimizing the CoE) are compared. The simulation results of 10,000 simulation runs with initial state $x[0] = [0.85\ 0.8]^\top$, initial distribution $\mu[0] = [1/3\ 1/3\ 1/3]^\top$, horizon length $N = 6$, and cost matrices $Q = 5I$ and $R = 1$ are shown in Fig. 7.2. The results show that the CoE approach without robustness and stability constraints fails to stabilize the JMLS for some simulation runs, while the JMLS is stabilized for all simulation runs by the robust MPC proposed in this section. In 17.4 % of the simulation runs, the simple CoE approach resulted in an infeasible optimization problem, while the QCQP of the robust MPC was feasible in all simulations. This demonstrates that the additional constraints can be used to stabilize the JMLS robustly and ensure recursive feasibility.

The average computation time for the CoE approach solving a QP is 3.8 ms and 6.9 ms for the robust MPC employing the QCQP formulation. Hence, the QCQP formulation exhibits an increased computational effort. The absolute computation time, however, is still low.   △
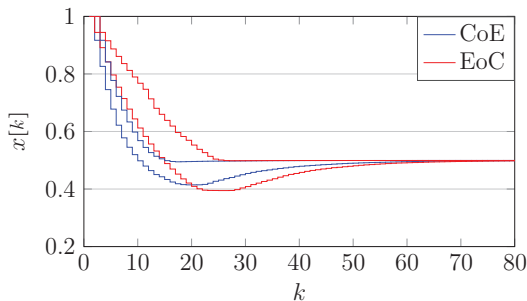
Figure 7.3.: State envelopes for both MPCs with stability constraint (7.42e).

Example 7.3. In Example 6.4, it has been shown that both the CoE as well as the EoC approach do not guarantee stationary accuracy of the reference x_r, if the input reference $u_{r,i}$ varies with the Markov state. This problem is solved by adding the stability constraint (7.42e) to the CoE or EoC optimization problem. To illustrate this, the same simulation setup as in in Example 6.4 is considered, but the stability constraint (7.42e) parameterized with (7.46) is added to the MPC problems. The results in Fig. 7.3 show that both MPC formulations converge to the reference $x_r = [0.5\ 0.5]^\top$. In addition, the convergence speed of the EoC approach is improved compared to the simple EoC approach considered in Example 6.4. $\triangle$

All in all, the examples illustrate the effectiveness of the constraints proposed. They can be used to guarantee stability and recursive feasibility for both the CoE and the EoC approach. In addition, the contracting stability constraints ensure stationary accuracy.

7.2. Robust MPC for Time-Variant JMLS

A robust MPC design approach for time-variant JMLS

$$\mathcal{S}_l := \begin{cases} x[k+1] & = A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[k] + G_{\theta_k}[k]\,w[k] \\ \mathcal{M} & = \big(\Theta, P[k], \mu[k]\big) \end{cases} \tag{7.47}$$

is considered in this section. Note that this JMLS is the same as in (6.1). Hence, the formulation of the CoE and the EoC cost function as quadratic functions can be adopted from Chapter 6. The general idea of the robust MPC design presented in the previous section (design ellipsoidal RCIS, controllers, and a Lyapunov function offline, and add invariance and stability constraints to the MPC problem) can be used also for time-variant JMLS. Only the RCIS design and the constraint formulation have to be adopted to some extend. The approach presented in this section is based on results published in [134].

113

Design of the RCIS and the Controller

The controllers and corresponding RCIS have to guarantee robustness, PMSS, and constraint satisfaction for all possible values of the system matrices and transition probabilities. Thus, the offline design is only possible if these matrices take values from a known bounded set. To this end, it is assumed that matrices $A_i[k]$, $B_i[k]$, $G_i[k]$, and $P[k]$ are elements of matrix polytopes $\mathbb{A}_i$, $\mathbb{B}_i$, $\mathbb{G}_i$, and $\mathbb{P}$ for all $i \in \Theta$. These polytopes are defined by convex combinations of the vertices $A_{i,[l_A]}$, $B_{i,[l_B]}$, $G_{i,[l_G]}$, and $P_{[l_P]}$ with $l_A \in \mathbb{N}_{n_{A,i}}$, $l_B \in \mathbb{N}_{n_{B,i}}$, $l_G \in \mathbb{N}_{n_{G,i}}$, and $l_P \in \mathbb{N}_{n_P}$:

$$A_i[k] \in \mathbb{A}_i = \left\{ A_i[k] = \sum_{l_A=1}^{n_{A,i}} \mathfrak{a}_{i,l_A}[k]\, A_{i,[l_A]} : \quad \mathfrak{a}_{i,l_A}[k] \geq 0, \quad \sum_{l_A=1}^{n_{A,i}} \mathfrak{a}_{i,l_A}[k] = 1 \right\}. \quad (7.48)$$

Analogous definitions are used for $\mathbb{B}_i$, $\mathbb{G}_i$, and $\mathbb{P}$ with the coefficients $\mathfrak{b}_{i,l_B}[k]$, $\mathfrak{g}_{i,l_G}[k]$, and $\mathfrak{p}_{l_P}[k]$. The elements of $P_{[l_P]}$, i.e., the vertices for the transition probabilities, are denoted by $p_{i,m,[l_P]}$. Similar setups are used in robust MPC approaches for JMLS with parameter uncertainties (see, e.g., [84, 103]). However, in all of these approaches, it is assumed that the system matrices are unknown. This results in relatively conservative controllers and RCIS. In contrast, it is assumed here that the system matrices $A_i[k]$, $B_i[k]$, $G_i[k]$, and $P[k]$ are known. This is motivated by the fact that the JMLS may result from a linearization of a nonlinear JMS (as presented in Chapter 8), or that the system parameters can be measured.

As for the time-invariant case, RCIS $\mathcal{E}_i = \mathcal{E}(\mathcal{Q}_i)$ and controllers are determined offline for the delta system. To reduce the conservatism of the design approach, a time-variant controller $K_i[k]$ depending on the actual values of the system matrices is employed. Motivated by controller design approaches for linear parameter varying (LPV) systems, the following controller parameterization is considered:

$$K_i[k] = \mathscr{Y}_i[k] \left(\mathscr{G}_i[k] \right)^{-1} = \left(\sum_{l_A=1}^{n_{A,i}} \mathfrak{a}_{i,l_A}[k] \mathscr{Y}_{i,[l_A]} \right) \cdot \left(\sum_{l_B=1}^{n_{B,i}} \mathfrak{b}_{i,l_B}[k] \mathscr{G}_{i,[l_B]} \right)^{-1} \quad (7.49)$$

with $\mathscr{Y}_{i,[l_A]} \in \mathbb{R}^{n_u \times n_x}$ and $\mathscr{G}_{i,[l_B]} \in \mathbb{R}^{n_x \times n_x}$ for all $l_A \in \mathbb{N}_{n_{A,i}}$, $l_B \in \mathbb{N}_{n_{B,i}}$, and $i \in \Theta$. Once the vertices for the controller parameters $\mathscr{Y}_{i,[l_A]}$ and $\mathscr{G}_{i,[l_B]}$ have been determined offline, the controller can be determined at each time step by (7.49). Note that the controller parameters $\mathscr{Y}_i[k]$ and $\mathscr{G}_i[k]$ depend on the values of $A_i[k]$ and $B_i[k]$. This parameterization reduces the conservatism of the RCIS design compared to the cited approaches employing time-invariant controllers.

The closed-loop system dynamics for the delta system with (7.49) is:

$$x_\delta[k+1] = \left(A_{\theta_k}[k] + B_{\theta_k}[k] \mathscr{Y}_i[k] \left(\mathscr{G}_i[k] \right)^{-1} \right) x_\delta[k] + G_{\theta_k}[k]\, \tilde{w}[k]. \quad (7.50)$$

For the sake of a brief notation, the effects of the disturbances $G_{\theta_k}[k]\, w[k]$ are aggregated in time-invariant disturbance ellipsoids $\mathcal{E}_{w,i} := \mathcal{E}(E_i)$. These ellipsoids can be determined by, e.g., employing approaches for the calculation of an outer approximation of the union of the disturbance ellipsoids for all vertices of $G_{i,[l_G]}$. For example, the SDP based approaches presented in [23, Section 3.7.1] can be used.

SDP Formulation   Concepts from LPV controller design (see, e.g., [23, 114] and the references therein) are employed to state an SDP that can be used to determine the RCIS and the vertices of the controller parameters: For a suitable controller parameterization, it is sufficient to design the controller vertices such that the desired properties (PMSS, constraint satisfaction, and invariance) hold for all combinations of the system matrix vertices. Due to the linearity of the system, the properties hold for all system matrices (determined by convex combinations of the corresponding vertex matrices). This idea is formalized in the following Lemma:

Lemma 7.3. *Let a matrix inequality:*

$$\mathscr{L}\big(A_i[k], B_i[k], \mathscr{G}_i[k], \mathscr{Y}_i[k], \mathscr{Q}_i, \mathscr{X}, P[k], \varrho_{i,m}\big) \geq 0 \tag{7.51}$$

be given, where all elements are linear in the arguments or contain the following products $A_i[k]\mathscr{G}_i[k]$, $B_i[k]\mathscr{Y}_i[k]$, $p_{i,m}[k]\big(A_i[k]\mathscr{G}_i[k] + B_i[k]\mathscr{Y}_i[k]\big)$, $p_{i,m}[k]\mathscr{Q}_m$, $\varrho_{i,m}\mathscr{G}_i[k]$, and $\varrho_{i,m}\mathscr{Q}_i$ with $i, m \in \Theta$. Then, the matrix inequality (7.51) holds if:

$$\mathscr{L}\big(A_{i,[l_\mathrm{A}]}, B_{i,[l_\mathrm{B}]}, \mathscr{G}_{i,[l_\mathrm{A}]}, \mathscr{Y}_{i,[l_\mathrm{A}]}, \mathscr{Q}_i, \mathscr{X}, P_{[l_\mathrm{P}]}, \varrho_{i,m}\big) \geq 0 \tag{7.52}$$

is satisfied for all vertex combinations $l_\mathrm{A} \in \mathbb{N}_{n_{\mathrm{A},i}}$, $l_\mathrm{B} \in \mathbb{N}_{n_{\mathrm{B},i}}$, and $l_\mathrm{P} \in \mathbb{N}_{n_\mathrm{P}}$.

Proof. Note that none of the products, that are possibly contained in $\mathscr{L}(\cdot)$, have factors that depend on the same coefficient $\mathfrak{a}_{i,l_\mathrm{A}}[k]$, $\mathfrak{b}_{i,l_\mathrm{B}}[k]$, or $\mathfrak{p}_{l_\mathrm{P}}[k]$. Employing the fact that the sums of the coefficients $\mathfrak{a}_{i,l_\mathrm{A}}[k]$, $\mathfrak{b}_{i,l_\mathrm{B}}[k]$, or $\mathfrak{p}_{l_\mathrm{P}}[k]$ over l_A, l_B, or l_P equal 1, all elements $\mathscr{L}_{n,m}[k]$ of $\mathscr{L}(\cdot)$ can be written as a linear combination:

$$\mathscr{L}_{n,m}[k] = \sum_{l_\mathrm{A}=1}^{n_{\mathrm{A},i}} \sum_{l_\mathrm{B}=1}^{n_{\mathrm{B},i}} \sum_{l_\mathrm{P}=1}^{n_\mathrm{P}} \mathfrak{a}_{i,l_\mathrm{A}}[k]\, \mathfrak{b}_{i,l_\mathrm{B}}[k]\, \mathfrak{p}_{l_\mathrm{P}}[k]\, \mathscr{L}_{n,m}(l_\mathrm{A}, l_\mathrm{B}, l_\mathrm{P}). \tag{7.53}$$

For example, $A_i[k]\mathscr{G}_i[k] = \sum_{l_\mathrm{A}=1}^{n_{\mathrm{A},i}} \sum_{l_\mathrm{B}=1}^{n_{\mathrm{B},i}} \sum_{l_\mathrm{P}=1}^{n_\mathrm{P}} \mathfrak{a}_{i,l_\mathrm{A}}[k]\, \mathfrak{b}_{i,l_B}[k]\, \mathfrak{p}_{l_\mathrm{P}}[k]\, A_{i,[l_\mathrm{A}]}\, \mathscr{G}_{i,[l_\mathrm{B}]}$. The same holds for all time-invariant elements. For the overall matrix inequality, it holds:

$$\mathscr{L}\big(A_i[k], B_i[k], \mathscr{G}_i[k], \mathscr{Y}_i[k], \mathscr{Q}_i, \mathscr{X}, P[k], \varrho_{i,m}\big) \tag{7.54}$$
$$= \sum_{l_\mathrm{A}=1}^{n_{\mathrm{A},i}} \sum_{l_\mathrm{B}=1}^{n_{\mathrm{B},i}} \sum_{[l_\mathrm{P}]=1}^{n_\mathrm{P}} \mathfrak{a}_{i,l_\mathrm{A}}[k]\, \mathfrak{b}_{i,l_\mathrm{B}}[k]\, \mathfrak{p}_{l_\mathrm{P}}[k]\, \mathscr{L}\big(A_{i,[l_\mathrm{A}]}, B_{i,[l_\mathrm{B}]}, \mathscr{G}_{i,[l_\mathrm{B}]}, \mathscr{Y}_{i,[l_\mathrm{A}]}, \mathscr{Q}_i, \mathscr{X}, P_{[l_\mathrm{P}]}, \varrho_{i,m}\big).$$

Since $\mathfrak{a}_{i,l_\mathrm{A}}[k] \geq 0$, $\mathfrak{b}_{i,l_\mathrm{B}}[k] \geq 0$, and $\mathfrak{p}_{l_\mathrm{P}}[k] \geq 0$, the matrix inequality (7.51) holds provided that the corresponding matrix inequality (7.52) holds for all vertices. $\square$

Based on this result, the SDP (7.34) can be used for time-variant JMLS and controllers if the constraints are considered for all vertices:

Theorem 7.3. *Let the matrix sets $\mathscr{Q} = \{\mathscr{Q}_i : i \in \Theta\}$, $\mathscr{Y} = \{\mathscr{Y}_{i,[l_\mathrm{A}]} : l_\mathrm{A} \in \mathbb{N}_{n_{\mathrm{A},i}}, i \in \Theta\}$, $\mathscr{G} = \{\mathscr{G}_{i,[l_\mathrm{B}]} : l_\mathrm{B} \in \mathbb{N}_{n_{\mathrm{B},i}}, i \in \Theta\}$, the matrix $\mathscr{X}$, and the parameters $\varrho = \{\varrho_{i,m} \in (0,1) : i, m \in \Theta\}$ be a solution of the SDP problem:*

$$\max_{\mathscr{Q}, \mathscr{G}, \mathscr{Y}, \mathscr{X}, \varrho} \sum_{i=1}^{n_\theta} \log \det(\mathscr{Q}_i) \qquad \text{s.t.} \tag{7.55a}$$

$$\begin{bmatrix} \mathcal{Q}_m & A_{i,[l_A]}\mathcal{G}_{i,[l_B]} + B_{i,[l_B]}\mathcal{Y}_{i,[l_A]} & E_i \\ \star & \varrho_{i,m}\big(\mathcal{G}_{i,[l_B]}^{\top} + \mathcal{G}_{i,[l_B]} - \mathcal{Q}_i\big) & \mathbf{0} \\ \star & \star & (1-\varrho_{i,m})E_i \end{bmatrix} \geq 0, \tag{7.55b}$$

$$\begin{bmatrix} \mathcal{G}_{i,[l_B]}^{\top} + \mathcal{G}_{i,[l_B]} - \mathcal{Q}_i & \mathcal{G}_{i,[l_B]}^{\top} & \big[p_{i,1,[l_P]} \cdots p_{i,n_\theta,[l_P]}\big] \otimes \big(A_{i,[l_A]}\mathcal{G}_{i,[l_B]} + B_{i,[l_B]}\mathcal{Y}_{i,[l_A]}\big)^{\top} \\ \star & \mathscr{L} & \mathbf{0} \\ \star & \star & \mathrm{diag}(p_{i,1,[l_P]}\mathcal{Q}_1, \cdots, p_{i,n_\theta,[l_P]}\mathcal{Q}_{n_\theta}) \end{bmatrix} \geq 0, \tag{7.55c}$$

$$\begin{bmatrix} \big(h_{\mathrm{u},l_u} - H_{\mathrm{u},l_u}\,u_{\mathrm{r},i}\big)^2 & H_{\mathrm{u},l_u}\,\mathcal{Y}_{i,[l_A]} \\ \star & \mathcal{G}_{i,[l_B]}^{\top} + \mathcal{G}_{i,[l_B]} - \mathcal{Q}_i \end{bmatrix} \geq 0, \tag{7.55d}$$

$$\begin{bmatrix} \big(h_{\mathrm{x},l_x} - H_{\mathrm{x},l_x}\,x_{\mathrm{r}} - \big\|H_{\mathrm{x},l_x}^{\top}\big\|_{E_i}\big)^2 & H_{\mathrm{x},l_x}\big(A_{i,[l_A]}\mathcal{G}_{i,[l_B]} + B_{i,[l_B]}\mathcal{Y}_{i,[l_A]}\big) \\ \star & \mathcal{G}_{i,[l_B]}^{\top} + \mathcal{G}_{i,[l_B]} - \mathcal{Q}_i \end{bmatrix} \geq 0, \tag{7.55e}$$

$$h_{\mathrm{u}} - H_{\mathrm{u}}\,u_{\mathrm{r},i} \geq \mathbf{0}, \qquad h_{\mathrm{x},l_x} - H_{\mathrm{x},l_x}\,x_{\mathrm{r}} - \big\|H_{\mathrm{x},l_x}^{\top}\big\|_{E_i} \geq 0 \tag{7.55f}$$

for all $i,m \in \Theta$, $l_A \in \mathbb{N}_{n_{A,i}}$, $l_B \in \mathbb{N}_{n_{B,i}}$, $l_P \in \mathbb{N}_{n_P}$, $l_x \in n_{\mathrm{h,x}}$, and $l_u \in n_{\mathrm{h,u}}$. Then, the controllers (7.49) stabilize the JMLS in the PMSS sense, satisfy the constraints (7.1), and hold the closed-loop system in the RCIS, if $x[0] \in \mathcal{E}_{\theta_0}$. The closed-loop system guarantees a decrease of the Lyapunov function according to (2.67) with Lyapunov matrices $\mathscr{P}_i = \mathcal{Q}_i^{-1}$ and a decrease rate defined by $L = \mathscr{L}^{-1}$.

Proof. The closed-loop dynamics (7.50) differs from that of the time-invariant case in (7.6) only in the time arguments. Thus, the derivations in Section 7.1.1 still hold and the ellipsoids $\mathcal{E}(\mathcal{Q}_i)$ define RCIS if $\varrho_{i,m} \in (0,1)$ exist such that for all $i,m \in \Theta$:

$$\begin{bmatrix} \mathcal{Q}_m & A_i[k]\mathcal{G}_i[k] + B_i[k]\mathcal{Y}_i[k] & E_i \\ \star & \varrho_{i,m}\big(\mathcal{G}_i^{\top}[k] + \mathcal{G}_i[k] - \mathcal{Q}_i\big) & \mathbf{0} \\ \star & \star & (1-\varrho_{i,m})E_i \end{bmatrix} \geq 0. \tag{7.56}$$

According to Lemma 7.3, this condition is satisfied if (7.55b) holds for all vertices.

Applying the reformulation of the stability condition presented in Section 7.1.1 and a congruence transformation[4] with $\mathscr{T} = \mathrm{diag}\big(I, I, \sqrt{p_{i,1}[k]}, \cdots, \sqrt{p_{i,n_\theta}[k]}\big)$ according to Lemma 2.4 results in the following LMI condition for PMSS:

$$\begin{bmatrix} \mathcal{G}_i^{\top}[k] + \mathcal{G}_i[k] - \mathcal{Q}_i & \mathcal{G}_i^{\top}[k] & \big[p_{i,1}[k] \cdots p_{i,n_\theta}[k]\big] \otimes \big(A_i[k]\mathcal{G}_i[k] + B_i[k]\mathcal{Y}_i[k]\big)^{\top} \\ \star & \mathscr{L} & \mathbf{0} \\ \star & \star & \mathrm{diag}(p_{i,1}[k]\mathcal{Q}_1, \cdots, p_{i,n_\theta}[k]\mathcal{Q}_{n_\theta}) \end{bmatrix} \geq 0. \tag{7.57}$$

According to Lemma 7.3, this stability condition is satisfied if (7.55c) holds for all vertices. Hence, PMSS is guaranteed and the decrease of the Lyapunov function is defined by $L = \mathscr{L}^{-1}$. The formulation of the input and state constraints as LMI constraints according to Section 7.1.1 applies also for time-variant JMLS. Hence, the resulting LMIs can be stated with the time-dependent matrices $A_i[k], B_i[k], \mathcal{G}_i[k]$, and $\mathcal{Y}_i[k]$. Then, Lemma 7.3 applies and the state and input constraints are satisfied, if the LMIs (7.55d) and (7.55e) hold for all vertices. $\qquad\square$

---

[4]The transformation can be used also if $p_{i,m}[k] = 0$, since the corresponding off-diagonal elements $\sqrt{p_{i,m}[k]}\big(A_i[k]\mathcal{G}_i[k] + B_i[k]\mathcal{Y}_i[k]\big)$ are also zero and the diagonal elements $\mathcal{Q}_m$ are positive definite.

Considering the constraints (7.55b) - (7.55e), the motivation for the controller parameterization according to (7.49) becomes obvious. Employing this parameterization guarantees that no quadratic terms of $\mathfrak{a}_{i,l_A}[k]$ and $\mathfrak{b}_{i,l_B}[k]$ occur in the time-dependent elements of the LMIs and BMIs. Hence, Lemma 7.3 can be applied. Thus, the SDP (7.55) is still a BMI problem and can be used to determine the RCIS. For larger numbers of matrix vertices and larger system dimensions, the computational effort is significantly higher than for time-invariant JMLS. This, however, is acceptable for most applications, since the SDP is solved offline.

Formulation of the MPC Problem

Once the error ellipsoids and the RCIS are determined offline, the formulation of the MPC is similar to the time-invariant case (see Section 7.1.2). The calculation of the matrices $W[k]$ and $q[k]$ according to Chapter 6 can be used without changes. The procedure for the state constraint tightening presented in Section 7.1.2 applies also to time-variant JMLS. To reduce the conservatism, the current value of the matrix $G_i[k]$ can be used instead of the over-approximation by E_i:

$$H_{x,l_x} \left(A_{\theta_k}[k]\, x[k] + B_{\theta_k}[k]\, u[\![0]\!] + G_{\theta_k}[k]\, \bar{w} \right) \le h_{x,l_x}[k] - \left\| G_{\theta_k}^{\top}[k] H_{x,l_x}^{\top} \right\|_{\Lambda_w} \quad \forall\, l_x \in \mathbb{N}_{n_{h,x}}. \quad (7.58)$$

The RCIS $\mathcal{E}_i$ and the disturbance ellipsoids $\mathcal{E}_{w,i}$ are time-invariant. Thus, the formulation of the invariance constraints (employing the Minkowski difference $\mathcal{E}_m \ominus \mathcal{E}_{w,\theta_k}$) can be used as described in Section 7.1.2. In addition, the stability condition (2.67) directly applies for time-variant system matrices. All in all, the MPC can be formulated as presented in Section 7.1.2 by replacing the time-invariant system matrices with its current values:

Theorem 7.4. *Let the matrices* $L = \mathscr{X}^{-1}$ *and* $\mathscr{P}_i = \mathscr{Q}_i^{-1}$ *be a solution of* (7.55)*, the matrices* $\breve{\mathscr{Q}}_{i,m}$ *be determined as described in Section 7.1.2,* $\boldsymbol{H}_u = I_N \otimes H_u$*, and* $\boldsymbol{h}_u[k] = \left[h_u^{\top}[k] \; \cdots \; h_u^{\top}[k+N-1] \right]^{\top}$*. Then, the following optimization problem:*

$$\min_{\boldsymbol{u}[k]} \; \|\boldsymbol{u}[k]\|_{W[k]}^2 + q[k]\,\boldsymbol{u}[k] \tag{7.59a}$$

$$\text{s. t. } \boldsymbol{H}_u \boldsymbol{u}[k] \le \boldsymbol{h}_u[k], \tag{7.59b}$$

$$H_{x,l_x} \left(A_{\theta_k}[k]\, x[k] + B_{\theta_k}[k]\, u[\![0]\!] + G_{\theta_k}[k]\, \bar{w} \right) \le h_{x,l_x}[k] - \left\| G_{\theta_k}^{\top}[k] H_{x,l_x}^{\top} \right\|_{\Lambda_w}, \tag{7.59c}$$

$$\left\| A_{\theta_k}[k]\, x[k] + B_{\theta_k}[k]\, u[\![0]\!] + G_{\theta_k}[k]\, \bar{w} - x_r \right\|_{\breve{\mathscr{Q}}_{\theta_k,m}^{-1}}^2 \le 1, \tag{7.59d}$$

$$\left\| A_{\theta_k}[k]\, x[k] + B_{\theta_k}[k]\, u[\![0]\!] + G_{\theta_k}[k]\, \bar{w} - x_r \right\|_{\mathcal{T}_{\theta_k}(\mathscr{P},0)}^2 \le \|x[k] - x_r\|_{\mathscr{P}_{\theta_k}-L}^2 \tag{7.59e}$$

with $l_x \in \mathbb{N}_{n_{h,x}}$ *and* $m \in \Theta$*, such that* $p_{\theta_k,m} > 0$*, to be solved in any time* k*, establishes a robust MPC that is recursively feasible, satisfies the constraints* (7.1)*, and guarantees PMSS of the closed-loop system.*

Proof. The proof follows directly from the proofs of Theorem 7.2 and 7.3. $\qquad \square$
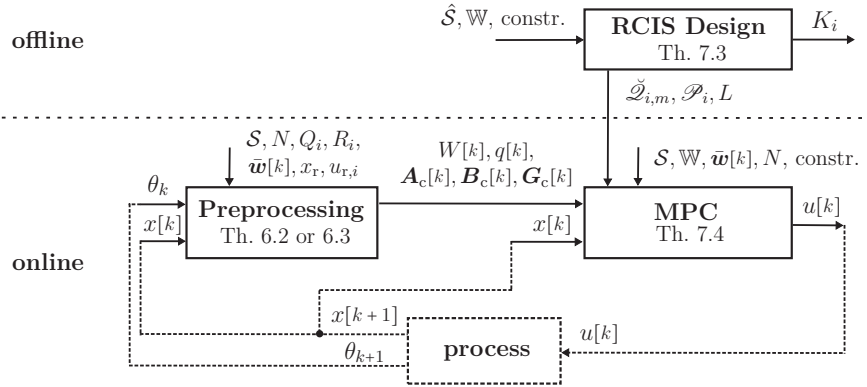
Figure 7.4.: Sketch of the design and implementation of the robust MPC scheme for time-variant JMLS and the corresponding theorems. The abbreviation "constr." includes $H_{\mathrm{x}}, H_{\mathrm{u}}, h_{\mathrm{x}},$ and h_{u}. The polytopic over-approximation of the system $\mathcal{S}$ is denoted by $\hat{\mathcal{S}}$.

The resulting optimization problem is a QCQP with the same dimensions as for the time-invariant case presented in Theorem 7.2. Hence, the computation times for the MPC for time-invariant and time-variant JMLS are very similar. The memory required for storing the quantities computed offline ($\breve{\mathcal{Q}}_{i,m}, \mathscr{P}_i, L$) is relatively small and increases with n_θ^2 and n_{x}^2.

The whole design and implementation procedure is illustrated in Fig. 7.4. The MPC formulation uses the shape matrices of the tightened RCIS ($\breve{\mathcal{Q}}_{i,m}$), the Lyapunov matrices $\mathscr{P}_i$, and the matrix L determined offline. The MPC (Theorem 7.4) can be applied independently of the cost function parameterization. The cost matrices $W[k]$ and $q[k]$ can be calculated for the CoE or EoC cost function according to Theorem 6.2 or 6.3, respectively. The auxiliary controllers K_i can be interpreted as a switched robust LPV controller. They are used offline for the design of the RCIS, such that they would account for all disturbances, time-variant behavior, and Markov state transitions possibly encountered online. Hence, the conservatism of the RCIS and controller design is larger than for the time-invariant case. In the MPC formulation, however, the time-variant behavior and the Markov state transitions are considered directly in the cost function and the constraints, and the controllers K_i are not used (cf. Fig. 7.4). Thus, the only source of increased conservatism compared to the time-invariant case is a possibly smaller RCIS resulting from the offline computation. Hence, for time-variant JMLS, the advantage of the robust MPC over the robust controllers K_i is significantly larger.

As stated in the introduction of this section, the presented MPC design approach can be employed, if the time-variant JMLS results from a linearization of a nonlinear JMS. This approach is considered in the next chapter.

8. Model Predictive Control for Jump Markov Nonlinear Systems

The dynamics of real production systems is often nonlinear. Thus, this Chapter presents an approach to apply the robust MPC formulation for JMLS presented in the previous Chapter to nonlinear JMS:

$$\mathcal{S} := \begin{cases} x[k+1] &= f_{\theta_k}\big(x[k], u[k], w[k]\big) \\ \mathcal{M} &= \big(\Theta, P[k], \mu[0]\big) \end{cases} . \tag{8.1}$$

As in Chapter 7, separate linear state and input constraints (7.1) are considered[1]. It is assumed that the constraints define compact time-invariant sets $\mathbb{U}$ and $\mathbb{X}$ that contain the references $u_{\mathrm{r},i}$ and x_{r}. The disturbances $w[k]$ are contained in the compact set $\mathbb{W}$. For notational convenience, the following abbreviation is introduced:

$$\xi[k] := \begin{bmatrix} x^\top[k] & u^\top[k] & w^\top[k] \end{bmatrix}^\top . \tag{8.2}$$

The corresponding space is given by $\Xi := \mathbb{X} \times \mathbb{U} \times \mathbb{W}$. It is compact, since the input, state, and disturbance sets are compact. As a result of this definition, the following abbreviation can be used for the system dynamics: $f_i\big(x[k], u[k], w[k]\big) = f_i\big(\xi[k]\big)$.

Assumption 8.1. *It is assumed that the reference $\xi_{\mathrm{r},i} := \begin{bmatrix} x_{\mathrm{r}}^\top & u_{\mathrm{r},i}^\top & \bar{w}^\top \end{bmatrix}^\top$ is a steady state of the JMS according to* (2.56), *i.e.,* $x_{\mathrm{r}} = f_i\big(\xi_{\mathrm{r},i}\big)$ *for all* $i \in \Theta$.

As stated in the introduction of this part (see Section 5.1), the most promising approach to formulate a robust MPC for nonlinear JMS with low computation time is to linearize the JMS at each time k and employ the robust MPC approach proposed in the previous chapter. Hence, the procedure is also divided into an offline and an online part. First, RCIS, a Lyapunov function, and corresponding quadratic constraints are determined offline. To guarantee robustness, an extended disturbance set including both the disturbance set $\mathbb{W}$ and the linearization errors is employed. Then, the following procedure is performed online at each time k:

1. Measure the current state $x[k]$ and the Markov state θ_k. Determine the expected values of the disturbances $\bar{w}[\![j]\!]$ for all $j \in \mathbb{N}_{N-1}^0$ from historical data and from the disturbance models $\hat{\nu}$ (see Section 3.2.2).

---

[1]Mixed input and state constraints can be considered as presented in Chapter 7.

2. Linearize the JMS $\mathcal{S}$ at $\xi_l[k]$ defined by the current state, the predicted inputs, and the expected value of the disturbance. The result is a JMLS $\mathcal{S}_l[k]$.

3. Apply the robust MPC formulation for time-variant JMLS, presented in Section 7.2, to the linearized system $\mathcal{S}_l[k]$. To guarantee robustness, the error set $\mathbb{W}$ and the linearization errors are employed during the design of the quadratic constraints (offline) and the tightening of the state constraints (online).

4. Apply $u[k]$ to the JMS and go to step 1 at $k+1$.

This approach has been published by the author in [134] assuming that $x_r = \mathbf{0}$ and $u_{r,i} = \mathbf{0}$. Main parts of this chapter are based on this publication.

8.1. Design of a Robust MPC

In this section, the linearization procedure and the determination of the extended disturbance set are considered. Subsequently, a way to compute RCIS for the JMS is described. Finally, the formulation of the robust MPC is presented.

System Linearization

The linearization of the nonlinear JMS (8.1) in a linearization point $\xi_l[k] = \big(x_l[k], u_l[k], w_l[k]\big)$ is given by the first order Taylor polynomial:

$$
\begin{aligned}
f_i\big(\xi[k]\big) \;=\; & f_i\big(\xi_l[k]\big) + A_i[k]\big(x[k] - x_l[k]\big) + B_i[k]\big(u[k] - u_l[k]\big) \\
& + G_i[k]\big(w[k] - w_l[k]\big) + e_{l,i}[k],
\end{aligned}
\tag{8.3}
$$

where the system matrices are defined by:

$$
A_i = \left.\frac{\partial f_i}{\partial x}\right|_{\xi_l[k]}, \qquad B_i = \left.\frac{\partial f_i}{\partial u}\right|_{\xi_l[k]}, \qquad G_i = \left.\frac{\partial f_i}{\partial w}\right|_{\xi_l[k]}.
\tag{8.4}
$$

The linearization error is denoted by $e_{l,i}[k]$. Omitting the linearization error, the linear system dynamics (8.3) can be used to define a JMLS that approximates the behavior of the nonlinear JMS. The offset, caused by the linearization, is considered by an augmented disturbance $(w'[k])^\top := \begin{bmatrix} 1 & 1 & w^\top[k] \end{bmatrix}$ and $G_i'[k] := \begin{bmatrix} f_i(\xi_l[k]) & -\begin{bmatrix} A_i[k] & B_i[k] & G_i[k] \end{bmatrix}\xi_l[k] & G_i[k] \end{bmatrix}$:

$$
\mathcal{S}_l[k] := \begin{cases} x[1] &= A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[k] + G_{\theta_k}'[k]\,w'[k] \\ \mathcal{M} &= (\Theta, P[k], \mu[k]) \end{cases}.
\tag{8.5}
$$

The resulting JMLS $\mathcal{S}_l[k]$ has the same form as in Chapter 7. Hence, the nonlinear JMS is approximated at each sampling instant k with the JMLS $\mathcal{S}_l[k]$, so that the robust MPC for time-variant JMLS can be applied. For the linearization, the current

state, the inputs predicted in the previous time step, and the expected values of the disturbances are used:

$$\xi_l^\top[k] = \begin{bmatrix} x^\top[k] & u^\top[k\,|\,k-1] & \bar{w}^\top[k] \end{bmatrix}. \tag{8.6}$$

For $k = 0$, a reasonable initial value has to be chosen for $u_l[k]$. In general, this is possible based on the knowledge about the production process.

Remark 8.1. *If the states, inputs, or disturbances change significantly during the prediction horizon N, the simple linearization at $\xi_l[k]$, according to (8.6), may result in an inaccurate state prediction. This issue may be mitigated by linearizing the JMS around state, input, and disturbance trajectories for the whole prediction horizon. To this end, the trajectories $\boldsymbol{u}[k-1]$ and $\boldsymbol{x}[k-1]$, determined by the MPC at the previous time step, can be used.*

Prediction Error Set

In the RCIS and controller design, the error of the one-step prediction of the state $x[\![1]\!]$ has to be considered. This error results from the linearization error and the disturbance $w[k]$. The linearization error $e_{l,i}[k]$ can be over-approximated employing the *Lagrange remainder*. The *Lagrange remainder* for the m-th component of the linearization error vector $e_{l,i}[k]$ is given by (cf. [8]):

$$e_{l,i,m}[k] = \frac{1}{2} \left(\xi[k] - \xi_l[k]\right)^\top \left.\frac{\partial^2 f_{i,m}}{\partial^2 \xi}\right|_{\xi=\xi'_{i,m}[k]} \left(\xi[k] - \xi_l[k]\right). \tag{8.7}$$

According to the mean value theorem, there is a $\xi'_{i,m}[k] \in \left\{\xi_l[k] + \epsilon\left(\xi[\![j]\!] - \xi_l[k]\right) \mid \epsilon \in [0,1]\right\}$ for all $i \in \Theta$, $m \in \mathbb{N}_{n_x}$, and k such that Equation (8.7) holds. Due to the definition of the linearization point in (8.6), one has that $\xi_l[k] \in \Xi$. Since the set Ξ is bounded, a box-over-approximation $\lfloor e_{l,i} \rfloor$ of all possible linearization errors can be calculated by applying interval arithmetics to (8.7). To this end, $x[k] - x_l[k] = \boldsymbol{0}$ is used, since the current state is measured.

In addition to the linearization error, the prediction error caused by the difference between the value $w[\![0]\!]$ used for the state prediction and the actual value $w[k]$ has to be considered. This error can be over-approximated employing the linearized system:

$$e_{w,i}[k] = \left.\frac{\partial f_i}{\partial w}\right|_{\xi_l[k]} \left(w[k] - w[\![0]\!]\right). \tag{8.8}$$

Analogously to the linearization error, box over-approximations $\lfloor e_{w,i} \rfloor$ can be determined by interval arithmetics for all $i \in \Theta$.

The overall one-step prediction error is given by $e_i[k] = e_{l,i}[k] + e_{w,i}[k]$. Hence, the overall error sets can be approximated with the boxes $\lfloor e_i \rfloor = \lfloor e_{l,i} \rfloor + \lfloor e_{w,i} \rfloor$. For the RCIS design, ellipsoidal error sets $\mathcal{E}(E_i) \supseteq \lfloor e_i \rfloor$ are needed. To this end, ellipsoids with minimal volume are determined. The shape matrices E_i can be calculated by

solving common SDP formulations. This approach introduces a certain degree of conservatism to the RCIS design, but guarantees robustness with respect to the linearization errors and the disturbances.

Remark 8.2. *The computation of the linearization error with the Lagrange remainder is quite conservative and may result in disturbance sets that are too large for the RCIS design. To reduce the conservatism, the errors can be approximated by Monte Carlo simulations of the nonlinear JMS and the linearized system employing many different linearization points and disturbances. This approach may considerably reduce the conservatism of the design approach, but removes the theoretical robustness guarantee. To reach a sufficiently high degree of robustness, a large number of simulations should be used.*

Polytopic Linear Differential Inclusion and RCIS Design

As for JMLS, ellipsoidal RCIS and linear time-variant controllers $K_i[k]$ are determined for the nonlinear JMS. To control the system to the reference x_{r}, the nonlinear analogue of the delta system (7.5) is introduced:

$$\mathcal{S}_\delta := \begin{cases} x_\delta[k+1] = f_{\theta_k}\big(\xi_\delta[k] + \xi_{\mathrm{r},\theta_k}\big) - x_{\mathrm{r}} =: f_{\delta,\theta_k}\big(\xi_\delta[k]\big) \\ \qquad \mathcal{M} = \big(\Theta, P[k], \mu[0]\big) \end{cases} \tag{8.9}$$

with $\xi_\delta[k] := \xi[k] - \xi_{\mathrm{r},\theta_k}$. The admissible set for $\xi_\delta[k]$ is denoted by Ξ_δ. With this definition, one has that $f_{\delta,i}(\mathbf{0}) = \mathbf{0}$ for all $i \in \Theta$. Thus, the JMS converges to the reference if the delta system $\mathcal{S}_\delta$ converges to the origin.

From the mean value theorem (cf. [43]), it follows that parameters $\epsilon_m \in [0,1]$ exist for each component $m \in \mathbb{N}_{n_{\mathrm{x}}}$ of the state vector $x_\delta[k]$ such that

$$x_\delta[k+1] = f_{\delta,\theta_k}\big(\xi_\delta[k]\big) - f_{\delta,\theta_k}(\mathbf{0}) = \begin{bmatrix} \nabla f_{\delta,\theta_k,1}\big(\epsilon_1\,\xi_\delta[k]\big) \\ \vdots \\ \nabla f_{\delta,\theta_k,n_{\mathrm{x}}}\big(\epsilon_{n_{\mathrm{x}}}\,\xi_\delta[k]\big) \end{bmatrix} \big(\xi_\delta[k] - \mathbf{0}\big). \tag{8.10}$$

In Equation (8.10), ∇ defines the gradient of a function and $f_{\delta,i,m}(\cdot)$ denotes the m-th component of the vector-valued function $f_{\delta,i}(\cdot)$. In this formulation, the nonlinear dynamics at time k are described exactly by a linear system. By separating the gradient matrix columns with respect to the dimensions of the state, input, and disturbance vectors, a time-variant linear system of the following form results:

$$\begin{aligned} x_\delta[k+1] &= A_{\theta_k}\big(\xi_\delta[k]\big)\,x_\delta[k] + B_{\theta_k}\big(\xi_\delta[k]\big)\,u_\delta[k] + G_{\theta_k}\big(\xi_\delta[k]\big)\,w_\delta[k] \\ &= \tilde{A}_{\theta_k}[k]\,x_\delta[k] + \tilde{B}_{\theta_k}[k]\,u_\delta[k] + \tilde{G}_{\theta_k}[k]\,w_\delta[k]. \end{aligned} \tag{8.11}$$

The resulting time-variant JMLS represents the dynamics of the nonlinear JMS exactly. Note that the matrices $A_i[k]$, $B_i[k]$, and $G_i[k]$ resulting from the linearization are different from $\tilde{A}_i[k]$, $\tilde{B}_i[k]$, and $\tilde{G}_i[k]$ due to the constant terms resulting from the linearization and the linearization error in (8.3). In addition, the matrices $\tilde{A}_i[k]$,

$\tilde{B}_i[k]$, and $\tilde{G}_i[k]$ should not be confused with the prediction matrices $\tilde{A}_i[\![j]\!]$, $\tilde{B}_i[\![j,l]\!]$, and $\tilde{G}_i[\![j,l]\!]$ in Section 6.1. The values of $\tilde{A}_{\theta_k}[k]$, $\tilde{B}_{\theta_k}[k]$, and $\tilde{G}_{\theta_k}[k]$ are unknown in advance. To design the RCIS and the controllers, all possible values of these matrices are over-approximated by the matrix polytopes $\mathbb{A}_i$, $\mathbb{B}_i$, and $\mathbb{G}_i$ such that:

$$\begin{bmatrix} \tilde{A}_i[k] & \tilde{B}_i[k] & \tilde{G}_i[k] \end{bmatrix} \in \begin{bmatrix} \mathbb{A}_i & \mathbb{B}_i & \mathbb{G}_i \end{bmatrix} \supseteq \left\{ \begin{bmatrix} A_i(\xi_\delta) & B_i(\xi_\delta) & G_i(\xi_\delta) \end{bmatrix}, \ \xi_\delta \in \Xi_\delta \right\} \quad (8.12)$$

for all $i \in \Theta$. Since Ξ_δ is a known bounded set, the minimal and maximal values of the elements of $\tilde{A}_i(\cdot), \tilde{B}_i(\cdot)$, and $\tilde{G}_i(\cdot)$ can be determined by interval arithmetics or simulations. These intervals can be used to determine the vertices of the polytopes $\mathbb{A}_i$, $\mathbb{B}_i$, and $\mathbb{G}_i$. With this approach, *polytopic linear differential inclusions* of the nonlinear system dynamics are defined for each Markov state. A similar approach has been proposed for deterministic, continuous-time systems in [151].

Once the polytopes $\mathbb{A}_i$, $\mathbb{B}_i$, and $\mathbb{G}_i$ have been determined, the controllers $K_i[k]$, the RCIS, Lyapunov matrices $\mathscr{P}_i$, and the matrix L can be determined by solving the SDP (7.55). Note that the implementation of the controllers $K_i[k]$ is not possible, since $\mathfrak{a}_{i,l_A}[k]$ and $\mathfrak{b}_{i,l_B}[k]$ depend on the unknown inputs $u[k]$ and disturbances $w[k]$. However, for the formulation of the MPC, only the RCIS are needed. Only the theoretical existence of controllers is necessary to guarantee recursive feasibility.

MPC Formulation

Due to the online linearization, a new JMLS $\mathcal{S}_l[k]$ is determined at each time step k. This JMLS has the same structure as (7.47). Hence, the MPC formulation presented in Section 7.2 can be applied to the JMLS $\mathcal{S}_l[k]$. Only the state constraint tightening has to be adapted slightly since the linearization errors have to be considered. To this end, the overall disturbance set $\mathcal{E}(E_{\theta_k})$ is employed.

Theorem 8.1. *Let the matrices $L = \mathscr{Z}^{-1}$ and $\mathscr{P}_i = \mathscr{Q}_i^{-1}$ be a solution of (7.55). The matrices $\breve{\mathscr{Z}}_{i,m}$ define inner approximations of the Minkowski difference $\mathcal{E}_m \ominus \mathcal{E}_{w,\theta_k}$ with maximum volume. In addition, let $\boldsymbol{h}_u[k] = \begin{bmatrix} h_u^\top[k] & \cdots & h_u^\top[k+N-1] \end{bmatrix}^\top$ and $\boldsymbol{H}_u = I_N \otimes H_u$. Then, an MPC procedure that solves the following optimization problem at each time k using the linearized system $\mathcal{S}_l[k]$:*

$$\min_{\boldsymbol{u}[k]} \ \|\boldsymbol{u}[k]\|_{W[k]}^2 + q[k]\,\boldsymbol{u}[k] \quad (8.13a)$$

$$\text{s.t. } \boldsymbol{H}_u \boldsymbol{u}[k] \le \boldsymbol{h}_u[k], \quad (8.13b)$$

$$H_{x,l_x}\left(A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[\![0]\!] + G'_{\theta_k}[k]\,\bar{w}'[k] \right) \le h_{x,l_x}[k] - \left\| H_{x,l_x}^\top \right\|_{E_{\theta_k}}, \quad (8.13c)$$

$$\left\| A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[\![0]\!] + G'_{\theta_k}[k]\,\bar{w}'[k] - x_r \right\|_{\breve{\mathscr{Z}}_{\theta_k,m}^{-1}}^2 \le 1, \quad (8.13d)$$

$$\left\| A_{\theta_k}[k]\,x[k] + B_{\theta_k}[k]\,u[\![0]\!] + G'_{\theta_k}[k]\,\bar{w}'[k] - x_r \right\|_{\mathcal{T}_{\theta_k}(\mathscr{P},0)}^2 \le \|x[k] - x_r\|_{\mathscr{P}_{\theta_k}-L}^2, \quad (8.13e)$$

where $l_x \in \mathbb{N}_{n_{h,x}}$ and $m \in \Theta$ with $p_{\theta_k,m} > 0$, stabilizes $\mathcal{S}$ in the PMSS sense. Furthermore, the optimization problem is recursively feasible, and the constraints (7.1) are satisfied by the closed-loop system.

Proof. The overall disturbance set $\mathcal{E}\left(E_{\theta_k}\right)$ over-approximates the linearization error and the effects of the disturbances $w[k]$, i.e., it includes all possible errors of the one-step prediction. The proof follows directly from the proofs of Theorem 7.2 and 7.3, since $\mathcal{E}\left(E_{\theta_k}\right)$ is used for the RCIS design as well as the state and invariance constraint formulation in (8.13c) and (8.13d). PMSS is guaranteed by the constraint (8.13e). This follows from Lemma 2.10 and Corollary 2.2 on page 36, since the linearization error is bounded. □

This MPC can be used with both cost functions (CoE and EoC) if the matrices $W[k]$ and $q[k]$ are chosen as defined in Theorem 6.2 or 6.3. The optimization problem (8.13) is a QCQP with the same dimension as in (7.42). The design and implementation procedure is as shown in Fig. 7.4 for time-variant JMLS. The only additional computation needed, compared to the approaches presented in Chapter 7, is the system linearization. It is now the first step of the preprocessing in each cycle. Thus, the resulting computation time is similar to that of the robust MPC formulation for JMLS, i.e., the computation time is relatively low.

Since the JMS is linearized at each time step k, the linearization error is relatively small for most applications. The remaining conservatism is mainly caused by the RCIS design based on the polytopic system approximation and the ellipsoidal disturbance over-approximation. However, due to the consideration of time-variant controllers, this problem is mitigated significantly.

The control performance may be improved by an iterative linearization scheme, i.e., it is iterated between the system linearization and solving the QCQP (8.13) several times before the input $u[k] = u[\![0]\!]$ is applied to the system. Due to the iteration, the linearization error may be reduced but the computation time is increased significantly. This approach is not considered, since the performance improvement is marginal for the examples considered (including the hot stamping process models).

8.2. Simulation Results

This section illustrates the MPC design procedure and demonstrates its properties. To this end, the following JMS with $n_x = 2, n_u = n_w = 1$, and $n_\theta = 2$ is considered:

$$f_1\big(\xi[k]\big) = \begin{bmatrix} 0.2\,x_1^3[k] + 0.2\,x_2[k] + 1.4\,\mathrm{atan}\,\frac{u[k]}{8} + 0.2\,w[k] \\ 0.5\,x_1[k] + 0.3\,x_2^2[k] + 1.8\,\mathrm{atan}\,\frac{u[k]}{8} + 0.3\,\mathrm{atan}\,w[k] \end{bmatrix}, \tag{8.14a}$$

$$f_2\big(\xi[k]\big) = \begin{bmatrix} 1.05\,x_1[k]\,\mathrm{e}^{-0.05\,x_2[k]} - 0.3\,x_2[k] + 2.2\,\mathrm{atan}\,\frac{u[k]}{8} - 0.2\,w[k] \\ 0.5\,x_1^4[k] + 0.5\,x_2[k] + \mathrm{atan}\,\frac{u[k]}{8} + 0.2\,\sin w[k] \end{bmatrix}, \tag{8.14b}$$

$$\mathcal{M} = \left(\{1, 2\}, \begin{bmatrix} 0.9 & 0.1 \\ 0.55 & 0.45 \end{bmatrix}, \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \right), \qquad x[0] = \begin{bmatrix} -1 \\ 1 \end{bmatrix}. \tag{8.14c}$$

The JMS is based on a JMS system used in [82]. The constraints and admissible sets $\mathbb{X}$ and $\mathbb{U}$ are defined by the box constraints $[-1.2 \ -2]^\top \leq x[k] \leq [1.2\ 2]^\top$ and

$|u[k]| \leq 4$. The disturbances are taken from a truncated normal distribution with $w[k] \sim \mathcal{N}(0, 0.1)$ and $|w[k]| \leq 0.3$. The state and input reference is the origin.

RCIS Design: The system matrix polytopes $\mathbb{A}_i$ are determined by factorization of $f_i(\cdot)$ and inserting the bounds of the state set $\mathbb{X}$:

$$\mathbb{A}_1 = \begin{bmatrix} [0, 0.29] & 0.2 \\ 0.5 & [-0.6, 0.6] \end{bmatrix}, \qquad \mathbb{A}_2 = \begin{bmatrix} [0.95, 1.16] & -0.3 \\ [-0.86, 0.86] & 0.5 \end{bmatrix}. \tag{8.15}$$

For the input matrices, the worst case is considered. For steering the system to the origin, these are the matrices that result from (8.4) and have the lowest absolute values of the elements, since the largest absolute input values are needed (which are constrained) to impose the same effect on the states. The worst case matrices are:

$$\mathbb{B}_1 = \begin{bmatrix} 0.14 \\ 0.18 \end{bmatrix}, \qquad\qquad \mathbb{B}_2 = \begin{bmatrix} 0.22 \\ 0.10 \end{bmatrix}. \tag{8.16}$$

The polytopes $\mathbb{G}_i$ are not needed for the RCIS design, since the disturbance terms are included in the overall disturbance sets. The box approximations of the disturbance set are determined employing interval arithmetics considering a maximum input prediction error $|u[k|k-1] - u[k]| \leq 1.5$:

$$\lfloor e_1 \rfloor = \begin{bmatrix} [-0.11, 0.11] \\ [-0.15, 0.15] \end{bmatrix}, \qquad \lfloor e_2 \rfloor = \begin{bmatrix} [-0.14, 0.14] \\ [-0.10, 0.10] \end{bmatrix}. \tag{8.17}$$

For the resulting polytopic differential inclusion and error sets $\mathcal{E}(E_{\theta_k})$, the RCIS are determined by solving (7.55) with PenBMI.

MPC simulation: The JMS (8.14) is simulated in closed-loop with the MPC proposed in Theorem 8.1. To illustrate the effects of the quadratic constraints, the system is also simulated with an MPC that solves (8.13) without the quadratic constraints (8.13d) and (8.13e). Both setups are simulated for 10,000 Markov state trajectories employing the EoC cost function with a prediction horizon of $N = 10$ and costs defined by $Q_1 = Q_2 = I$ as well as $R_1 = R_2 = 2$. The mean values of the states and inputs as well as the corresponding envelopes of all simulations are shown in Fig. 8.1. The results demonstrate that the MPC solving the QCQP (8.13) stabilizes the JMS robustly and satisfies the constraints. If the quadratic constraints are omitted, the closed-loop system is unstable for some simulation runs (see top plots in Fig. 8.1). The corresponding QP has been infeasible 723 times. Hence, the quadratic invariance and stability constraints are essential for a robust control.

The average computation time for all steps performed online (linearization, calculation of $W[k]$ and $q[k]$, and solving the optimization problem) is about 7.3 ms for the robust MPC, and 6.2 ms if the quadratic constraints are omitted. As for JMLS,
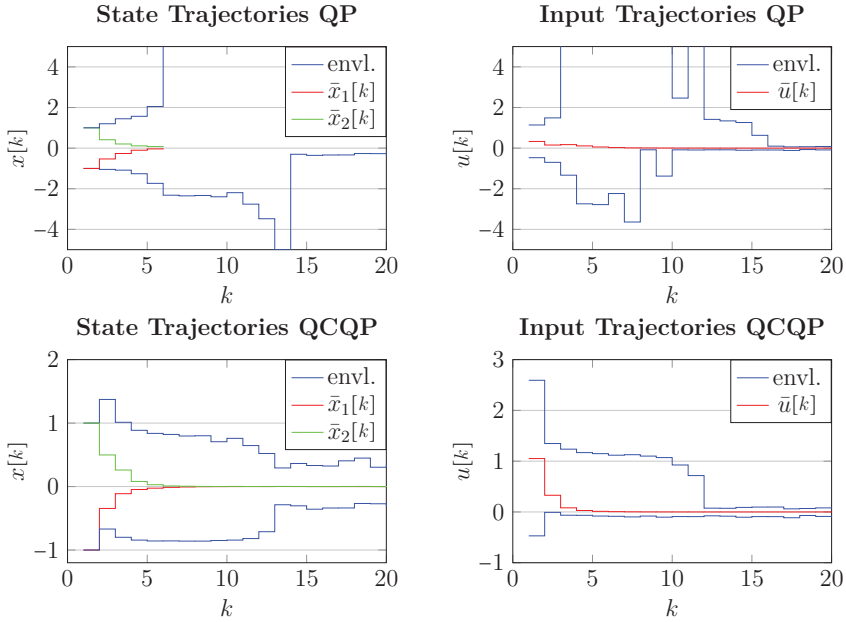
Figure 8.1.: Mean values and envelopes of states (left) and inputs (right) for 10,000 simulations of the JMS controlled by the simple QP-based MPC (top) and the robust QCQP-based MPC (bottom). The upper bound and mean values of the state trajectories for the QP problem are only shown until the sixth time step, since the corresponding values grow over all bounds due to instability.

the resulting computation times increase if the quadratic constraints are considered. However, the increase in computation time is relatively low and the absolute computation times are still very low for an MPC approach for a nonlinear JMS.

All in all, the simulations demonstrate that the robust MPC for nonlinear JMS satisfies the requirements stated in Section 4.4. The MPC stabilizes the nonlinear JMS robustly in the PMSS sense. State and input constraints are satisfied and the optimization problem is recursively feasible. If necessary, e.g., for economic MPC setups, the quadratic stability constraint can be omitted and the JMS may converge to any state within the union of the RCIS and the set defined by the input and state constraints. In addition, the resulting computation times are significantly below one second. This renders the proposed approach applicable to systems with relatively large state and input dimensions.

9. State Estimation

For some applications, not all continuous states $x[k]$ are measurable. In this cases, state estimation has to be employed to apply the MPC approaches proposed in the previous chapters. This chapter demonstrates how the states can be estimated, provided that the JMS is observable. The aim of this chapter is not to derive new filtering concepts, but to provide a brief overview of existing approaches and to illustrate how some of these approaches can be applied to the considered JMS.

9.1. State of the Art in Filtering for JMS

The topic of state estimation for JMS has been investigated to a considerable extend. The existing approaches can be grouped into approaches for linear and nonlinear JMS. In addition, the approaches can be characterized by whether the Markov state is accessible or not.

1. **Jump Markov Linear Systems**
   a) **The Markov state is known:** If the Markov state is known, the underlying system can be regarded as a switched or a general time-variant linear system. An overview of filters for linear switched systems is provided in [61]. The optimal filter in case of Gaussian noise processes is the well known time-variant discrete-time Kalman filter [16, 39]. The result is a switched Kalman filter [16]. To reduce the computational effort, switched linear filters can be employed. In [46], the LMI-based design of mean square stable switched linear filters is considered.

   Robust filters guarantee mean square stability and boundedness of the estimation error. In this context, switched $\mathcal{H}_\infty$-filters are considered that are robust to uncertainties in the system matrices [48] or the transition probabilities [153, 154]. For a more detailed overview, the reader is referred to these references.

   b) **The Markov state is unknown:** The optimal non-linear filtering solution would exhibit an exponential increase in computational complexity with time [39]. Hence, sub-optimal filters are employed. To this end, linear filters minimizing the mean square error are proposed [39]. In addition, robust filters can be designed for JMLS with unknown Markov state [39, 48]. In particle filter approaches for JMLS, a set of random samples is used to approximate the posterior probability density function of both the Markov states and the continuous states [41].

The concept of receding horizon estimation (RHE) establishes an approach similar to MPC: The trajectories of the continuous state, the Markov state, and the disturbance for the previous N time steps are estimated by solving an optimization problem [3].

2. **Nonlinear Jump Markov Systems**

   a) **The Markov State is known:** In this case, the system can be interpreted as a time-variant or switched nonlinear system. Hence, common filtering approaches for these system classes, such as *extended Kalman filters* (EKF), iterated EKFs, unscented Kalman filters, or particle filters can be employed. For details, the reader is referred to the books [5, 122].

   b) **The Markov State is unknown:** Most approaches aim to approximate the posterior probability density functions of the states. To this end, particle filters are commonly used (cf. [6, 80, 157]). In addition, the RHE concept can also be applied to switched nonlinear systems with unknown switching mode (cf. [51]).

While many approaches for MPC of JMS and estimation for JMS exist, very few approaches are considered for the combination of both. In [143], a robust $\mathcal{H}_\infty$-filter is combined with a state feedback pre-stabilization and a one-step MPC. However, the disturbances $w[k]$ are assumed to be measurable. In [31], a Kalman filter is combined with a pre-stabilization and a one-step MPC considering chance constraints. To the best of the author's knowledge, there are no approaches combining state estimation and MPC for nonlinear JMS.

9.2. State Estimation Problem and Approaches

To formulate the estimation problems, the nonlinear JMS defined in (5.2) is extended by a measurement output $z[k] \in \mathbb{R}^{n_z}$ with measurement noise $\nu_z[k] \in \mathbb{R}^{n_z}$:

$$\mathcal{S} := \begin{cases} x[k+1] &= f_{\theta_k}\big(x[k], u[k], w[k]\big) \\ z[k] &= h_{\theta_k}\big(x[k], u[k], w[k]\big) + \nu_z[k] \\ \mathcal{M} &= \big(\Theta, P[k], \mu[k]\big) \end{cases} . \tag{9.1}$$

The corresponding JMLS is given by:

$$\mathcal{S}_l := \begin{cases} x[k+1] &= A_{\theta_k}[k]\, x[k] + B_{\theta_k}[k]\, u[k] + G_{\theta_k}[k]\, w[k] \\ z[k] &= C_{\theta_k}[k]\, x[k] + D_{\theta_k}[k]\, u[k] + F_{\theta_k}[k]\, w[k] + \nu_z[k] \\ \mathcal{M} &= \big(\Theta, P[k], \mu[k]\big) \end{cases} . \tag{9.2}$$

For the sake of a brief notation, the output $y[k]$ is omitted. The resulting system is a general time-variant nonlinear or linear system, since the current Markov state θ_k is measurable. For notational convenience, the following abuse of notation is used:

$$A_k := A_{\theta_k}[k], \quad B_k := B_{\theta_k}[k], \quad G_k := G_{\theta_k}[k], \quad C_k := C_{\theta_k}[k], \quad D_k := D_{\theta_k}[k], \quad F_k := F_{\theta_k}[k]. \tag{9.3}$$

The goal is to derive an estimate of the states, denoted by $\check{x}[k]$, based on the measurements $z[k]$. To this end, two estimation approaches, using the fact that the Markov state is known, are considered with the following motivations:

1. **Kalman filter:** The Kalman filter is the optimal filter for a JMLS (9.2) [39], if the noise processes are normally distributed. It is easy to implement and requires low computational effort.

2. **Receding Horizon Estimator (RHE):** RHE is the estimation analogue to MPC. It is easy to implement and capable of handling nonlinearities, general noise distributions, and known bounds of the disturbances.

In addition, both concepts are used for linear and nonlinear systems. The corresponding formulations are presented in the following sections.

9.2.1. Kalman Filtering

This section presents a Kalman filter formulation for the JMLS (9.2) and an extended Kalman filter for the JMS (9.1). The following derivations do not aim at proposing a new filtering concept, but to adapt the concept of Kalman filtering [5, 122] to the employed system dynamics. For Kalman filters, the following assumption on the noise processes is commonly made (see, e.g., [122]):

Assumption 9.1. *The noise processes are governed by a zero-mean normal distribution with known covariance, i.e., $\nu_z[k] \sim \mathcal{N}(0, \Sigma_\nu)$ and $w[k] \sim \mathcal{N}(0, \Sigma_{\mathrm{w}})$. Both are independent of the state $x[k]$, the input $u[k]$, the output $y[k]$, and the Markov state θ_k, but may be correlated with the covariance $\mathrm{E}(w[k]\nu_z^\top[k]) = \Sigma_{\mathrm{w}\nu}$.*

The zero mean assumption for $w[k]$ contradicts Assumption 2.2. Since the mean $\bar{w}[k]$ is known, the disturbance $w[j]$ can be split into its mean $\bar{w}[k]$ and a zero mean part $\tilde{w}[j]$. The mean $\bar{w}[k]$ can be considered as a second known input. An extended JMS, for which Assumption 9.1 holds, results (with a slight abuse of notation) by redefining the corresponding functions and quantities in (9.1) and (9.2) such that:

$$u[k] := \begin{bmatrix} u^\top[k] & \bar{w}^\top[k] \end{bmatrix}^\top, \quad w[k] := \tilde{w}[k], \quad B_k := \begin{bmatrix} B_k & G_k \end{bmatrix}, \quad D_k := \begin{bmatrix} D_k & F_k \end{bmatrix}. \quad (9.4)$$

Let the estimation error and its covariance be defined as follows:

$$\check{e}[k \,|\, k] := x[k] - \check{x}[k \,|\, k], \qquad \Sigma_{k \,|\, k} := \mathrm{Cov}(\check{e}[k \,|\, k]), \qquad (9.5)$$

$$\check{e}[k \,|\, k-1] := x[k] - \check{x}[k \,|\, k-1], \qquad \Sigma_{k \,|\, k-1} := \mathrm{Cov}(\check{e}[k \,|\, k-1]). \qquad (9.6)$$

For this setup, the filter formulations are presented below.

Linear Case

Commonly, a discrete-time Kalman filter is divided into two steps [5, 122]:

1. **Prediction:** The estimate for the current time step $\check{x}[k\,|\,k-1]$ is a prediction based on the state estimate $\check{x}[k-1\,|\,k-1]$ of the previous time step:

$$\check{x}[k\,|\,k-1] = A_{k-1}\check{x}[k-1\,|\,k-1] + B_{k-1}u[k-1]. \tag{9.7}$$

2. **Correction:** The predicted state estimate $\check{x}[k\,|\,k-1]$ is corrected employing the current measurement $z[k]$:

$$\check{x}[k\,|\,k] = \check{x}[k\,|\,k-1] + L_k\left(z[k] - C_k\,\check{x}[k\,|\,k-1] - D_k\,u[k]\right). \tag{9.8}$$

The Kalman gain L_k is selected such that the variance of the estimation error $\mathrm{tr}(\Sigma_{k\,|\,k})$ is minimized. The values of the Kalman gain matrices L_k can be determined as shown in the following theorem:

Theorem 9.1. *Given an initialization $\Sigma_{0|-1}$ according to* (9.5)*, the gain L_k and the estimation error covariances $\Sigma_{k\,|\,k-1}$ and $\Sigma_{k\,|\,k}$ can be calculated recursively:*

$$
\begin{aligned}
L_k &= \Sigma_{k\,|\,k-1}\,C_k^\top\left(C_k\Sigma_{k\,|\,k-1}\,C_k^\top + F_k\Sigma_{\mathrm{w}\nu} + \Sigma_{\mathrm{w}\nu}^\top F_k^\top + F_k\Sigma_{\mathrm{w}}F_k^\top + \Sigma_\nu\right)^{-1}, &(9.9)\\
\Sigma_{k\,|\,k} &= \Sigma_{k\,|\,k-1} - L_kC_k\Sigma_{k\,|\,k-1}, &(9.10)\\
\Sigma_{k+1|k} &= A_k\Sigma_{k\,|\,k}A_k^\top - A_kL_k\left(F_k\Sigma_{\mathrm{w}} + \Sigma_{\mathrm{w}\nu}^\top\right)G_k^\top - G_k\left(\Sigma_{\mathrm{w}}^\top F_k^\top + \Sigma_{\mathrm{w}\nu}\right)L_k^\top A_k^\top + G_k\Sigma_{\mathrm{w}}G_k^\top. &(9.11)
\end{aligned}
$$

Proof. The proof is based on the procedures presented in [122]. First, the correction step is considered. Inserting the correction equation (9.8) into the definition of the estimation error in (9.5) and substituting the measurement signal $z[k]$ by the corresponding system equation (9.2) leads to:

$$
\begin{aligned}
\check{e}[k\,|\,k] = x[k] - \check{x}[k\,|\,k] &= x[k] - \check{x}[k\,|\,k-1] - L_k\left(C_k\,x[k] + D_k\,u[k] + F_k\,w[k] + \nu_{\mathrm{z}}[k]\right.\\
&\qquad\qquad\qquad\qquad \left. - C_k\,\check{x}[k\,|\,k-1] - D_k\,u[k]\right)\\
&= \left(I - L_kC_k\right)\check{e}[k\,|\,k-1] - L_kF_kw[k] - L_k\nu_{\mathrm{z}}[k]. \tag{9.12}
\end{aligned}
$$

The covariances $\mathrm{Cov}\big(\check{e}[k\,|\,k-1], w[k]\big)$ and $\mathrm{Cov}\big(\check{e}[k\,|\,k-1], \nu_{\mathrm{z}}[k]\big)$ are zero, since $\check{x}[k]$ and $x[k]$ are independent of $w[k]$ and $\nu_{\mathrm{z}}[k]$ (cf. (9.2) and (9.7)). Employing this and the unbiased property of the Kalman filter approach [122], the covariance $\Sigma_{k\,|\,k}$ can be calculated as follows:

$$
\begin{aligned}
\Sigma_{k|k} &= \mathrm{E}\big(\check{e}[k\,|\,k]\,\check{e}^\top[k\,|\,k]\big) = \left(I - L_kC_k\right)\Sigma_{k\,|\,k-1}\left(I - L_kC_k\right)^\top + L_kF_k\Sigma_{\mathrm{w}}F_k^\top L_k^\top\\
&\quad + L_kF_k\Sigma_{\mathrm{w}\nu}L_k^\top + L_k\Sigma_{\mathrm{w}\nu}^\top F_k^\top L_k^\top + L_k\Sigma_\nu L_k^\top\\
&= \Sigma_{k|k-1} + L_k\left(C_k\Sigma_{k|k-1}C_k^\top + F_k\Sigma_{\mathrm{w}\nu} + \Sigma_{\mathrm{w}\nu}^\top F_k^\top + F_k\Sigma_{\mathrm{w}}F_k^\top + \Sigma_\nu\right)L_k^\top\\
&\quad - L_kC_k\Sigma_{k\,|\,k-1} - \Sigma_{k\,|\,k-1}C_k^\top L_k^\top. \tag{9.13}
\end{aligned}
$$

The Kalman gain L_k has to be chosen such that $\text{tr}(\Sigma_{k|k})$ is minimized. Hence, the following condition has to hold for L_k:

$$\mathbf{0} \overset{!}{=} \frac{\partial \text{tr}(\Sigma_{k|k})}{\partial L_k} = 2L_k \left(C_k \Sigma_{k|k-1} C_k^\top + F_k \Sigma_{w\nu} + \Sigma_{w\nu}^\top F_k^\top + F_k \Sigma_w F_k^\top + \Sigma_\nu \right) - 2\Sigma_{k|k-1} C_k^\top. \quad (9.14)$$

Solving this condition for L_k results in Equation (9.9). Substituting L_k on the left hand side of the quadratic term in (9.13) with (9.9) results in equation (9.10) for the calculation of the error covariance $\Sigma_{k|k}$.

Employing the prediction equation (9.7), the prediction error is given by:

$$\check{e}[1] = x[k+1] - \check{x}[1] = A_k\, x[k] + B_k\, u[k] + G_k w[k] - \left(A_k \check{x}[k|k] + B_k u[k] \right) \quad (9.15)$$
$$= A_k \check{e}[k|k] + G_k w[k]. \quad (9.16)$$

The covariance of the prediction error $\Sigma_{k+1|k}$ can be calculated as follows:

$$\Sigma_{k+1|k} = \text{E}\left(\left(A_k \check{e}[k|k] + G_k w[k] \right) \left(A_k \check{e}[k|k] + G_k w[k] \right)^\top \right)$$
$$= A_k \Sigma_{k|k} A_k^\top + \text{E}\left(A_k\, \check{e}[k|k]\, w^\top[k] G_k^\top + G_k\, w[k]\, \check{e}^\top[k|k] A_k^\top \right) + G_k \Sigma_w G_k^\top. \quad (9.17)$$

The covariance $\text{Cov}\big(\check{e}[k|k], w[k]\big)$ can be calculated employing (9.12):

$$\text{E}\left(\left[\left[(I - L_k C_k)\, \check{e}[k|k-1] - L_k F_k w[k] - L_k \nu_z[k] \right] w^\top[k] \right] \right) = -L_k F_k \Sigma_w - L_k \Sigma_{w\nu}^\top. \quad (9.18)$$

Equation (9.18) holds since $\check{x}[k]$ and $x[k]$ are independent of $w[k]$ and $\nu_z[k]$, i.e., the covariances $\text{Cov}(\check{e}[k|k-1], w[k])$ and $\text{Cov}(\check{e}[k|k-1], \nu_z[k])$ are zero. Inserting (9.18) in (9.17) results in (9.11) for the calculation of the error covariance $\Sigma_{k+1|k}$. $\qquad \square$

This theorem provides a very general Kalman filter formulation that can be applied to the JMLS considered in this thesis.

Nonlinear Case

The common extension of the Kalman filter to nonlinear systems is the *extended Kalman filter* (EKF). The basic idea is to linearize the system at each sampling incident and apply the standard Kalman filter to the linearized system. Based on the Kalman filter formulation in Theorem 9.1, the EKF algorithm can be stated as in Algorithm 9.1 on page 132 (cf. [5, 122]). Due to the involved linearizations, the algorithm is not optimal nor is stability of the estimation error guaranteed. However, many applications demonstrate a good performance of the EKF – particularly for processes that are slow or do not exhibit significant nonlinearities [122]. The estimation error may be reduced by employing an iterated EKF, an unscented Kalman filter, or higher order extended Kalman filter approaches [122].

9.2.2. Receding Horizon Estimation

The Kalman filtering concept is only optimal for normally distributed disturbances. Receding horizon estimation, which is the analogue to MPC in the estimation

---

Algorithm 9.1 Extended Kalman Filter.

1: **given:** $\check{x}[1\,|\,0], \Sigma_{1\,|\,0}$
2: **for** $k \in \mathbb{N}$ **do**
3:      define: $C_k := \left. \frac{\partial h_{\theta_k}(x, u[k], 0)}{\partial x} \right|_{x = \check{x}[k\,|\,k-1]}, \qquad F_k := \left. \frac{\partial h_{\theta_k}(\check{x}[k\,|\,k-1], u[k], w)}{\partial w} \right|_{w=0}$
4:      compute: $L_k = \Sigma_{k\,|\,k-1} C_k^\top \left(C_k \Sigma_{k\,|\,k-1} C_k^\top + F_k \Sigma_{\mathrm{w}\nu} + \Sigma_{\mathrm{w}\nu}^\top F_k^\top + F_k \Sigma_{\mathrm{w}} F_k^\top + \Sigma_\nu \right)^{-1}$
5:      **correction**: $\check{x}[k\,|\,k] = \check{x}[k\,|\,k-1] + L_k \left(z[k] - h_{\theta_k}(\check{x}[k\,|\,k-1], u[k], 0) \right)$
6:      **prediction**: $\check{x}[1] = f_{\theta_k}(\check{x}[k\,|\,k], u[k], 0)$
7:      compute: $\Sigma_{k\,|\,k} = \Sigma_{k\,|\,k-1} - L_k C_k \Sigma_{k\,|\,k-1}$
8:      define: $A_k := \left. \frac{\partial f_{\theta_k}(x, u[k], 0)}{\partial x} \right|_{x = \check{x}[k\,|\,k]}, \qquad G_k := \left. \frac{\partial f_{\theta_k}(\check{x}[k\,|\,k], u[k], w)}{\partial w} \right|_{w=0}$
9:      compute: $\Sigma_{k+1|k}$ according to (9.11)
10: **end for**

---

domain, provides an estimation approach that is capable of considering general disturbance distributions and known bounds of the disturbances. The idea is to use the $N_{\mathrm{e}} + 1$ most recent measurements $\boldsymbol{z}[k] = \begin{bmatrix} z[k - N_{\mathrm{e}}] & \dots & z[k] \end{bmatrix}$ to estimate the initial state $\check{x}[-N_{\mathrm{e}}]$ and the disturbance sequence $\check{\boldsymbol{w}}[k] = \begin{bmatrix} \check{w}[k - N_{\mathrm{e}}] & \dots & \check{w}[k] \end{bmatrix}$ for $k \geq N_{\mathrm{e}}$ in a receding horizon fashion by solving optimization problems online. Employing the estimated disturbances $\check{\boldsymbol{w}}[k]$, the estimate of the current state $\check{x}[0]$ can be calculated employing the system dynamics (see [3, 51] and the references therein).

In order to illustrate the approach briefly, this section presents the optimization problem to be solved online for the linear and the nonlinear case. Compared to the Kalman filter setup, less restrictive assumptions are made for $w[k]$ and $\nu_{\mathrm{z}}[k]$:

Assumption 9.2. *The disturbance $w[k]$ and the measurement noise $\nu_{\mathrm{z}}[k]$ are bounded by the compact sets $\mathbb{W}$ and $\mathbb{V}$, respectively. In addition, the expected value of the disturbance $w[k]$ is known.*

Typically, approximations of these quantities are known from process recordings and from data sheets of the measurement equipment.

In [3], a RHE for switched linear systems with unknown switching mode is proposed. This approach can be adopted in a simplified version for the state estimation of the JMLS (9.2). For $k \geq N_{\mathrm{e}}$, the corresponding optimization problem is given by:

$$\min_{\check{x}[-N_{\mathrm{e}}], \check{\boldsymbol{w}}[k]} \|\check{x}[-N_{\mathrm{e}}] - \check{x}[k - N_{\mathrm{e}}|k-1]\|_Q^2 + \sum_{j=-N_{\mathrm{e}}}^{0} \left(\|\check{w}[\![j]\!] - \bar{w}[k+j]\|_R^2 + \|\check{z}[\![j]\!] - z[k+j]\|_S^2 \right) \quad (9.19\mathrm{a})$$

$$\text{s. t. } \check{z}[\![j]\!] = C_{k+j}\,\check{x}[\![j]\!] + D_{k+j}\,u[k+j] + F_{k+j}\,\check{w}[\![j]\!] \qquad \forall j \in \mathbb{N}_{-N_{\mathrm{e}}:0}, \qquad (9.19\mathrm{b})$$

$$\check{x}[\![j+1]\!] = A_{k+j}\,\check{x}[\![j]\!] + B_{k+j}\,u[k+j] + G_{k+j}\,\check{w}[\![j]\!] \qquad \forall j \in \mathbb{N}_{-N_{\mathrm{e}}:-1}, \qquad (9.19\mathrm{c})$$

$$\check{w}[\![j]\!] \in \mathbb{W} \qquad \forall j \in \mathbb{N}_{-N_{\mathrm{e}}:0}, \qquad (9.19\mathrm{d})$$

$$z[k+j] - \check{z}[\![j]\!] \in \mathbb{V} \qquad \forall j \in \mathbb{N}_{-N_{\mathrm{e}}:0}, \qquad (9.19\mathrm{e})$$

with positive definite matrices Q, R, and S. In the optimization problem (9.19), $\check{x}[-j]$ denotes the states estimated for j steps prior to the current time k. The

estimate $\check{x}[k-N_e|k-1]$ denotes a state estimated one time step before (at $k-1$). The first cost term penalizes the distance from the previous state estimate, the second one penalizes the deviation of the estimated disturbance from its expected value, and the last term penalizes the distance between the estimated measurement output and the actual measurements. By recursively inserting constraints (9.19b) and (9.19c), the optimization problem can be formulated in terms of $\check{x}[-N_e]$ and $\check{\boldsymbol{w}}[k]$. If the sets $\mathbb{W}$ and $\mathbb{V}$ are polytopes, the optimization problem (9.19) can be stated as a QP [3]. Once the measurement $z[k]$ becomes available, the optimization problem (9.19) can be solved in each time step to estimate the state of the JMLS.

The RHE formulation (9.19) can be extended to nonlinear JMS by replacing the linear system equations with their nonlinear counterparts:

$$\min_{\check{x}[-N_e],\check{\boldsymbol{w}}[k]} \|\check{x}[-N_e] - \check{x}[k-N_e|k-1]\|_Q^2 + \sum_{j=-N_e}^{0} \left(\|\check{w}[j] - \bar{w}[k+j]\|_R^2 + \|\check{z}[j] - z[k+j]\|_S^2 \right) \quad (9.20a)$$

$$\text{s.\,t.\ } \check{z}[j] = h_{\theta_{k+j}}(\check{x}[j], u[k+j], \check{w}[j]) \qquad\qquad \forall j \in \mathbb{N}_{-N_e:0}, \qquad (9.20b)$$

$$\check{x}[j+1] = f_{\theta_{k+j}}(\check{x}[j], u[k+j], \check{w}[j]) \qquad\quad \forall j \in \mathbb{N}_{-N_e:-1}, \qquad (9.20c)$$

$$\check{w}[j] \in \mathbb{W} \qquad\qquad\qquad\qquad\qquad\qquad\quad \forall j \in \mathbb{N}_{-N_e:0}, \qquad (9.20d)$$

$$z[k+j] - \check{z}[j] \in \mathbb{V} \qquad\qquad\qquad\qquad\quad\ \forall j \in \mathbb{N}_{-N_e:0}. \qquad (9.20e)$$

This optimization problem is a general nonlinear optimization problem that can be computationally very expensive and non-convex. To reduce the computational effort, the noise free case can be considered resulting in an optimization problem where $\check{x}[-N_e]$ is the only optimization variable [51]. Due to the direct consideration of the nonlinearities, the estimation quality can be increased compared to linearization based methods, like the EKF.

All in all, the advantage of the RHE compared to the Kalman filter is that no assumptions on the disturbance distributions are necessary and that known bounds on the disturbances and measurement noise can be considered. These advantages come at the expense of higher computational effort for solving the optimization problem. Hence, RHE is not applicable to systems with a very short sampling time.

9.3. State Estimation and MPC

In general, all MPC approaches proposed in the previous chapters can be used in combination with state estimation by replacing the state $x[k]$ with its estimate $\check{x}[k]$. The combinations of both Kalman filter and RHE with both the non-robust CoE and EoC approaches (according to Theorem 6.2 and 6.3) have been tested with a linearization of a simplified model[1] of a hot stamping tryout pressing tool in [116]. The simulations show that the average estimation errors are low for the Kalman

---

[1]In [116], the estimation combined with MPC has been investigated for a model of a test tool with just one tool temperature and one blank temperature.

filter and RHE (below 3 °C for both tool and blank temperature). In addition, a good closed-loop performance results for both estimation concepts in combination with the EoC approach (average control errors of the blank and tool temperature below 5 °C). This investigation demonstrates that the MPC approaches proposed in Chapter 6 can be combined with state estimation in a straightforward manner.

To use state estimation for the robust MPC approaches presented in Chapter 8, the estimation errors have to be considered. To this end, a compact set $\check{\mathbb{E}}$ containing all possible estimation errors $\check{e}[k]$ has to be determined. For RHE, bounds for the estimation error have been proposed in [4, 51]. However, these bounds are quite conservative. For the Kalman filter, the covariance matrices $\Sigma_{k|k}$ can be used to determine significance ellipsoids that approximate $\check{\mathbb{E}}$. However, the most promising way to approximate the set $\check{\mathbb{E}}$ seems to be the determination of approximations of the estimation error by Monte Carlo simulations. Once this approximation has been determined, the resulting one-step prediction error can be computed. Considering a JMLS (which is the result of the online linearization), the one-step prediction can be formulated as follows:

$$x[k+1] = A_{\theta_k} x[k] + B_{\theta_k} u[k] + G'_{\theta_k} w'[k] = A_{\theta_k}\big(\check{x}[k] + \check{e}[k]\big) + B_{\theta_k} u[k] + G'_{\theta_k} w'[k]. \quad (9.21)$$

Thus, the one-step prediction error caused by the estimation error is given by:

$$e_{\mathrm{e},i}[k] = \left.\frac{\partial f_i}{\partial x}\right|_{\xi_\mathrm{l}[k]} \check{e}[k]. \quad (9.22)$$

Analogously to the linearization error, box over-approximations $\lfloor e_{\mathrm{e},i}\rfloor$ can be determined by interval arithmetics employing the sets Ξ and $\check{\mathbb{E}}$. The overall one-step prediction error is determined by the sum of the linearization error, the disturbance-related error, and the estimation error. Thus, ellipsoidal error sets $\mathcal{E}(E_i) \supseteq \lfloor e_{\mathrm{l},i}\rfloor + \lfloor e_{\mathrm{w},i}\rfloor + \lfloor e_{\mathrm{e},i}\rfloor$ are obtained. Once these sets have been calculated, the RCIS design and the formulation of the MPC can be applied as presented in Chapter 8. If the SDP can be solved considering the estimation errors contained in $\check{\mathbb{E}}$, the resulting MPC formulation is robust against the disturbances and estimation errors.

9.4. Conclusions

In this chapter, state estimation for JMS and its combination with the MPC approaches proposed in this part are discussed. Since the Markov state is assumed to be measurable, the resulting system can be regarded as a common linear or nonlinear time-variant (switched) system. Numerous estimation approaches exist for these system classes and can be adapted to the considered setting. In this chapter, the Kalman filter, the EKF, and RHE concepts have been adapted to the definitions of JMS and JMLS used throughout this thesis. For JMLS, a time-variant Kalman filter approach constitutes the optimal filter, if the disturbances are normally distributed.

The RHE approaches consider general bounded disturbances and nonlinearties in a straightforward manner, at the expense of a higher computational effort.

The MPC approaches can be implemented by replacing $x[k]$ by its estimate $\check{x}[k]$. The results in [116] show that the combinations of the simple EoC MPC presented in Chapter 6 with the Kalman filter and the RHE perform well. To formulate a robust MPC, the estimation error can be considered in the design of the RCIS and the constraint tightening. Then, the resulting MPC approach is robust against all uncertainties including the estimation errors.

All in all, the results in this chapter demonstrate that the MPC approaches can be combined with common estimation approaches. In particular, for the simple (non-robust) MPC approaches, the combination with state estimation is straightforward. For the robust MPCs, state estimation may introduce significant conservatism to the design of the RCIS and the constraints. In the context of the hot stamping process, state estimation could be used for the estimation of tool temperatures as shown in [116], or the estimation of model errors.

Part IV.

Control of Distributed Jump Markov Systems

10. MPC for Distributed Systems

In the previous part, robust MPC for single JMS has been considered. In this part, the control of the overall manufacturing system, i.e., of multiple coupled JMS, is considered. In terms of control performance, a centralized MPC would be favorable. However, due to the following reasons, decentralized and distributed MPC setups are considered in this part instead:

- The overall manufacturing system is characterized by an asynchronous sampling of the subsystems (see Section 3.3). Thus, a centralized controller setup, determining inputs for all subsystems at the same time, is not suitable.

- The computation times of a centralized MPC may be prohibitively large.

- The communication effort for a centralized MPC is relatively high.

In what follows, the requirements for the control of the distributed production system are specified, and a literature overview is provided. Subsequently, different MPC architectures, which are suitable for the problem at hand, are discussed. The MPC design for the different architectures is presented in Chapter 11 in detail.

Requirements for the MPC Design

In principle, the requirements stated for the control of a single production unit (see Section 4.4) also have to be considered for the overall production system. The distributed MPCs must be recursively feasible, guarantee that constraints are satisfied robustly, and minimize the costs specified for the whole production process. To minimize the overall costs, it can be desirable that one subsystem deviates from its reference in order to minimize the costs in the downstream processes. Hence, PMSS of the subsystems is not required in the distributed setup, i.e., only boundedness of the states and constraint satisfaction is required in this part. In addition, the computation times of the local MPCs have to be low enough. Considering processes like hot stamping, computation times below 1 second are needed.

10.1. Literature Review

This section provides a brief overview of general MPC approaches for distributed systems. It serves as a basis to identify and develop suitable MPC approaches for the distributed production system. Finally, the state of the art in control for distributed JMS is presented.

Model Predictive Control for Distributed Systems

This section is meant to provide an overview of MPC concepts for distributed systems, rather than an in-depth classification of the different approaches. To this end, only overview articles, books, and representative approaches for the individual groups are cited. In general, the MPC approaches can be categorized by means of the underlying architecture [113]: **centralized** approaches, **decentralized** approaches, **distributed** approaches, and **hierarchical** approaches (cf. Section 1.2.1 and Fig. 1.3 on page 9). Since centralized approaches are not suitable for distributed production systems, these approaches are not considered. In literature, there is no generally accepted distinction between the notions of "decentralized" and "distributed" architectures. In this thesis, both architectures are distinguished by the fact whether communication between the local MPC laws is performed or not, regardless of the usage of the information transmitted by the controllers. As soon as communication is performed, the approaches are called distributed MPC (DMPC). For a thorough overview of non-centralized MPC approaches, see the review papers [35, 93] and the book [91].

Decentralized MPC   In decentralized MPC, all subsystems are controlled by local controllers independently of each other. The influence of the couplings is unknown to the controllers, i.e., stability can only be guaranteed if the couplings are weak [35, 113]. Considering the couplings as additional disturbances, robust MPC approaches (see Section 2.5) can be used as local controllers. Such an approach is proposed in [92], employing a robust MPC formulation for discrete-time nonlinear subsystems.

Distributed MPC   In contrast, DMPC is based on communication between the local controllers. At each sampling instant, the local controllers share the predicted input and state trajectories with the other subsystems. By this means, a significantly better performance concerning costs, stability, and robustness can be achieved [35]. The approaches can be categorized by the formulation of the (local) cost functions:

- **Non-cooperative DMPC** approaches minimize local cost functions, independently of the effects on the other subsystems, and communicate the resulting input and state trajectories to the remaining subsystems [35, 91]. A robust non-cooperative DMPC approach based on local robust tube-based MPCs (cf. Section 2.5) is proposed in [45]. The deviations from the predicted trajectories are considered as bounded disturbances. The boundedness of the deviations is guaranteed by additional constraints forcing the local state trajectories to be in the defined neighborhood of the communicated trajectories.

- In contrast, **cooperative DMPC** approaches minimize a common (global) cost function and share the resulting input and state trajectories with the other subsystems [35, 91]. See, e.g., [109, 126] for cooperative DMPC approaches, where all local MPCs minimize the same convex combination of all local

cost functions. For serially coupled systems, such as the production systems considered in this thesis, the local cost functions reduce to the costs of the considered subsystem and all downstream subsystems [109].

In addition, DMPC approaches can be categorized as *sequential* or as *parallel* approaches. In sequential approaches, the input trajectories of the subsystems are calculated successively in a predefined order, while parallel approaches perform the local optimizations simultaneously [35]. Parallel approaches can be further distinguished into approaches that perform optimization and communication once in each sampling interval, and approaches that iterate over optimization and trajectory communication until a stopping criteria is met (cf. [109, 126]). See [91, Cha. 30] for a comparison of sequential and iterative robust DMPC approaches.

Distributed Control of JMS

There are few approaches for the design of feedback controllers for distributed JMLS, such as robust decentralized state feedback for continuous-time JMLS (see, e.g., [87]), or distributed state feedback for discrete-time JMLS [129]. In addition, specialized feedback design approaches for distributed systems with failure-prone communication networks modeled by Markov chains [13, 64] or for systems comprising identical JML subsystems are considered [140]. However, to the best of the author's knowledge, there are only two MPC approaches for distributed JMLS [123, 124]. Both present iterative cooperative DMPCs that solve SDPs in each iteration online. Hence, these approaches are only suitable for small scale problems. In addition, no decentralized or distributed MPC approaches are considered for nonlinear JMS in literature. As a consequence, decentralized MPC and DMPC approaches, that can be applied to distributed nonlinear JMS, are considered in this part.

10.2. Control Setup

This section describes the general control setup for the distributed production system. After a short review of the underlying coupling structure, the implications for the communication structure and MPC design are described.

Characterization of the Couplings

The coupling structure is essential for the design of DMPC approaches. As described in Section 3.3, the coupling of the subsystems is caused exclusively by the properties of the processed products. In addition, only acyclic processing sequences are considered (Assumptions 3.2 and 3.3). For notional convenience, the results are presented for the case that each production unit has not more than **one** predecessor and successor. The predecessors and successors may change with time (e.g., in an alternating pattern). The extension to multiple predecessors and successors is
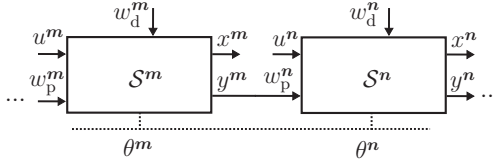
Figure 10.1.: Illustration of the dynamic couplings of two consecutive subsystems.

straightforward and the corresponding changes to the presented approaches will be described where necessary. The couplings are defined by the implication (3.11) on page 47. It states that the product-related outputs y^m become the product-related disturbances of the successor unit w_p^n with $n = \text{succ}(m, k_b)$. The coupling structure for two subsequent subsystems is illustrated in Fig. 10.1. Using the fact that the outputs $y^m[k]$ are included in the augmented state vector $\hat{x}^m[k+1]$ (cf. Section 5.2; the hat $\hat{\cdot}$ will be left out in the following derivations), the couplings are defined by:

$$w_p^n[k] = y^m[k] = T_y^m x^m[k+1], \tag{10.1}$$
$$= T_y^m f_{\theta_k}^m\big(x^m[k], u^m[k], w^m[k]\big) \tag{10.2}$$

for all times k and $(m, n) \in \mathfrak{E}[k]$. Here, T_y^m is a matrix that "selects" the product-related outputs from $x^m[k]$. The local disturbance vectors are defined by the product-related disturbances and the local disturbances $w^n[k] = \big[(w_p^n[k])^\top \ (w_d^n[k])^\top\big]^\top$. The plant model mismatch processes $\nu_x^n[k]$ and $\nu_y^n[k]$ are included in the vector $w_d^n[k]$.

Remark 10.1. *Note that the time indices k of the subsystems $\mathcal{S}^m$ and $\mathcal{S}^n$, i.e., $k = \kappa_m(k_b)$ and $k = \kappa_n(k_b)$, may be different. The index k is used without index in order to present the results in a clearer notation.*

Control Architectures and MPC Design Concept

Considering the coupling structure, the following control and communication architectures apply for the distributed manufacturing system (cf. Fig. 10.2):

- In the **decentralized MPC architecture** no communication is performed between the local controllers (see Fig. 10.2 (a)). This architecture is suitable for old manufacturing systems without modern communication networks. Large disturbances caused by neglected couplings, however, may lead to a decreased performance and robustness.

- In the **non-cooperative DMPC architecture**, the local controllers transmit the predicted output trajectories, i.e., the properties of the transferred products, to the direct successors. The successors use the predicted trajectories in order to determine the expected disturbance trajectory $\bar{w}_p^n[k]$. This architecture requires direct communication of consecutive production units.
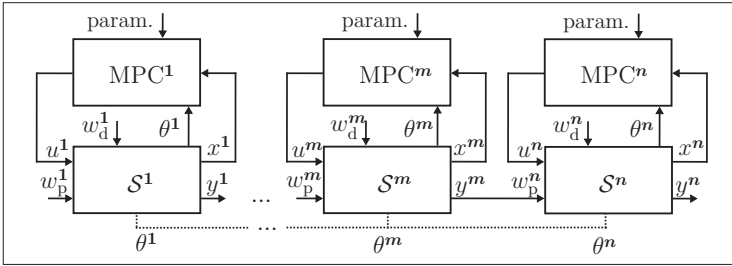
(a) Decentralized MPC architecture
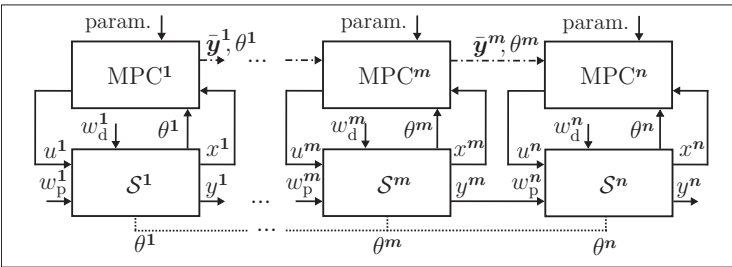
(b) Non-cooperative DMPC architecture
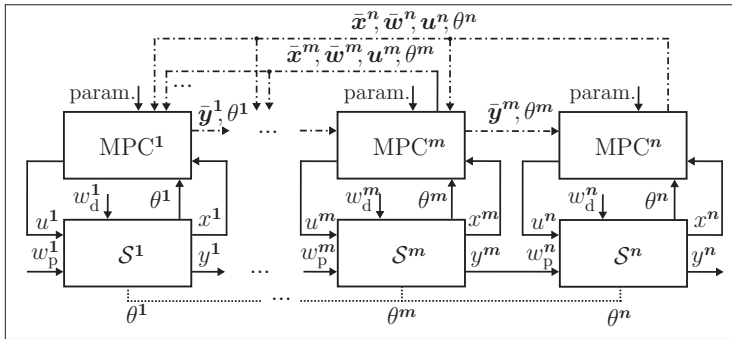
(c) Cooperative DMPC architecture

Figure 10.2.: Illustration of decentralized and distributed MPC architectures, including physical signals (solid), Markov states couplings (dotted), communication (dash-dotted), and the MPC parameterization. The indices m and n are used to reference arbitrary subsystems in this part (not necessarily the last ones).

- To minimize the global cost criterion in a **cooperative DMPC architecture**, each local controller needs to consider the costs of the considered subsystem and all downstream subsystems, since these subsystems are also affected by the actions of the considered subsystem (cf. [109, 156]). To predict these costs, the controllers need the predicted state, input, and disturbance trajectories of all succeeding subsystems as well as the predicted output trajectory of the preceding subsystem. Hence, each subsystem transmits the corresponding trajectories and the current Markov state to all predecessors and the predicted output trajectory to the direct successors (see Fig. 10.2 (c)). For this setup, a complete communication graph is not needed due to the sequential coupling of the subsystems (sequential and asynchronous processing of the products).

In order to implement these MPC architectures employing the MPC approaches proposed in the previous part, different modifications are necessary. The following paragraphs illustrate how the two-step design approach for robust MPC has to be adapted. To this end, it is assumed that the communication is faultless and that transmission times can be neglected. For most modern manufacturing systems equipped with modern communication networks, this assumption is reasonable.

Decentralized MPC Architecture: The only difference to an MPC for single JMS is a potentially larger error set caused by the couplings (product properties). The larger error sets have to be considered throughout the design of the RCIS. Then, the local MPC approaches can be used as presented in Chapter 8 without conceptual changes. The whole design procedure is presented in Section 11.1.

Non-Cooperative DMPC Architecture: As for the decentralized architecture, local disturbances and couplings have to be considered for the determination of the local error sets and the RCIS. However, the size of the error sets can be reduced significantly, since the expected value of the couplings $\bar{\boldsymbol{w}}_{\mathrm{p}}^{\boldsymbol{n}}[k]$ can be determined directly from the transmitted predictions $\bar{\boldsymbol{y}}^{\boldsymbol{n}}[k]$. An approach for the RCIS design, which employs this additional information, is presented in Section 11.2.1. Once the RCIS have been determined, the MPC formulation proposed in Chapter 8 can be used without changes by employing the new definition of the disturbance vector.

Cooperative DMPC Architecture: The RCIS can be determined in the same manner as for the non-cooperative architecture, since the disturbances and couplings as well as their predicted values are the same. However, the cost function formulation, employed in the previous chapters, has to be extended in order to predict the costs caused in all downstream subsystems. Suitable cost function formulations are introduced in Section 11.2.3. The asynchronous sampling of the production units renders parallel and iterative MPC approaches impractical. Hence, in this thesis, only sequential DMPC setups are considered.

11. Decentralized and Distributed Model Predictive Control

In order to control a manufacturing system consisting of coupled production units, the couplings have to be considered in the design of the local MPCs. To this end, approaches for the design and implementation of robust decentralized and distributed MPC for nonlinear JMS are presented in this section. For the sake of brevity, the special case of JMLS is not considered separately. It is assumed that:

Assumption 11.1. *For all subsystems n, a robust MPC can be designed as presented in Chapter 8, if the couplings are neglected. Hence, for all subsystems, polytopes $\mathbb{A}_i^n$, $\mathbb{B}_i^n$, and $\mathbb{G}_i^n$ with $i \in \Theta^n$, defining polytopic linear differential inclusions, have to exist. In addition, for all subsystems $n \in \mathbb{N}_{n_s}$, ellipsoidal sets $\mathcal{E}_{w,i}^n = \mathcal{E}(E_i^n)$, that contain all possible values of the prediction error caused by the linearization and the local disturbances $w_d^n[k]$, exist for all $i \in \Theta^n$.*

This assumption is necessary to guarantee that RCIS can be designed for decentralized and distributed MPC setups based on the results presented in Chapter 8. The procedure for a decentralized setup is explained in the following section. The DMPC setups are considered in Section 11.2. Both approaches are compared in a simulation study in Section 11.3.

11.1. Decentralized MPC Architecture

This section illustrates the design of a decentralized MPC setup for the manufacturing systems considered. As for a single JMS, all RCIS defined by $\mathcal{E}_i^n = \mathcal{E}(\mathcal{Q}_i^n)$ and local control laws $u_\delta^n[k] = K_{\theta_k}^n[k]x_\delta^n[k]$ are calculated offline for all subsystems $n \in \mathbb{N}_{n_s}$. The RCIS are used to formulate robust local MPC problems.

11.1.1. Design of Robust Control Invariant Sets

In a decentralized setup, the couplings are considered as additional unknown disturbances. Hence, the overall disturbance $(w^n[k])^\top = \left[(w_p^n[k])^\top \quad (w_d^n[k])^\top \right]$ of the n-th subsystem with $n \in \mathbb{N}_{n_s}$ is defined by the couplings, i.e., the product-related disturbances, and the local process disturbances. In principle, the RCIS design approach presented in the Sections 7.2 and 8.1 can be applied independently for each subsystem if ellipsoidal error sets are known that contain the disturbance $w^n[k]$. The

values of the couplings $w_\mathrm{d}^n[k]$ depend on the MPC design of the preceding subsystem. Hence, the RCIS of the different subsystems influence each other, and an independent RCIS design is inappropriate. An approach for a coordinated RCIS design is presented in this section. To this end, it is assumed that the reference $\xi_{\mathrm{r},i}^n$ with $i \in \Theta^n$ defines a steady state for all subsystems $n \in \mathbb{N}_{n_\mathrm{s}}$ (cf. Assumption 8.1). The corresponding reference system formulations ($x_\delta^n[k] := x^n[k] - x_\mathrm{r}^n$, $u_\delta^n[k] := u^n[k] - u_{\mathrm{r},\theta_k}^n$, etc.) are employed.

Based on this setup, the linear time-variant representation of the n-th JMS (cf. (8.11)) can be extended by considering the influence of the coupling as follows:

$$x_\delta^n[k+1] = \tilde{A}_{\theta_k}^n[k]\, x_\delta^n[k] + \tilde{B}_{\theta_k}^n[k]\, u_\delta^n[k] + \tilde{G}_{\mathrm{p},\theta_k}^n[k]\, w_{\mathrm{p},\delta}^n[k] + e_{\theta_k}^n[k]. \qquad (11.1)$$

In this equation, $e_i^n[k] \in \mathcal{E}\big(E_i^n\big)$ with $i \in \Theta^n$ represents all local disturbances but the couplings (linearization errors and effects of the disturbances $w_\mathrm{d}^n[k]$). For the RCIS design, the overall error set $\mathbb{E}_i^n$ of the one-step prediction $x_\delta^n[1]$ has to be considered. It results from the coupling $w_{\mathrm{p},\delta}^n[k]$ and the local disturbances $e_i^n[k]$. The coupling $w_{\mathrm{p},\delta}^n[k]$ is defined by the output $y_\delta^m[k]$ of the predecessor, which is influenced by the MPC in subsystem m in time k (cf. (10.2)). Considering the closed-loop dynamics with the local controller $K_i^m[k]$ with $i \in \Theta^m$ of the m-th subsystem, it holds that:

$$y_\delta^m[k] = T_\mathrm{y}^m \left[\big(\tilde{A}_{\theta_k}^m[k] + \tilde{B}_{\theta_k}^m[k]\, K_{\theta_k}^m[k]\big) x_\delta^m[k] + \tilde{G}_{\mathrm{p},\theta_k}^m[k]\, w_{\mathrm{p},\delta}^m[k] + e_{\theta_k}^m[k] \right]. \qquad (11.2)$$

Using this result, the following recursive error set approximation can be derived:

$$\mathbb{E}_i^1 = \mathcal{E}\big(E_i^1\big), \qquad\qquad\qquad \forall i \in \Theta^1 \quad (11.3)$$

$$\mathbb{E}_i^n = \tilde{G}_{\mathrm{p},i}^n[k]\, T_\mathrm{y}^m \left[\big(\tilde{A}_{i_2}^m[k] + \tilde{B}_{i_2}^m[k]\, K_{i_2}^m[k]\big) \mathcal{E}\big(\mathscr{Q}_{i_2}^m\big) \oplus \mathbb{E}_{i_2}^m \right] \oplus \mathcal{E}\big(E_i^n\big) \qquad (11.4)$$

for all $i \in \Theta^n$ and $i_2 \in \Theta^m$ with $m = \mathrm{pre}(n)$. If systems with multiple predecessors or successors are considered, the local error sets $\mathbb{E}_i^n$ have to be determined by the Minkowski sum of the couplings of all subsystems involved.

Basically, there are two approaches to calculate the RCIS for the subsystems employing the error set relations in (11.4):

- The RCIS can be calculated successively from the first to the last subsystem employing the formulation for a single JMS presented in Section 7.2. The procedure starts with solving the SDP (7.55) for the first subsystem with the polytopes $\mathbb{A}_i^1$, $\mathbb{B}_i^1$, and $\mathbb{G}_i^1$ and the error set $\mathbb{E}_i^1$. Based on the resulting RCIS, the error set $\mathbb{E}_i^2$ can be calculated employing (11.4). Then, the SDP (7.55) can be solved with this error set and the polytopes $\mathbb{A}_i^2$, $\mathbb{B}_i^2$, and $\mathbb{G}_i^2$, and so forth. Since the volume of the RCIS is maximized, the couplings are maximized as well. Hence, it is expected that this approach often results in infeasible SDPs for the last subsystems. Alternatively, the cost function of the SDP (7.55) can be modified in order to minimize the volume of the output set $T_\mathrm{y}^m \mathcal{E}\big(\mathscr{Q}_i^m\big)$. This formulation more likely results in feasible SDPs for all subsystems. The resulting RCIS, however, are of small volume and result in restrictive constraints for the MPC.

- The RCIS can be determined simultaneously for all subsystems considering the interconnections between the RCIS defined by (11.4) as additional constraints. This approach is presented in this section. Due to the simultaneous computation of the RCIS, the overall volume of the RCIS can be maximized.

SDP Formulation

In order to determine RCIS and controllers simultaneously for all subsystems, the optimization variables used in the SDP (7.55) are introduced for all subsystems $n \in \mathbb{N}_{n_s}$ with appropriate dimensions: $\mathcal{Q}^n$, $\mathcal{G}^n$, $\mathcal{Y}^n$, and ϱ^n. In the SDP (7.55), ellipsoidal error sets are employed. Hence, the error sets $\mathbb{E}_i^n$ with $i \in \Theta^n$ defined by (11.4) are closely over-approximated by ellipsoidal sets $\mathcal{E}(\mathscr{E}_i^n) \supseteq \mathbb{E}_i^n$. The shape matrices $\mathscr{E}_i^n \in \mathbb{S}_{n_x^n}^{>0}$ of these error sets are introduced as additional matrix variables. These variables are collected in the set $\mathscr{E}^n := \{\mathscr{E}_i^n : i \in \Theta^n\}$. It is demonstrated below how these variables can be used to formulate BMI constraints that couple the local SDP formulations such that all RCIS can be calculated by solving one overall SDP.

The error set defined in (11.4) is given by a Minkowski sum of three ellipsoids. This sum can be approximated in two steps:

$$\acute{\mathbb{E}}_{i_2}^m \supseteq T_y^m \left(\tilde{A}_{i_2}^m[k] + \tilde{B}_{i_2}^m[k] \, K_{i_2}^m[k] \right) \mathcal{E}\left(\mathcal{Q}_{i_2}^m \right) \oplus T_y^m \mathbb{E}_{i_2}^m, \tag{11.5}$$

$$\mathbb{E}_i^n \supseteq \tilde{G}_{\mathrm{p},i}^n[k] \, \acute{\mathbb{E}}_{i_2}^m \oplus \mathcal{E}(E_i^n) \tag{11.6}$$

for all $i \in \Theta^n$ and $i_2 \in \Theta^m$ with $m = \mathrm{pre}(n)$. Let $\acute{\mathscr{E}}_{i_2}^m \in \mathbb{S}_{n_y^m}^{>0}$ define shape matrices for ellipsoidal over-approximations $\mathcal{E}\left(\acute{\mathscr{E}}_{i_2}^m \right) \supseteq \acute{\mathbb{E}}_{i_2}^m$. These matrices are used as matrix variables for the overall SDP. They are collected in the set $\acute{\mathscr{E}}^m := \{\acute{\mathscr{E}}_{i_2}^m : i_2 \in \Theta^m\}$. Employing the ellipsoidal over-approximations $\mathcal{E}(\mathscr{E}_i^n)$ and $\mathcal{E}(\acute{\mathscr{E}}_{i_2}^m)$ of $\mathbb{E}_i^n$ and $\acute{\mathbb{E}}_{i_2}^m$, the conditions for the Minkowski sums in (11.5) and (11.6) can be formulated as matrix inequalities. According to Corollary 2.1, the approximations in (11.5) and (11.6) hold, if parameters $\acute{\alpha}_{i_2}^m \in (0,1)$ and $\alpha_{i,i_2}^n \in (0,1)$ exist such that:

$$\acute{\mathscr{E}}_{i_2}^m \geq \left(\acute{\alpha}_{i_2}^m \right)^{-1} T_y^m \left(\tilde{A}_{i_2}^m[k] + \tilde{B}_{i_2}^m[k] \, K_{i_2}^m[k] \right) \mathcal{Q}_{i_2}^m \left(T_y^m \left(\tilde{A}_{i_2}^m[k] + \tilde{B}_{i_2}^m[k] \, K_{i_2}^m[k] \right) \right)^\top \tag{11.7}$$
$$+ \left(1 - \acute{\alpha}_{i_2}^m \right)^{-1} T_y^m \, \mathscr{E}_{i_2}^m \, (T_y^m)^\top$$

$$\mathscr{E}_i^n \geq \left(\alpha_{i,i_2}^n \right)^{-1} \tilde{G}_{\mathrm{p},i}^n[k] \, \acute{\mathscr{E}}_{i_2}^m \left(\tilde{G}_{\mathrm{p},i}^n[k] \right)^\top + \left(1 - \alpha_{i,i_2}^n \right)^{-1} E_i^n \tag{11.8}$$

for all $i \in \Theta^n$ and $i_2 \in \Theta^m$. According to the reasoning in Section 7.1.1 and Lemma 7.3, both matrix inequalities hold if the following BMIs hold:

$$\begin{bmatrix} \acute{\mathscr{E}}_{i_2}^m & T_y^m \left(A_{i_2,[l_A]}^m \mathscr{G}_{i_2,[l_B]}^m + B_{i_2,[l_B]}^m \mathscr{Y}_{i_2,[l_A]}^m \right) & T_y^m \mathscr{E}_{i_2}^m \\ \star & \acute{\alpha}_{i_2}^m \left(\left(\mathscr{G}_{i_2,[l_B]}^m \right)^\top + \mathscr{G}_{i_2,[l_B]}^m - \mathcal{Q}_{i_2}^m \right) & \mathbf{0} \\ \star & \star & \left(1 - \acute{\alpha}_{i_2}^m \right) \mathscr{E}_{i_2}^m \end{bmatrix} \geq 0 \tag{11.9}$$

$$\begin{bmatrix} \mathcal{E}_i^n & G_{\text{p},i,[l_G]}^n \, \acute{\mathcal{E}}_{i_2}^m & E_i^n \\ \star & \alpha_{i,i_2}^n \, \acute{\mathcal{E}}_{i_2}^m & \mathbf{0} \\ \star & \star & \left(1 - \alpha_{i,i_2}^n\right) E_i^n \end{bmatrix} \geq 0 \tag{11.10}$$

for all $i \in \Theta^n$ and $i_2 \in \Theta^m$ and for all vertices of the matrix polytopes $l_A \in \mathbb{N}_{n_{\text{A},i_2}^m}$, $l_B \in \mathbb{N}_{n_{\text{B},i_2}^m}$, and $l_G \in \mathbb{N}_{n_{\text{G},i}^n}$. These constraints are only necessary for subsystems with predecessors. For the first subsystem, the error set is defined by $\mathcal{E}_i^1 = E_i^1$ with $i \in \Theta^1$. For subsystems with time-variant successors or predecessors, the condition (11.10) has to hold for all combinations of successors and predecessors.

Invariance Conditions: If the error set $\mathcal{E}(\mathcal{E}_i^n)$ is used instead of the local disturbance set $\mathcal{E}(E_i^n)$, the invariance conditions can be stated as a single JMS (cf. Equation (7.55b)). Hence, robust invariance is guaranteed by the following BMIs:

$$\begin{bmatrix} \mathcal{Q}_{i_2}^n & A_{i,[l_A]}^n \mathcal{G}_{i,[l_B]}^n + B_{i,[l_B]}^n \mathcal{Y}_{i,[l_A]}^n & \mathcal{E}_i^n \\ \star & \varrho_{i,i_2}^n \left(\left(\mathcal{G}_{i,[l_B]}^n \right)^\top + \mathcal{G}_{i,[l_B]}^n - \mathcal{Q}_i^n \right) & \mathbf{0} \\ \star & \star & \left(1 - \varrho_{i,i_2}^n\right) \mathcal{E}_i^n \end{bmatrix} \geq 0 \tag{11.11}$$

for all subsystems $n \in \mathbb{N}_{n_s}$, all Markov states i, $i_2 \in \Theta^n$, and for all vertices of the matrix polytopes $l_A \in \mathbb{N}_{n_{\text{A},i}^n}$ and $l_B \in \mathbb{N}_{n_{\text{B},i}^n}$. Since the error set $\mathcal{E}(\mathcal{E}_i^n)$ consists of the local disturbances and couplings, the RCIS are invariant with respect to both local disturbances and couplings.

Input and State Constraints: The input constraints are neither influenced by the local disturbances nor by the couplings. Thus, the LMIs (7.55d) presented in Section 7.2 can be used for all subsystems $n \in \mathbb{N}_{n_s}$ in order to guarantee that the local controllers satisfy the input constraints:

$$\begin{bmatrix} \left(h_{\text{u},l_u}^n - H_{\text{u},l_u}^n \, u_{\text{r},i}^n \right)^2 & H_{\text{u},l_u}^n \, \mathcal{Y}_{i,[l_A]}^n \\ \star & \left(\mathcal{G}_{i,[l_B]}^n \right)^\top + \mathcal{G}_{i,[l_B]}^n - \mathcal{Q}_i^n \end{bmatrix} \geq 0 \tag{11.12}$$

for all $l_u \in \mathbb{N}_{n_{\text{h},u}^n}$, $l_A \in \mathbb{N}_{n_{\text{A},i}^n}$, $l_B \in \mathbb{N}_{n_{\text{B},i}^n}$, and $i \in \Theta^n$.

The state constraints have to be tightened by the overall error set $\mathcal{E}(\mathcal{E}_i^n)$ instead of the local disturbance sets. The set containing the states at time $k + 1$ can be determined by the Minkowski sum of the one-step prediction of the closed-loop system and the overall error set:

$$x_\delta^n[k+1] \in \left(\tilde{A}_{\theta_k}^n[k] + \tilde{B}_{\theta_k}^n[k] \, K_{\theta_k}^n[k] \right) \mathcal{E}\left(\mathcal{Q}_{\theta_k}^n \right) \oplus \mathcal{E}\left(\mathcal{E}_{\theta_k}^n \right). \tag{11.13}$$

According to Corollary 2.1, the shape matrix of an outer ellipsoidal approximation of this set is a member of the following family of shape matrices:

$$\left(\varepsilon_i^n \right)^{-1} \left(\tilde{A}_i^n[k] + \tilde{B}_i^n[k] K_i^n[k] \right) \mathcal{Q}_i^n \left(\tilde{A}_i^n[k] + \tilde{B}_i^n[k] K_i^n[k] \right)^\top + \left(1 - \varepsilon_i^n \right)^{-1} \mathcal{E}_i^n \tag{11.14}$$

for all $i \in \Theta^n$ and $\varepsilon_i^n \in (0,1)$. As presented in Section 7.1.1, the state constraints $H_x^n x^n[k] \leq h_x^n$ hold if these constraints hold for the support functions of ellipsoids defined by the shape matrix (11.14). These constraints can be formulated as follows, employing the controller parameterization $K_i^n[k] = \mathscr{Y}_i^n[k](\mathscr{G}_i^n[k])^{-1}$:

$$\left[H_{x,l_x}^n \left(\left(\varepsilon_i^n\right)^{-1} \left(\tilde{A}_i^n[k] + \tilde{B}_i^n[k]\mathscr{Y}_i^n[k](\mathscr{G}_i^n[k])^{-1} \right) \mathscr{Q}_i^n \left(\tilde{A}_i^n[k] + \tilde{B}_i^n[k]\mathscr{Y}_i^n[k](\mathscr{G}_i^n[k])^{-1} \right)^\top + \dots \right.$$
$$\left. \dots + \left(1 - \varepsilon_i^n \right)^{-1} \mathscr{E}_i^n \right) \left(H_{x,l_x}^n \right)^\top \right]^{1/2} \leq h_{x,l_x}^n - H_{x,l_x}^n x_r^n \qquad (11.15)$$

for all $i \in \Theta^n$ and $l_x \in \mathbb{N}_{n_{h,x}^n}$. The constraints (11.15) can be formulated as time-dependent BMIs by applying the Schur complement, a congruence transformation with $\mathscr{T} = \mathrm{diag}(I, \mathscr{G}_i^n, \mathscr{E}_i^n)$, and the approximation $(\mathscr{G}_i^n)^\top (\mathscr{Q}_i^n)^{-1} \mathscr{G}_i^n \geq (\mathscr{G}_i^n)^\top + \mathscr{G}_i^n - \mathscr{Q}_i^n$ (cf. Lemmata 2.3 - 2.5). According to Lemma 7.3, these BMIs hold if corresponding BMIs hold for all vertices of the matrix polytopes $\mathbb{A}_i^n$ and $\mathbb{B}_i^n$. Hence, the constraints (11.15) are satisfied if the following BMIs:

$$\begin{bmatrix} \left(h_{x,l_x}^n - H_{x,l_x}^n x_r^n \right)^2 & H_{x,l_x}^n \left(A_{i,[l_A]}^n \mathscr{G}_{i,[l_B]}^n + B_{i,[l_B]}^n \mathscr{Y}_{i,[l_A]}^n \right) & H_{x,l_x}^n \mathscr{E}_i^n \\ \star & \varepsilon_i^n \left(\left(\mathscr{G}_{i,[l_B]}^n \right)^\top + \mathscr{G}_{i,[l_B]}^n - \mathscr{Q}_i^n \right) & \mathbf{0} \\ \star & \star & (1 - \varepsilon_i^n)\mathscr{E}_i^n \end{bmatrix} \geq 0 \qquad (11.16)$$

hold for all $l_x \in \mathbb{N}_{n_{h,x}^n}, l_A \in \mathbb{N}_{n_{A,i}^n}, l_B \in \mathbb{N}_{n_{B,i}^n}$, and $i \in \Theta^n$.

Complete SDP: Let, for notional convenience, all scaling parameters $\alpha_{i,i_2}^n, \acute{\alpha}_{i_2}^n, \varepsilon_i^n$, and ϱ_{i,i_2}^n be collected in the vectors $\alpha^n, \acute{\alpha}^n, \varepsilon^n$, and ϱ^n. Combining the modifications presented in this section with the results from Section 7.2, the following SDP can be stated to determine the RCIS and the controllers:

Theorem 11.1. *Let $\mathscr{Q}^n, \mathscr{G}^n$, and $\mathscr{Y}^n$ be a solution of the SDP:*

$$\max_{\substack{\mathscr{Q}^n,\mathscr{G}^n,\mathscr{Y}^n,\mathscr{E}^n \\ \mathscr{E}^n,\alpha^n,\acute{\alpha}^n,\varepsilon^n,\varrho^n}} \sum_{n=1}^{n_s} \sum_{i=1}^{n_\theta^n} \log \det(\mathscr{Q}_i^n), \qquad (11.17a)$$

$$\text{s.t.} \quad \alpha_{i,i_2}^n, \ \acute{\alpha}_{i_2}^n, \ \varepsilon_i^n, \ \varrho_{i,i_2}^n \in (0,1), \qquad (11.17b)$$

$$(11.9), (11.10), (11.11), (11.12), (11.16) \qquad (11.17c)$$

for all $n \in \mathbb{N}_{n_s}$ and all indices $i, i_2, l_x, l_u, l_A, l_B$, and l_G according to the subsystem dimensions. Then, the ellipsoids $\mathcal{E}(\mathscr{Q}_i^n)$ define RCIS that consider both local disturbances and couplings. The local controllers $K_i^n[k] = \mathscr{Y}_i^n[k](\mathscr{G}_i^n[k])^{-1}$ hold the states of the coupled JMS robustly in the RCIS and satisfy the constraints.

Proof. The proof follows from Theorem 7.3 for isolated JMS, since the dynamics of the subsystems are represented exactly by the time-variant dynamics (11.1) that are over-approximated by the matrix polytopes $\mathbb{A}_i^n$, $\mathbb{B}_i^n$, and $\mathbb{G}_i^n$, and by construction of the constraints described in this section. $\qquad \square$

Theorem 11.1 presents a design approach for RCIS for a decentralized MPC architecture. The SDP (11.17) establishes a BMI problem that can be solved by PENBMI. The problem contains a larger number of BMI constraints rendering this problem formulation only suitable for small to medium scale problems. For larger systems, it is likely that the problem is intractable and only the sequential approach described in the beginning of this section is applicable to determine the RCIS. In addition, the error sets are growing from one production stage to the next, since the local errors are aggregated. Hence, it is likely that the RCIS design fails for a larger number of subsystems. This effect is particularly strong for a decentralized setup, since the couplings are regarded as disturbances.

11.1.2. Formulation of the MPC Problem

The only difference for the local MPCs in a decentralized architecture compared to classical MPC is that the disturbances and the corresponding sets are larger due to the couplings. Hence, the formulation of the local optimization problems is the same as for a single JMS (Theorem 8.1) using the error sets $\mathcal{E}(\mathscr{E}_i^n)$ instead of $\mathcal{E}(E_i^n)$. For each subsystem, the design of the quadratic invariance constraints and the state constraint tightening is performed analogously to Section 8.1:

- The invariance constraints are formulated with an ellipsoidal inner approximation

$$\mathcal{E}\big(\breve{\mathscr{Q}}_{i,i_2}^n\big) \subseteq \mathcal{E}\big(\mathscr{Q}_{i_2}^n\big) \ominus \mathcal{E}\big(\mathscr{E}_i^n\big) \tag{11.18}$$

  of the Minkowski difference of the RCIS and the overall error set for all $n \in \mathbb{N}_{n_s}$ and all $i, i_2 \in \Theta^n$. The shape matrices $\breve{\mathscr{Q}}_{i,i_2}^n$ can be determined according to Lemma 2.2 such that the volume of the ellipsoids $\mathcal{E}\big(\breve{\mathscr{Q}}_{i,i_2}^n\big)$ is maximized.

- The state constraints are tightened with the support functions of $\mathcal{E}\big(\mathscr{E}_i^n\big)$:

$$H_{\mathrm{x},l_{\mathrm{x}}}^n x^n[1] \leq h_{\mathrm{x},l_{\mathrm{x}}}^n[k] - \|H_{\mathrm{x},l_{\mathrm{x}}}^n\|_{\mathscr{E}_{\theta_k}^n} \qquad \forall\; l_{\mathrm{x}} \in \mathbb{N}_{n_{\mathrm{h,x}}^n}, n \in \mathbb{N}_{n_s}. \tag{11.19}$$

In the SDP (11.17), the values of the outputs $y_\delta^n[k]$ are over-approximated by the set $\acute{\mathbb{E}}_{\theta_k}^n$. Robustness is guaranteed as long as the outputs $y_\delta^n[k]$ of all subsystems take values from the set $\acute{\mathbb{E}}_{\theta_k}^n$. For the controllers $K_i[k]$, this is guaranteed by the constraints of the SDP (11.17). However, since the feasible set of the MPC is larger than the input set of the controllers, this condition is not satisfied by the MPC formulation in general. Hence, additional quadratic constraints are introduced that guarantee that $y_\delta^n[k] \in \mathcal{E}\big(\acute{\mathscr{E}}_{\theta_k}^n\big)$. Obviously, this is only necessary for subsystems with successors. To ensure that these constraints are robust to the disturbances and couplings, the sets $\mathcal{E}\big(\acute{\mathscr{E}}_i^n\big)$ are tightened with the overall error sets $\mathcal{E}\big(\mathscr{E}_i^n\big)$. To this end, an ellipsoidal inner approximation

$$\mathcal{E}\big(\breve{\mathscr{Q}}_{\mathrm{y},i}^n\big) \subseteq \mathcal{E}\big(\acute{\mathscr{E}}_i^n\big) \ominus T_{\mathrm{y}}^n \mathcal{E}\big(\mathscr{E}_i^n\big) \tag{11.20}$$
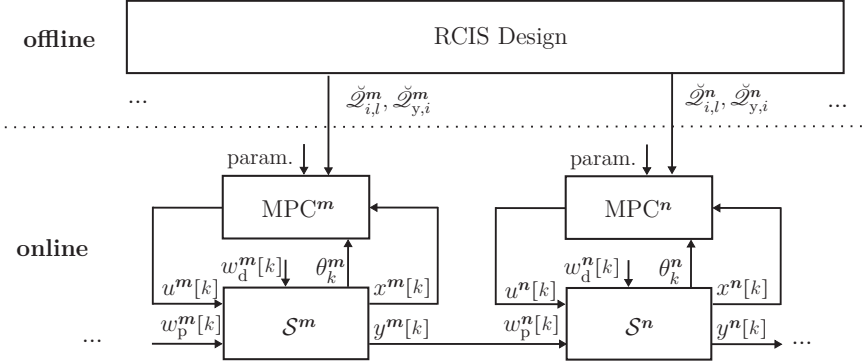
Figure 11.1.: Sketch of the design and implementation of the robust decentralized MPC scheme for two subsystems. The dots indicate that the overall system may consist of more than these two subsystem.

is used for all $n \in \mathbb{N}_{n_s}$ and all $i \in \Theta^n$. The shape matrices $\breve{\mathscr{Q}}_{y,i}^n \in \mathbb{S}_{n_y}^{>0}$ are determined according to Lemma 2.2 such that the volume is maximized.

The complete design and implementation procedure for the decentralized MPC is sketched in Fig. 11.1. It can be seen that the RCIS and output sets (defined by $\breve{\mathscr{Q}}_{i,i_2}^n$ and $\breve{\mathscr{Q}}_{y,i}^n$) are designed offline for all subsystems in parallel. These sets are used online by the local MPCs to guarantee recursive feasibility, despite the fact that the subsystems do not communicate or cooperate with each other.

Once a subsystem $n \in \mathbb{N}_{n_s}$ receives a new product group (at each time k), the subsystem performs a linearization of the dynamics as described in Section 8.1. Then, the resulting matrices $A_{\theta_k}^n[k]$, $B_{\theta_k}^n[k]$, and $G_{\theta_k}'^n[k]$ as well as the extended disturbance vector $\bar{w}'^n[k]$ (cf. page 120) are used to formulate the optimization problem to be solved in each cycle k:

$$\min_{\boldsymbol{u}^n[k]} \|\boldsymbol{u}^n[k]\|_{W^n[k]}^2 + q^n[k]\,\boldsymbol{u}^n[k] \tag{11.21a}$$

$$\text{s.t. } \boldsymbol{H}_u^n \boldsymbol{u}^n[k] \le \boldsymbol{h}_u^n[k], \tag{11.21b}$$

$$H_{x,l_x}^n \left(A_{\theta_k}^n[k]\,x^n[k] + B_{\theta_k}^n[k]\,u^n[\![0]\!] + G_{\theta_k}'^n[k]\,\bar{w}'^n[k] \right) \le h_{x,l_x}^n[k] - \|H_{x,l_x}^n\|_{\mathscr{E}_{\theta_k}^n}, \tag{11.21c}$$

$$\left\| A_{\theta_k}^n[k]\,x^n[k] + B_{\theta_k}^n[k]\,u^n[\![0]\!] + G_{\theta_k}'^n[k]\,\bar{w}'^n[k] - x_r^n \right\|_{\left(\breve{\mathscr{Q}}_{\theta_k,i}^n\right)^{-1}}^2 \le 1, \tag{11.21d}$$

$$\left\| T_y^n \left[A_{\theta_k}^n[k]\,x^n[k] + B_{\theta_k}^n[k]\,u^n[\![0]\!] + G_{\theta_k}'^n[k]\,\bar{w}'^n[k] - x_r^n \right] \right\|_{\left(\breve{\mathscr{Q}}_{y,\theta_k}^n\right)^{-1}}^2 \le 1 \tag{11.21e}$$

where $\boldsymbol{H}_u^n := I_{N^n} \otimes H_u^n$, $(\boldsymbol{h}_u^n[k])^\top := \left[(h_u^n[k])^\top \cdots (h_u^n[k+N-1])^\top \right]$, $l_x \in \mathbb{N}_{n_{h,x}^n}$, and $i \in \Theta^n$ with $p_{\theta_k,i}^n > 0$. Finally, the input $u[\![0]\!]$ is applied to the subsystem.

Theorem 11.2. *Let a solution of the SDP* (11.17) *exist and the matrices* $\breve{\mathscr{Q}}^n_{i,i_2}$ *and* $\breve{\mathscr{Q}}^m_{\mathrm{y},i}$ *be defined such that* (11.18) *and* (11.20) *hold. Finally, suppose that for all subsystems* $n \in \mathbb{N}_{n_\mathrm{s}}$, *an initial solution of the optimization problem* (11.21) *exists and that the optimization problems can be solved fast enough to apply* $u^n[\![0]\!]$ *to the subsystem. Then, the states of the coupled JMS are kept robustly in the RCIS, the optimization problems* (11.21) *are recursively feasible, and the constraints are satisfied for all subsystems.*

Proof. The input constraints of the subsystems are represented by (11.21b). Robust satisfaction of the state constraints is guaranteed by (11.21c), since the state constraints are tightened with the overall error set $\mathcal{E}\big(\mathscr{E}^n_i\big)$ that includes the disturbances and couplings. By design (Theorem 11.1), the input $u'[k] = K^n_{\theta_k}[k]\big(x^n[k]-x^n_\mathrm{r}[k]\big)+u^n_{\mathrm{r},\theta_k}$ is a feasible solution of the optimization problem (11.21), if $x^n[k] - x^n_\mathrm{r}[k] \in \mathcal{E}^n_{\theta_k}$. The constraint (11.21e) ensures that the outputs are confined to the ellipsoids $\mathcal{E}\big(\mathscr{\acute{E}}^n_{\theta_k}\big)$. In consequence, the prediction errors are bounded by the sets $\mathcal{E}\big(\mathscr{E}^n_i\big)$ and the constraints (11.21d) guarantee that $x^n[k]-x^n_\mathrm{r}[k] \in \mathcal{E}^n_{\theta_k}$ holds recursively. Thus, recursive feasibility is guaranteed. Since the states of all subsystems stay in the corresponding RCIS, the states are bounded and the overall system is stable. □

In the optimization problem (11.21), the cost matrices $W^n[k]$ and $q^n[k]$ can be chosen such that the CoE (see Theorem 6.2) or the EoC of the subsystem are minimized (see Theorem 6.3) for the linearized systems. Except for the constraint (11.21e), the local MPC formulation is the same as for a single JMS (cf. Theorem 8.1). Thus, the dimensions of the optimization problems are similar, i.e., the computation time is similar to that of an MPC for a single JMS of the same dimension.

The resulting MPC formulation robustly stabilizes the coupled JMS. However, since the couplings are considered as unknown disturbances, the resulting control performance may be relatively low. This negative effect may be enhanced by the conservatism of the SDP formulation resulting in RCIS with a low volume.

11.2. Distributed MPC Architectures

In general, the control performance can be increased if a DMPC architecture is considered, employing more precise knowledge about the coupling variables. The following sections show how the results presented so far can be used for non-cooperative as well as cooperative DMPC architectures. First, the design of RCIS for a distributed setup is presented. Subsequently, the different cost function formulations for the non-cooperative and the cooperative MPC architecture are presented.

11.2.1. Design of Robust Control Invariant Sets

The basic idea for the design of the RCIS (simultaneous determination of the RCIS for all subsystems) is the same as for the decentralized architecture presented in the

previous section. In contrast to the decentralized MPC setup, the expected value of the couplings $\bar{w}_{\mathrm{p}}^n[k]$ is known in a distributed setup. This knowledge can be used to reduce the conservatism of the RCIS design. To this end, a disturbance feedback term is added to the linear state feedback law used for the RCIS design:

$$u_\delta^n[k] = K_{\theta_k}^n[k]\, x_\delta^n[k] + \hat{K}_{\theta_k}^n\, \bar{w}_{\mathrm{p},\delta}^n[k]. \tag{11.22}$$

The closed-loop dynamics with the control law (11.22) is given by:

$$x_\delta^n[k+1] = \left(\tilde{A}_{\theta_k}^n[k] + \tilde{B}_{\theta_k}^n[k]\, K_{\theta_k}^n[k] \right) x_\delta^n[k] + \left(\tilde{G}_{\mathrm{p},\theta_k}^n[k] + \tilde{B}_{\theta_k}^n[k]\, \hat{K}_{\theta_k}^n \right) \bar{w}_{\mathrm{p},\delta}^n[k] + e_{\theta_k}^n[k]. \tag{11.23}$$

Note that the expected value $\bar{w}_{\mathrm{p},\delta}^n[k]$ is used in this form, as the MPC of the predecessor unit only predicts expected values of the outputs. Hence, linearization errors and disturbances resulting from the difference to the real $w_{\mathrm{p},\delta}^n[k]$ are incorporated[1] in the local errors $e_{\theta_k}^n[k]$. Compared to the dynamics in Equation (11.1), the only difference is that the disturbance matrix $\tilde{G}_{\mathrm{p},\theta_k}^n[k]$ is replaced by its closed-loop form shown in (11.23) and that the local error set is extended. The formulation could be stated less conservative, if time-variant disturbance feedback laws $\hat{K}_{\theta_k}^n[k]$ or a dependency on the Markov state of the predecessor would be considered. However, this would lead to an even more complex notation and SDP formulation.

SDP Formulation

The RCIS design for decentralized MPC architectures (presented in the previous section) can be used with slight modifications also for distributed architectures. The main step is to replace $\tilde{G}_{\mathrm{p},i}^n[k]$ with the closed-loop form $\tilde{G}_{\mathrm{p},i}^n[k] + \tilde{B}_i^n[k]\,\hat{K}_i^n$ for all $n \in \mathbb{N}_{n_{\mathrm{s}}}$ and $i \in \Theta^n$. To this end, the following parameterization of the disturbance feedback is introduced:

$$\hat{K}_i^n = \hat{\mathscr{Y}}_i^n \big(\hat{\mathscr{G}}_i^n \big)^{-1}, \quad i \in \Theta^n, \tag{11.24}$$

where the matrices $\hat{\mathscr{G}}_i^n \in \mathbb{R}^{n_{\mathrm{w_p}}^n \times n_{\mathrm{w_p}}^n}$ and $\hat{\mathscr{Y}}_i^n \in \mathbb{R}^{n_{\mathrm{u}}^n \times n_{\mathrm{w_p}}^n}$ are used as additional matrix variables. This modification has an impact only on the error sets $\mathbb{E}_i^n$ and the values of the inputs. Hence, the LMI and BMI constraints regarding invariance (11.11) and state constraints (11.16) can be adopted from Section 11.1.1. The modifications to the error set description and the input constraints are presented below.

Error Set Over-Approximation: Considering the additional feedback, the two-step over-approximation of the error sets $\mathbb{E}_i^n$ in (11.5) and (11.6) results in:

$$\acute{\mathbb{E}}_{i_2}^m \supseteq T_{\mathrm{y}}^m \left(\tilde{A}_{i_2}^m[k] + \tilde{B}_{i_2}^m[k]\, K_{i_2}^m[k] \right) \mathcal{E}\big(\mathscr{Q}_{i_2}^m \big) \oplus T_{\mathrm{y}}^m \mathbb{E}_{i_2}^m, \tag{11.25}$$

$$\mathbb{E}_i^n \supseteq \left(\tilde{G}_{\mathrm{p},i}^n[k] + \tilde{B}_i^n[k]\, \hat{K}_i^n \right) \acute{\mathbb{E}}_{i_2}^m \oplus \mathcal{E}\big(E_i^n \big). \tag{11.26}$$

---

[1]Since most production systems measure the properties of the products, the errors can be reduced to the measurement error in these cases.

In this formulation, the disturbance feedback controller uses the exact value of the disturbance $w_{\mathrm{p},\delta}^m[k]$. If only the expected value is known, the corresponding uncertainties are incorporated in the local disturbance set $\mathcal{E}(E_i^n)$. As for the decentralized case, the sets $\acute{\mathbb{E}}_{i_2}^m$ and $\mathbb{E}_i^n$ are over-approximated by the ellipsoids $\mathcal{E}(\acute{\mathscr{E}}_{i_2}^m)$ and $\mathcal{E}(\mathscr{E}_i^n)$. Analogously to the derivations on page 147, the set relations (11.25) and (11.26) hold for $\mathcal{E}(\acute{\mathscr{E}}_{i_2}^m)$ and $\mathcal{E}(\mathscr{E}_i^n)$, if $\acute{\alpha}_{i_2}^m \in (0,1)$ and $\alpha_{i,i_2}^n \in (0,1)$ exist such that the following BMIs hold:

$$
\begin{bmatrix}
\acute{\mathscr{E}}_{i_2}^m & T_{\mathrm{y}}^m\left(A_{i_2,[l_{\mathrm{A}}]}^m \mathscr{G}_{i_2,[l_{\mathrm{B},1}]}^m + B_{i_2,[l_{\mathrm{B},1}]}^m \mathscr{Y}_{i_2,[l_{\mathrm{A}}]}^m\right) & T_{\mathrm{y}}^m \mathscr{E}_{i_2}^m \\
\star & \acute{\alpha}_{i_2}^m\left(\left(\mathscr{G}_{i_2,[l_{\mathrm{B},1}]}^m\right)^{\top} + \mathscr{G}_{i_2,[l_{\mathrm{B},1}]}^m - \mathscr{Q}_{i_2}^m\right) & 0 \\
\star & \star & \left(1-\acute{\alpha}_{i_2}^m\right)\mathscr{E}_{i_2}^m
\end{bmatrix} \geq 0 \tag{11.27}
$$

$$
\begin{bmatrix}
\mathscr{E}_i^n & G_{i,[l_{\mathrm{G}}]}^n \hat{\mathscr{G}}_i^n + B_{i,[l_{\mathrm{B},2}]}^n \hat{\mathscr{Y}}_i^n & E_i^n \\
\star & \alpha_{i,i_2}^n\left(\left(\hat{\mathscr{G}}_i^n\right)^{\top} + \hat{\mathscr{G}}_i^n - \acute{\mathscr{E}}_{i_2}^m\right) & 0 \\
\star & \star & \left(1-\alpha_{i,i_2}^n\right)E_i^n
\end{bmatrix} \geq 0 \tag{11.28}
$$

for all Markov states $i \in \Theta^n$ and $i_2 \in \Theta^m$ with $m = \mathrm{pre}(n)$, and for all vertices of the matrix polytopes $l_{\mathrm{A}} \in \mathbb{N}_{n_{\mathrm{A},i}^m}$, $l_{\mathrm{B},1} \in \mathbb{N}_{n_{\mathrm{B},i_2}^m}$, $l_{\mathrm{B},2} \in \mathbb{N}_{n_{\mathrm{B},i}^n}$, and $l_{\mathrm{G}} \in \mathbb{N}_{n_{\mathrm{G},i}^n}$.

Input Constraints: Based on the control law (11.22) and the definition of the error sets $\acute{\mathbb{E}}_{i_2}^m$ in (11.25), the input set $\mathbb{U}^n$ can be calculated as follows:

$$
\mathbb{U}^n = K_i^n[k]\,\mathcal{E}\left(\mathscr{Q}_i^n\right) \oplus \hat{K}_i^n \acute{\mathbb{E}}_{i_2}^m + u_{\mathrm{r},i}^n \tag{11.29}
$$

all $i \in \Theta^n$ and $i_2 \in \Theta^m$ with $m = \mathrm{pre}(n)$. If the ellipsoidal over-approximation $\mathcal{E}(\acute{\mathscr{E}}_{i_2}^m)$ is used, the input set can be approximated by the Minkowski sum of two ellipsoids. The linear input constraints $H_{\mathrm{u}}^n u^n[k] \leq h_{\mathrm{u}}^n$ can be formulated row-wise using the support functions of the ellipsoidal Minkowski sum over-approximation (cf. Corollary 2.1):

$$
\left[H_{\mathrm{u},l_{\mathrm{u}}}^n\left(\left(\varepsilon_{i,i_2}^n\right)^{-1}\left(\mathscr{Y}_i^n[k](\mathscr{G}_i^n[k])^{-1}\right)\mathscr{Q}_i^n\left(\mathscr{Y}_i^n[k](\mathscr{G}_i^n[k])^{-1}\right)^{\top} + \cdots \right.\tag{11.30}
$$

$$
\left.\cdots + (1-\varepsilon_{i,i_2}^n)^{-1}\hat{\mathscr{Y}}_i^n\left(\hat{\mathscr{G}}_i^n\right)^{-1}\acute{\mathscr{E}}_{i_2}^m\left(\hat{\mathscr{Y}}_i^n\left(\hat{\mathscr{G}}_i^n\right)^{-1}\right)^{\top}\right)\left(H_{\mathrm{u},l_{\mathrm{u}}}^n\right)^{\top}\right]^{1/2} \leq h_{\mathrm{u},l_{\mathrm{u}}}^n - H_{\mathrm{u},l_{\mathrm{u}}}^n u_{\mathrm{r},i}^n
$$

for all $i \in \Theta^n$, $i_2 \in \Theta^m$, $l_{\mathrm{u}} \in \mathbb{N}_{n_{\mathrm{H},\mathrm{u}}^n}$, and $\acute{\varepsilon}_{i,i_2}^n \in (0,1)$. The constraints (11.30) can be formulated as time-dependent BMIs by applying the Schur complement, a congruence transformation with $\mathscr{T} = \mathrm{diag}(I, \mathscr{G}_i^n, \hat{\mathscr{G}}_i^n)$, and the approximations $(\mathscr{G}_i^n[k])^{\top}(\mathscr{Q}_i^n)^{-1}\mathscr{G}_i^n[k] \geq (\mathscr{G}_i^n[k])^{\top} + \mathscr{G}_i^n[k] - \mathscr{Q}_i^n$ and $(\hat{\mathscr{G}}_i^n)^{\top}(\acute{\mathscr{E}}_{i_2}^n)^{-1}\hat{\mathscr{G}}_i^n \geq (\hat{\mathscr{G}}_i^n)^{\top} + \hat{\mathscr{G}}_i^n - \acute{\mathscr{E}}_{i_2}^n$ (cf. Lemmata 2.3 - 2.5). According to Lemma 7.3, these BMIs hold if the corresponding BMIs hold for all vertices of the matrix polytopes $\mathbb{A}_i^n$ and $\mathbb{B}_i^n$. Hence, the

constraints (11.30) are satisfied if the following BMIs:

$$
\begin{bmatrix}
\left(h^n_{\mathrm{u},l_{\mathrm{u}}} - H^n_{\mathrm{u},l_{\mathrm{u}}}\, u^n_{\mathrm{r},i} \right)^2 & H^n_{\mathrm{u},l_{\mathrm{u}}}\, \mathscr{Y}^n_{i,[l_{\mathrm{A}}]} & H^n_{\mathrm{u},l_{\mathrm{u}}}\, \hat{\mathscr{Y}}^n_i \\
\star & \varepsilon^n_{i,i_2}\left(\left(\mathscr{G}^n_{i,[l_{\mathrm{B}}]} \right)^\top + \mathscr{G}^n_{i,[l_{\mathrm{B}}]} - \mathscr{Q}^n_i \right) & \mathbf{0} \\
\star & \star & (1 - \varepsilon^n_{i,i_2})\left(\left(\hat{\mathscr{G}}^n_i \right)^\top + \hat{\mathscr{G}}^n_i - \hat{\mathscr{E}}^n_{i_2} \right)
\end{bmatrix} \geq 0 \quad (11.31)
$$

hold for all $i \in \Theta^n$, $i_2 \in \Theta^m$, $l_{\mathrm{u}} \in \mathbb{N}_{n^n_{\mathrm{h,u}}}$, $l_{\mathrm{A}} \in \mathbb{N}_{n^n_{\mathrm{A},i}}$, and $l_{\mathrm{B}} \in \mathbb{N}_{n^n_{\mathrm{B},i}}$.

Complete SDP: Combining these results with the invariance conditions and state constraints, that have been derived for the decentralized setup in the previous section, the following SDP formulation results:

Theorem 11.3. *Let $\mathscr{Q}^n, \mathscr{G}^n, \mathscr{Y}^n, \hat{\mathscr{G}}^n$, and $\hat{\mathscr{Y}}^n$ be a solution of the SDP:*

$$
\max_{\substack{\mathscr{Q}^n,\mathscr{G}^n,\mathscr{Y}^n,\hat{\mathscr{G}}^n,\hat{\mathscr{Y}}^n,\varepsilon^n \\ \hat{\mathscr{E}}^n,\alpha^n,\acute{\alpha}^n,\varepsilon^n,\acute{\varepsilon}^n,\varrho^n}} \sum_{n=1}^{n_{\mathrm{s}}} \sum_{i=1}^{n^n_\theta} \log\,\det(\mathscr{Q}^n_i), \tag{11.32a}
$$

$$
\text{s.\,t.} \quad \alpha^n_{i,i_2},\ \acute{\alpha}^n_{i_2},\ \varepsilon^n_i,\ \acute{\varepsilon}^n_{i,i_2},\ \varrho^n_{i,i_2} \in (0,1), \tag{11.32b}
$$

$$
(11.11), (11.16), (11.27), (11.28), (11.31) \tag{11.32c}
$$

for all $n \in \mathbb{N}_{n_{\mathrm{s}}}$ and all indices $i, i_2, l_{\mathrm{x}}, l_{\mathrm{u}}, l_{\mathrm{A}}, l_{\mathrm{B,1}}, l_{\mathrm{B,2}}, l_{\mathrm{G}}$ according to the subsystem dimensions. Then, the ellipsoids $\mathcal{E}(\mathscr{Q}^n_i)$ define RCIS that consider both local disturbances and couplings. The distributed controllers (11.22) hold the states of the coupled JMS robustly in the RCIS and satisfy the constraints.

Proof. The proof follows from Theorem 11.1 for the decentralized case and by construction of the constraints as described in this section. $\qquad\square$

As for the decentralized case, the SDP (11.32) is a BMI problem that can be solved by PENBMI. Due to the formulation of the disturbance feedback, the problem formulation is less conservative and results in larger RCIS. This is demonstrated by simulations in Section 11.3. However, the problem formulation contains more BMI constraints than the optimization problem (11.17) for the decentralized architecture. Hence, this problem formulation is only suitable for small to medium scale problems.

11.2.2. Non-Cooperative Distributed MPC

The design and implementation of the local MPCs is similar to the decentralized case presented in Section 11.1.2 (cf. Fig. 11.1), with the following differences: The determination of the invariance constraints, output set constraints, as well as the state constraint tightening is performed with the local error sets $\mathcal{E}(E^n_i)$, instead of the overall error sets $\mathcal{E}(\mathscr{E}^n_i)$. This is possible, since the local MPCs receive the expected values $\bar{w}^n_{\mathrm{p},\delta}[k]$ of the couplings from the predecessors. Furthermore, for

the cost calculation, the predicted output trajectory $\bar{\mathbf{y}}^m[k]$ transmitted from the predecessor is used to determine the expected disturbances ($\bar{\mathbf{w}}^n_{\mathrm{p}}[k] = \bar{\mathbf{y}}^m_{\mathrm{p}}[k]$). All in all, only the construction of the shape matrices $\breve{\mathscr{Q}}^n_{i_2,i}$ and $\breve{\mathscr{Q}}^n_{\mathrm{y},i}$ and the constraint tightening have to be adapted. Thus, the resulting formulation of the local optimization problems is the same as presented in Theorem 11.2 for the decentralized case. The RCIS are larger and more accurate disturbance predictions are used. Hence, the calculation time is similar to that of the decentralized MPC formulation. The control performance is expected to increase due to more accurate information for the state prediction and a larger feasible set.

Remark 11.1. *In order to determine the values* $\bar{\mathbf{w}}^n_{\mathrm{p}}[k]$ *employing* $\bar{\mathbf{y}}^m[k]$, *the prediction horizon of a local MPC should be at least the same length as the prediction horizons of its successors' MPCs:*

$$N^m \geq N^n \qquad \forall n, m \in \mathbb{N}^{n_s} \mid n = \mathrm{succ}(m, k_b). \tag{11.33}$$

Otherwise, estimates for the missing values $\bar{w}^n_{\mathrm{p},\delta}[j]$ *are needed.*

11.2.3. Cooperative Distributed MPC

In a cooperative MPC architecture, the local MPCs minimize a global cost criterion $\hat{J}$. In this thesis, a weighted sum of the local cost criteria J^n is used (cf. [109]):

$$\hat{J} := \sum_{n=1}^{n_s} \gamma^n J^n, \qquad \gamma^n > 0. \tag{11.34}$$

Since the local MPCs can only influence their own subsystem and all successors until the end of the production process directly, the local representations of the global cost criterion can be reduced to:

$$\hat{J}^n := \sum_{m \,\in\, n \,\cup\, \mathrm{Succ}(n,k_b)} \gamma^m J^m, \qquad \gamma^m > 0. \tag{11.35}$$

The local MPC for subsystem $n \in \mathbb{N}_{n_s}$ can only influence costs of the succeeding subsystems that are related to the products processed in subsystem n during the prediction horizon N^n. Due to the processing times, it may take several production cycles until these products are processed in one of the downstream production units. This issue is illustrated for the considered hot stamping line in Fig. 13.1. Hence, the corresponding costs of the downstream subsystems are not necessarily the next N^n step costs of these subsystems. Due to the lag $l^{n,m}$ until the products are processed by subsystem $m \in \mathrm{Succ}(n, k_b)$, the step costs from time $k^m + l^{n,m}$ to $k^m + l^{n,m} + N^n$ are influenced by the MPC in subsystem n. To predict these costs, the corresponding input and disturbance trajectories $\mathbf{u}^m[k^m + l^{n,m}]$ and $\bar{\mathbf{w}}^m_{\mathrm{d}}[k^m + l^{n,m}]$, as well as the corresponding states $x^m[k^m + l^{n,m}]$ and $\theta^m_{k^m + l^{n,m}}$ have to be known. Since these times are set in the future, predictions of these quantities provided by the

local MPC at time k^n are employed. Hence, the correct indexing would be, e.g., $x^m[k^m + l^{n,m}|k^n]$. To ease the notation, a predicted quantity of subsystem m related to a product processed currently in subsystem n at time k^n is denoted by the bold faced subsystem index $\boldsymbol{n}$ in double brackets, e.g., $x^m[\![\boldsymbol{n}]\!] \coloneqq x^m[k^m + l^{n,m}|k^n]$.

This cost prediction setup has a direct impact on the selection of the prediction horizon length. In order to provide all trajectories needed for the prediction of the costs caused in the downstream subsystems, the prediction horizons of the downstream subsystems have to be chosen such that:

$$N^m \geq N^n + l^{n,m} \qquad \forall \; m \in \text{Succ}(n, k_b). \tag{11.36}$$

This condition reveals a problem concerning the prediction horizon length. According to Remark 11.1 on page 156, the prediction horizon of a local MPC has to be equal or shorter than the prediction horizon of the MPC of the direct predecessor. Both conditions are satisfied only if all $l^{n,m} = 0$ and if all prediction horizons have the same length. However, due to the processing times in real production systems, often $l^{n,m} > 0$ applies for some m, n. In these cases, parts of the trajectories needed for cost prediction or for the prediction of the couplings are not available to some subsystems. Either these quantities have to be approximated, e.g., by assuming that the predicted trajectories are constant for the last time steps, or the step costs of subsystems, for which no information is available, have to be neglected. In order to minimize the negative effects of this problem, the prediction horizon should be large compared to the lags $l^{n,m}$.

For the remainder of this chapter, it is assumed that the required trajectories are available or are approximated appropriately. Then, the local representations of the global cost function $\hat{J}^n$ can be formulated as quadratic functions. This is presented for both cost definitions (CoE and EoC) in the following sections.

Costs of the Expected Value

This section demonstrates how the local representation of the global cost criterion $\hat{J}^n$ can be formulated as a quadratic function of the following form:

$$\hat{J}^n = \left\| \boldsymbol{u}^n[k] \right\|^2_{W^n[k]} + q^n[k]\boldsymbol{u}^n[k] + \Psi[k], \tag{11.37}$$

employing the CoE definition (cf. Section 6.1). All cost components that are independent of the input trajectory $\boldsymbol{u}^n[k]$ are summarized in $\Psi[k]$. These terms are neglected, since they do not influence the optimal input trajectory.

Employing the approaches for calculating the prediction matrices $\boldsymbol{A}[k]$, $\boldsymbol{B}[k]$, and $\boldsymbol{G}[k]$ (see Equation (6.21) on page 85) as well as the coupling definition (10.2), prediction equations for the expected values of the predicted state trajectories $\bar{\boldsymbol{x}}^m[\![\boldsymbol{n}]\!]$ can be derived. The corresponding costs can be transformed into the quadratic form (11.37), as shown in the following theorem:

Theorem 11.4. *Let the matrices for the prediction of the expected value of the states $\bar{x}^m[n]$ of subsystem $m \in \mathrm{Succ}(n, k_{\mathrm{b}})$ employed by subsystem $n \in \mathbb{N}_{n_s}$ be denoted by $\boldsymbol{A}^m[k^n]$, $\boldsymbol{B}^m[k^n]$, and $\boldsymbol{G}^m[k^n]$. These matrices are determined according to (6.21) and Theorem 6.1 based on the linearized subsystem (8.5) with $\xi_{\mathrm{l}}^m[n] = \left[x^m[n] \quad u^m[n] \quad \bar{w}^m[n] \right]$. Then, the local representation of the global cost function $\hat{J}^n$ can be formulated by (11.37) with:*

$$W^n[k] \quad = \quad \gamma^n \left(W^{n,n}[k^n] + \boldsymbol{R}^n \right) \quad + \sum_{m \,\in\, \mathrm{Succ}(n, k_{\mathrm{b}})} \gamma^m \, W^{n,m}[k^n], \quad (11.38)$$

$$q^n[k] \quad = \quad 2\gamma^n \left(q^{n,n}[k^n] - \left(\bar{\boldsymbol{u}}_{\mathrm{r}}^n[k^n] \right)^\top \boldsymbol{R}^n \right) \quad + \sum_{m \,\in\, \mathrm{Succ}(n, k_{\mathrm{b}})} 2\gamma^m \, q^{n,m}[k^n], \quad (11.39)$$

where $\boldsymbol{R}^n = \mathrm{diag}\left(R^n, \cdots, R^n \right)$ and $\bar{\boldsymbol{u}}_{\mathrm{r}}^n[k^n]$ is determined according to (6.18). The matrices $W^{n,m}[k^n]$ and $q^{n,m}[k^n]$ describe the costs related to subsystem m that can be influenced by $\boldsymbol{u}^n[k]$. These matrices are calculated by the following recursion:

Initialization*: Set $m_1 = n$ and calculate:*

$$\boldsymbol{f}^{n,n}[k^n] = \boldsymbol{A}^n[k^n]\, x^n[k^n] + \boldsymbol{G}^n[k^n]\bar{\boldsymbol{w}}^n[k^n], \tag{11.40}$$
$$\boldsymbol{B}^{n,n}[k^n] = \boldsymbol{B}^n[k^n]. \tag{11.41}$$

Recursion*: Set $m_2 = \mathrm{succ}(m_1, k_{\mathrm{b}})$ and calculate:*

$$\boldsymbol{f}^{n,m_2}[k^n] = \boldsymbol{A}^{m_2}[k^n]\, x^{m_2}[n] + \boldsymbol{B}^{m_2}[k^n]\, \boldsymbol{u}^{m_2}[n] + \boldsymbol{G}^{m_2}[k^n] \begin{bmatrix} \boldsymbol{T}_{\mathrm{y}}^{m_1} \boldsymbol{f}^{n,m_1}[k^n] \\ \bar{\boldsymbol{w}}_{\mathrm{d}}^{m_2}[n] \end{bmatrix}, \quad (11.42)$$

$$\boldsymbol{B}^{n,m_2}[k^n] = \boldsymbol{G}_{\mathrm{p}}^{m_2}[k^n]\, \boldsymbol{T}_{\mathrm{y}}^{m_1} \boldsymbol{B}^{n,m_1}[k^n] \tag{11.43}$$
$$m_1 = m_2 \tag{11.44}$$

recursively for all subsystems. Here, $\boldsymbol{G}_{\mathrm{p}}^{m_2}[k^n]$ represents a matrix containing only the columns of $\boldsymbol{G}^{m_2}[k^n]$, which correspond to $\bar{\boldsymbol{w}}_{\mathrm{p}}^{m_2}[k^n]$.

Calculation of the Cost Prediction Matrices*:*

$$W^{n,m}[k^n] = \left(\boldsymbol{B}^{n,m}[k^n] \right)^\top \boldsymbol{Q}^m \boldsymbol{B}^{n,m}[k^n] \qquad \forall \; m \,\in\, n \cup \mathrm{Succ}(n, k_{\mathrm{b}}) \quad (11.45)$$
$$q^{n,m}[k^n] = \left(\boldsymbol{f}^{n,m}[k^n] - \boldsymbol{x}_{\mathrm{r}}^m[n] \right)^\top \boldsymbol{Q}^m \boldsymbol{B}^{n,m}[k^n] \quad \forall \; m \,\in\, n \cup \mathrm{Succ}(n, k_{\mathrm{b}}). \quad (11.46)$$

Proof. See Appendix A.2. □

This theorem demonstrates that the costs in the succeeding subsystems that are influenced by the local MPC in subsystem n can be also formulated as a quadratic term. To this end, the dynamics have to be linearized and the prediction matrices $\boldsymbol{A}[k]$, $\boldsymbol{B}[k]$, and $\boldsymbol{G}[k]$ have to be determined for all subsystems $m \in n \cup \mathrm{Succ}(n, k_{\mathrm{b}})$. Finally, the influence of the local inputs on the costs of all successors is calculated by the recursions described in Theorem 11.4. Hence, the computational effort for the calculation of the cost matrices $W^n[k]$ and $q^n[k]$ is significantly higher than for the

non-cooperative setup. However, due to the efficient calculation of the prediction matrices presented in Theorem 6.1, the overall computation effort is relatively low.

If a production system with multiple direct successors of one subsystems is considered, the theorem can be adapted in a straightforward manner, by including all successors in the recursion.

Expected Value of the Costs

Employing the expected value of the costs (EoC) definition according to Equation (5.9) on page 78, the local representation of the global cost function $\hat{J}^n$ can be stated as follows:

$$\sum_{m \in n \cup \mathrm{Succ}(n,k_\mathrm{b})} \gamma^m \, \mathrm{E} \left(\|\boldsymbol{x}^m[\![n]\!] - \boldsymbol{x}_\mathrm{r}^m[\![n]\!]\|_{\boldsymbol{Q}^m(\theta_{k,N}^m)}^2 + \|\boldsymbol{u}^m[\![n]\!] - \boldsymbol{u}_\mathrm{r}^m[\![n]\!]\|_{\boldsymbol{R}^m(\theta_{k,N}^m)}^2 \right). \quad (11.47)$$

Since the inputs of all downstream subsystems cannot be influenced by $\boldsymbol{u}^n[k]$, the input costs are only evaluated for subsystem n. Provided that the predicted input, state, and disturbance trajectories $\boldsymbol{u}^m[\![n]\!]$, $\boldsymbol{x}^m[\![n]\!]$, $\boldsymbol{w}_\mathrm{d}^m[\![n]\!]$, and $\boldsymbol{w}_\mathrm{p}^m[\![n]\!]$ as well as the state references $\boldsymbol{x}_\mathrm{r}^m[\![n]\!]$ are known for all downstream subsystems m, the expected value of the state costs in (11.47) can be predicted for all downstream subsystems separately as for single JMS. To this end, the prediction matrices $\Phi_{(\cdot,\cdot)}^m[\![j]\!]$ are determined for all downstream subsystems according to the Lemmata 6.2 and 6.3 using the linearized systems (8.5). For all downstream subsystems, only costs depending on $\bar{\boldsymbol{w}}_\mathrm{p}^m[\![n]\!]$ have to be considered, since only $\bar{\boldsymbol{w}}_\mathrm{p}^m[\![n]\!]$ can be influenced directly by $\boldsymbol{u}^n[k]$. To calculate these cost terms, the corresponding cost prediction matrices $\Phi_{\mathrm{w_p}(\cdot)}^m[\![j]\!]$ and $\Phi_{(\cdot)\mathrm{w_p}}^m[\![j]\!]$ are needed. These block matrices can be extracted directly from $\Phi_{\mathrm{w}(\cdot)}^m[\![j]\!]$ and $\Phi_{(\cdot)\mathrm{w}}^m[\![j]\!]$ presented in the Lemmata 6.2 and 6.3. Employing the facts that the trajectories $\boldsymbol{u}^m[\![n]\!]$, $\boldsymbol{x}^m[\![n]\!]$, $\boldsymbol{w}_\mathrm{d}^m[\![n]\!]$, and $\boldsymbol{x}_\mathrm{r}^m[\![n]\!]$ are transmitted by the MPCs of the downstream subsystems (cf. Fig. 11.3) and that $\bar{\boldsymbol{w}}_\mathrm{p}^m[\![n]\!]$ can be predicted with the results of Theorem 11.4, the following result holds:

Theorem 11.5. *Let the matrices $\Phi_{(\cdot,\cdot)}^m[\![j]\!]$ be determined according to Lemmata 6.1, 6.2, and 6.3 for all subsystems $m \in n \cup \mathrm{Succ}(n,k_\mathrm{b})$, based on the linearized systems (8.5) with $\xi_\mathrm{l}^m[\![n]\!] = \begin{bmatrix} x^m[\![n]\!] & u^m[\![n]\!] & \bar{w}^m[\![n]\!] \end{bmatrix}$. In addition, let $\boldsymbol{B}^{n,m}[k^n]$ and $\boldsymbol{f}^{n,m}[k^n]$ be defined according to Theorem 11.4. Then, the local representation of the global cost function $\hat{J}^n$ in (11.47) can be formulated as a quadratic function (11.37) with:*

$$W^n[k] = \gamma^n \left(\sum_{j=1}^{N^n} \Phi_{\mathrm{uu}}^n[\![j]\!] + \boldsymbol{R}^n[k^n] \right) + \sum_{m \in \mathrm{Succ}(n,k_\mathrm{b})} \gamma^m \, W^{n,m}[k^n], \quad (11.48)$$

$$q^n[k] = 2\gamma^n \left(\sum_{j=1}^{N^n} \left[(x^n[k^n])^\top \Phi_{\mathrm{xu}}^n[\![j]\!] + (\bar{\boldsymbol{w}}^n[k^n])^\top \Phi_{\mathrm{wu}}^n[\![j]\!] - \Phi_{\mathrm{x_r u}}^n[\![j]\!] \right] - \Phi_{\mathrm{u_r u}}^n[k^n] \right) \quad (11.49)$$

$$+ \sum_{m \in \mathrm{Succ}(n,k_\mathrm{b})} 2\gamma^m \, q^{n,m}[k^n], \quad (11.50)$$

where $\boldsymbol{R}^{\boldsymbol{n}}[k^n]$ is defined according to (6.31) on page 87. The matrices $W^{\boldsymbol{n},\boldsymbol{m}}[k^n]$ and $q^{\boldsymbol{n},\boldsymbol{m}}[k^n]$ describe the costs related to subsystem m that can be influenced by $\boldsymbol{u}^{\boldsymbol{n}}[k]$. These equations hold, if $W^{\boldsymbol{n},\boldsymbol{m}}[k^n]$ and $q^{\boldsymbol{n},\boldsymbol{m}}[k^n]$ are calculated as follows:

$$W^{\boldsymbol{n},\boldsymbol{m}}[k^n] = \left(\boldsymbol{T}_{\mathrm{y}}^{\boldsymbol{m}_1}\,\boldsymbol{B}^{\boldsymbol{n},\boldsymbol{m}_1}[k^n]\right)^{\top}\left(\sum_{j=1}^{N^m}\Phi_{\mathrm{w_p w_p}}^{m}[j]\right)\boldsymbol{T}_{\mathrm{y}}^{\boldsymbol{m}_1}\,\boldsymbol{B}^{\boldsymbol{n},\boldsymbol{m}_1}[k^n], \tag{11.51}$$

$$q^{\boldsymbol{n},\boldsymbol{m}}[k^n] = \sum_{j=1}^{N^m}\Big[\left(x^m[\boldsymbol{n}]\right)^{\top}\Phi_{\mathrm{x w_p}}^{m}[j] + \left(\bar{\boldsymbol{w}}_{\mathrm{d}}^{\boldsymbol{m}}[\boldsymbol{n}]\right)^{\top}\Phi_{\mathrm{w_d w_p}}^{m}[j] + \left(\boldsymbol{u}^{\boldsymbol{m}}[\boldsymbol{n}]\right)^{\top}\Phi_{\mathrm{u w_p}}^{m}[j] - \ldots$$

$$\ldots - \Phi_{\mathrm{x_r w_p}}^{m}[j] + \left(\boldsymbol{T}_{\mathrm{y}}^{\boldsymbol{m}_1}[k^n]\,\boldsymbol{f}^{\boldsymbol{n},\boldsymbol{m}_1}[k^n]\right)^{\top}\Phi_{\mathrm{w_p w_p}}^{m}[j]\Big]\boldsymbol{T}_{\mathrm{y}}^{\boldsymbol{m}_1}\,\boldsymbol{B}^{\boldsymbol{n},\boldsymbol{m}_1}[k^n] \tag{11.52}$$

for all $m \in \mathrm{Succ}(n, k_{\mathrm{b}})$ with $m_1 = \mathrm{pre}(m)$.

Proof. See Appendix A.3. □

This theorem provides an algorithm to formulate the global cost function as a quadratic function. Hence, the local optimization problems can still be formulated as QCQPs. To formulate the cost function, the prediction matrices $\Phi_{(\cdot,\cdot)}^{m}[j]$ have to be determined for the considered subsystem and all downstream subsystems. In addition, the procedure described in Theorem 11.4 has to be performed to determine $\boldsymbol{B}^{\boldsymbol{n},\boldsymbol{m}_1}[k^n]$ and $\boldsymbol{f}^{\boldsymbol{n},\boldsymbol{m}_1}[k^n]$. Hence, the computational effort to calculate $W^{\boldsymbol{n}}[k]$ and $q^{\boldsymbol{n}}[k]$ is significantly larger than for the non-cooperative DMPC formulation or a cooperative DMPC employing the CoE formulation. However, the overall computation time is still low, due to the efficient calculation of the prediction matrices. The whole DMPC procedure is presented in the following section.

Complete DMPC Procedure

The DMPC design and implementation is divided into the RCIS as well as constraint design performed offline and the online execution of the MPC.

Offline Part: The RCIS are determined according to Theorem 11.3. Analogously to the formulation of a non-cooperative DMPC in Section 11.2.2, the RCIS, output sets, and state constraints are tightened employing the local error sets $\mathcal{E}\left(E_i^{\boldsymbol{n}}\right)$, in order to derive robust constraints[2]:

- To guarantee that the system stays in the RCIS, ellipsoidal inner approximations $\mathcal{E}\left(\breve{\mathscr{Q}}_{i,i_2}^{\boldsymbol{n}}\right) \subseteq \mathcal{E}\left(\mathscr{Q}_{i_2}^{\boldsymbol{n}}\right) \ominus \mathcal{E}\left(E_i^{\boldsymbol{n}}\right)$ of the Minkowski differences of the RCIS and the local error set are determined for all $n \in \mathbb{N}_{n_{\mathrm{s}}}$ and $i, i_2 \in \Theta^{\boldsymbol{n}}$.

- To guarantee that the outputs take values that have been used for the RCIS design, ellipsoidal inner approximations $\mathcal{E}\left(\breve{\mathscr{Q}}_{\mathrm{y},i}^{\boldsymbol{n}}\right) \subseteq \mathcal{E}\left(\acute{\mathscr{E}}_i^{\boldsymbol{n}}\right) \ominus T_{\mathrm{y}}^{\boldsymbol{n}}\mathcal{E}\left(E_i^{\boldsymbol{n}}\right)$ of the Minkowski difference of the output set and the local error sets are determined for all $n \in \mathbb{N}_{n_{\mathrm{s}}}$ and $i \in \Theta^{\boldsymbol{n}}$.

---

[2]It is sufficient to use the local error sets $\mathcal{E}(E_i^{\boldsymbol{n}})$ for the constraint tightening, since the local MPCs know the values $\bar{w}_{\mathrm{p}}^{\boldsymbol{n}}[k]$ of the couplings.

- The state constraints are tightened with the support functions of the local error set $\mathcal{E}(E_i^n)$ for all $n \in \mathbb{N}_{n_s}$ and $i \in \Theta^n$, see Equation (11.53c).

The shape matrices $\breve{\mathcal{Q}}_{i,i_2}^n$ and $\breve{\mathcal{Q}}_{y,i}^n$ can be determined according to Lemma 2.2 such that the volume of the corresponding ellipsoid is maximized.

Online Part: In all subsystems $n \in \mathbb{N}_{n_s}$, the following procedure is executed in each production cycle / sampling interval:

0. Receive the predicted trajectories $\boldsymbol{u^m}[\![0]\!]$, $\boldsymbol{x^m}[\![0]\!]$, $\boldsymbol{w}_d^m[\![0]\!]$, $\boldsymbol{x}_r^m[\![0]\!]$, and the current Markov state θ_k^m from all downstream subsystems (cf. Fig. 10.2). Due to the asynchronous sampling of the subsystems, this information can be received at any time during the production cycle and at different times from the different subsystems.

1. As soon as a new product is delivered to the n-th subsystem: Receive the predicted output trajectory $\bar{\boldsymbol{y}}^m[\![0]\!]$ of the direct predecessor (which delivered the product to be processed). Measure the current Markov state θ_k^n and the continuous state $x^n[k]$. Determine the expected values of the disturbances $\bar{\boldsymbol{w}}^n[\![0]\!]$ from the predicted product properties ($\bar{\boldsymbol{w}}_p^n[\![0]\!] = \bar{\boldsymbol{y}}^m[\![0]\!]$), historical data, and the disturbance models $\breve{\nu}^n$.

2. Linearize the JMS $\mathcal{S}^n$ in $\xi_l^n[k] = \begin{bmatrix} x^n[k] & u^n[k|k-1] & \bar{w}^n[k] \end{bmatrix}$ and linearize the JMS $\mathcal{S}^m$ of all downstream subsystems in $\xi_l^m[\![n]\!] = \begin{bmatrix} x^m[\![n]\!] & u^m[\![n]\!] & \bar{w}^m[\![n]\!] \end{bmatrix}$.

3. Determine the matrices $W^n[k]$ and $q^n[k]$ according to Theorem 11.4 (for CoE) or Theorem 11.5 (for EoC) employing the linearized systems.

4. Solve the following QCQP:

$$\min_{\boldsymbol{u^n}[k]} \|\boldsymbol{u^n}[k]\|_{W^n[k]}^2 + q^n[k]\,\boldsymbol{u^n}[k] \tag{11.53a}$$

$$\text{s.t. } \boldsymbol{H}_u^n \boldsymbol{u^n}[k] \le \boldsymbol{h}_u^n[k], \tag{11.53b}$$

$$H_{x,l_x}^n \left(A_{\theta_k}^n[k]\,x^n[k] + B_{\theta_k}^n[k]\,u^n[\![0]\!] + G_{\theta_k}'^n[k]\,\bar{w}'^n[k] \right) \le h_{x,l_x}^n[k] - \|H_{x,l_x}^n\|_{E_{\theta_k}^n}, \tag{11.53c}$$

$$\left\| A_{\theta_k}^n[k]\,x^n[k] + B_{\theta_k}^n[k]\,u^n[\![0]\!] + G_{\theta_k}'^n[k]\,\bar{w}'^n[k] - x_r^n \right\|_{\left(\breve{\mathcal{Q}}_{\theta_k,i}^n \right)^{-1}}^2 \le 1, \tag{11.53d}$$

$$\left\| T_y^n \left[A_{\theta_k}^n[k]\,x^n[k] + B_{\theta_k}^n[k]\,u^n[\![0]\!] + G_{\theta_k}'^n[k]\,\bar{w}'^n[k] - x_r^n \right] \right\|_{\left(\breve{\mathcal{Q}}_{y,\theta_k}^n \right)^{-1}}^2 \le 1, \tag{11.53e}$$

where $l_x \in \mathbb{N}_{n_{h,x}^n}$ and $i \in \Theta^n$ with $p_{\theta_k,i}^n > 0$.

5. Apply $u^n[\![0]\!]$ to the JMS and transmit the trajectories $\boldsymbol{u^n}[\![0]\!]$, $\boldsymbol{x^n}[\![0]\!]$, $\boldsymbol{w}_d^n[\![0]\!]$, $\boldsymbol{x}_r^n[\![0]\!]$, and the current Markov state θ_k^n to all upstream subsystems. Transmit the predicted output trajectory $\bar{\boldsymbol{y}}^n[\![0]\!]$ to the succeeding subsystem.

6. Wait for the next product and go to *1.*

Theorem 11.6. *Let a solution of the SDP* (11.32) *exist, and the matrices* $\breve{\mathscr{Q}}^n_{i,i_2}$ *and* $\breve{\mathscr{Q}}^n_{y,i}$ *are determined as described in the offline part. Finally, suppose that for all subsystems* $n \in \mathbb{N}_{n_s}$ *an initial solution of the optimization problem* (11.53) *exists and that the optimization problem can be solved fast enough to apply* $u^n[0]$. *Then, the states of all subsystems are robustly kept in the RCIS if all subsystems implement the DMPC procedure presented above. The optimization problem* (11.53) *is recursively feasible and the constraints are satisfied for all subsystems.*

Proof. The constraints are robustly satisfied, due to (11.53b) and (11.53c). Robustness is guaranteed by the constraint tightening with the error sets $\mathcal{E}(E^n_i)$. The constraints (11.53d) and (11.53e) robustly keep the subsystems in the RCIS and ensure that the outputs are confined to the ellipsoids $\mathcal{E}(\acute{\mathscr{E}}^n_{\theta_k})$. Thus, according to Theorem 11.3, $u'[k] = K^n_{\theta_k}[k]x^n_\delta[k] + \hat{K}^n_{\theta_k}\bar{w}^n_{p,\delta}[k] + u^n_{r,\theta_k}$ provides a feasible solution of the optimization problem (11.53), and recursive feasibility is guaranteed. In addition, the states are bounded, and the JMS is stable. $\qquad\square$

The dimension of the resulting QCQP is the same as for the decentralized case. Hence, the time for solving (11.53) is expected to be in the same order as for the decentralized case. However, the determination of the cost matrices $W^n[k]$ and $q^n[k]$ is computationally more demanding, due to the cost calculation of the downstream production units. The computational effort for computing $W^n[k]$ and $q^n[k]$ depends on the number of downstream processing steps and their system dimensions.

The presented DMPC establishes a control approach for the distributed production system. Its properties are investigated with artificial systems in the following section and with the models of a hot stamping line in the Chapter 13.

Remark 11.2. *The DMPC scheme proposed can also be employed for JMLS. In this case, no polytopic system over-approximation is needed. Hence, less effort is needed for the computation of the RCIS, since the number of constraints in the SDP* (11.32) *reduces significantly. For the online implementation, the linearization (step 2.) is not needed. All other steps are applied without changes.*

11.3. Simulation Study

This section illustrates the properties of the decentralized and distributed MPC architectures with simulations. A small distributed test system consisting of two subsystems with two Markov states each is employed. The dynamics of both subsystems are adopted from the nonlinear sample system introduced in Section 8.2:

$$f^1_1(\xi^1[k]) = \begin{bmatrix} 0.2(x^1_1[k])^3 + 0.2x^1_2[k] + 1.4\operatorname{atan}\frac{u^1[k]}{8} + 0.2w^1_d[k] \\ 0.5x^1_1[k] + 0.3(x^1_2[k])^2 + 1.8\operatorname{atan}\frac{u^1[k]}{8} + 0.3\operatorname{atan}w^1_d[k] \end{bmatrix},$$

$$f^1_2(\xi^1[k]) = \begin{bmatrix} 1.05x^1_1[k]\cdot\mathrm{e}^{-0.05x^1_2[k]} - 0.3x^1_2[k] + 2.2\operatorname{atan}\frac{u^1[k]}{8} - 0.2w^1_d[k] \\ 0.5(x^1_1[k])^4 + 0.5x^1_2[k] + \operatorname{atan}\frac{u^1[k]}{8} + 0.2\sin w^1_d[k] \end{bmatrix},$$

$$w_\mathrm{p}^2[k] = y^1[k] = x_2^1[1], \tag{11.54}$$

$$f_1^2(\xi^2[k]) = \begin{bmatrix} 0.2(x_1^2[k])^3 + 0.2x_2^2[k] + 1.4\operatorname{atan}\frac{u^2[k]}{8} + 0.2(w_\mathrm{d}^2[k] + w_\mathrm{p}^2[k]) \\ 0.5x_1^2[k] + 0.3(x_2^2[k])^2 + 1.8\operatorname{atan}\frac{u^2[k]}{8} + 0.3\operatorname{atan}(w_\mathrm{d}^2[k] + w_\mathrm{p}^2[k]) \end{bmatrix},$$

$$f_2^2(\xi^2[k]) = \begin{bmatrix} 1.05x_1^2[k]\cdot\mathrm{e}^{-0.05x_2^2[k]} - 0.3x_2^2[k] + 2.2\operatorname{atan}\frac{u^2[k]}{8} - 0.2(w_\mathrm{d}^2[k] + w_\mathrm{p}^2[k]) \\ 0.5(x_1^2[k])^4 + 0.5x_2^2[k] + \operatorname{atan}\frac{u^2[k]}{8} + 0.2\sin\left(w_\mathrm{d}^2[k] + w_\mathrm{p}^2[k]\right) \end{bmatrix},$$

$$\mathcal{M}^1 = \mathcal{M}^2 = \left(\{1,2\}, \begin{bmatrix} 0.9 & 0.1 \\ 0.55 & 0.45 \end{bmatrix}, \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}\right).$$

For both subsystems, input constraints $|u^n[k]| \le 4$ and state constraints $[-1.2\ -2]^\top \le x^n[k] \le [1.2\ 2]^\top$ are considered. Furthermore, the local disturbances are governed by a truncated Gaussian noise process with $w_\mathrm{d}^n[k] \sim \mathcal{N}(0, 0.1)$ and $|w_\mathrm{d}^n[k]| \le 0.3$.

RCIS Design For the system (11.54), the RCIS are calculated for a decentralized setup according to Theorem 11.1 and for a distributed setup according to Theorem 11.3. The matrix polytopes $\mathbb{A}_i^n$ and $\mathbb{B}_i^n$ for $n \in \{1,2\}$ and $i \in \{1,2\}$ are the same as for the single JMS (see page 125). The matrix polytopes $\mathbb{G}_i^2$ with $i \in \{1,2\}$ are determined for $w_\mathrm{p}^2 \in [-1.5, 1.5]$ (cf. Fig. 11.2):

$$\tilde{G}_1^2[k] \in \mathbb{G}_1^2 = \begin{bmatrix} 0.2 \\ [0.07, 0.3] \end{bmatrix}, \qquad \tilde{G}_2^2[k] \in \mathbb{G}_2^2 = \begin{bmatrix} -0.2 \\ [-0.05, 0.2] \end{bmatrix}. \tag{11.55}$$

The RCIS are shown in Fig. 11.2. For both setups, the RCIS of the first subsystem are smaller than for the second subsystem. This is in order to keep the couplings small enough to guarantee robustness. In addition, the form of the RCIS is similar for both setups. However, the overall volume of the RCIS for the DMPC architecture (solid) is larger than that for the decentralized architecture (dashed). This illustrates that the knowledge about the couplings enables the usage of larger RCIS. Note that the volume of the RCIS of the first system is significantly larger, while the volume of the second system is slightly smaller. This indicates that a volume reduction of the error ellipsoids (due to the known couplings) enables a larger increase of the RCIS of the first subsystem than of the second subsystem. To maximize the overall volume, the RCIS of the first subsystem is expanded until the error set reaches a similar size as for the decentralized case. By adding high weights for $\mathcal{Q}_i^2$ in cost function (11.32a), an increase of the volume of $\mathcal{E}_i^2$ can be forced. However, this results in a smaller overall volume of the RCIS.

11.3.1. Decentralized MPC

This section demonstrates the properties of the decentralized MPC employing the distributed JMS (11.54). In order to investigate the effects of the RCIS, the coupled JMS (11.54) is simulated in closed-loop with three different MPC setups:

   I. Local MPCs without quadratic constraints. This setup is similar to the non-robust formulation considered in Section 8.2.
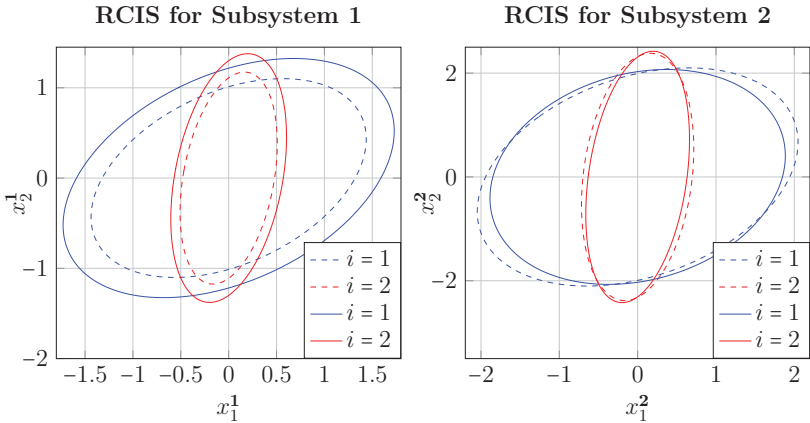
Figure 11.2.: RCIS $\mathcal{E}(\mathcal{Q}_i^n)$ for the first (left) and second (right) subsystem for a decentralized MPC setup according to Theorem 11.1 (dashed) and for a DMPC setup according to Theorem 11.3 (solid).

II. Local robust MPCs both employing the RCIS determined in Section 8.2 (neglecting the couplings during the RCIS design).

III. Robust decentralized MPC according to Theorem 11.2.

All formulations are simulated 200 times with $N^n = 4$, $Q^n = I$, and $R^n = 10$ and both cost function definitions. The initial states are $x^1[0] = [0.9\ 1.4]^\top$ and $x^2[0] = [-0.7\ 1.2]^\top$. The reference is the origin. The costs, number of infeasible problems, and computation times are shown in Table 11.1. The resulting costs and numbers of infeasible problems are similar for both cost functions. For the first subsystem, the costs are significantly larger for the decentralized approach (III.) than for setup I. and II., since larger inputs are needed to transfer the states into the smaller RCIS. However, the increased effort guarantees stability for both subsystems. This can be seen by the number of infeasible problems of the MPC for the second subsystem. While the consideration of the RCIS in setup II. reduces the number of infeasible problems compared to setup I., only the decentralized MPC with RCIS determined according to Theorem 11.2 (III.) stabilizes the overall system for all simulation runs. The constraints are satisfied only for setup III in all simulations.

In general, the computation time is larger for the EoC approach. In addition, it can be seen that solving the QCQP problem (in setup II. and III.) takes more time than solving the QP for setup I. Since the quadratic output constraints (11.21e) are only considered for the MPC of the first subsystem in setup III., the computation times are larger than for setup II. for the first subsystem. However, the computation times are very low considering that a nonlinear distributed JMS is controlled.

Table 11.1.: Comparison of the costs, number of infeasible problems, and computation times for the three setups employing both cost function definitions.

	subsystem	1			2		
	setup	I.	II.	III.	I.	II.	III.
CoE	cost	1.7	1.7	24.4	-	-	25.5
	infeas.	0	0	0	45	22	0
	time in ms	1.8	2.3	2.6	1.7	2.3	2.3
EoC	cost	1.7	1.7	25.1	-	-	27.5
	infeas.	0	0	0	43	22	0
	time in ms	2.4	2.9	3.2	2.3	2.9	2.9

Similar results are observed for an EoC MPC configuration with input references $u_r^1 = 2.5$ and $u_r^2 = -2.5$ and initial states $x^1[0] = x^2[0] = \mathbf{0}$. This configuration is intended to drive the system to the edge of the feasible set (which might be desired in economic setups). In this configuration, setup I. results in 370 infeasible problems for the second subsystem. The MPC of Setup II. results in 336 infeasible problems, while the decentralized the MPC of setup III. is feasible in all simulations.

These simulations demonstrate that the RCIS design proposed for decentralized MPC is valid, i.e., it guarantees recursive feasibility and constraint satisfaction.

11.3.2. Distributed MPC

In this section, the control performance and robustness of the decentralized MPC and DMPC approaches are compared. To this end, the JMS (11.54) is simulated with the decentralized MPC, the non-cooperative DMPC, and the cooperative DMPC. The setups are simulated 15 times for 200 different initial states $x^n[0]$ with both cost functions employing $Q^n = I$ and $R^n = 10$ for $n \in \{1, 2\}$. The computation times and costs are shown in Table 11.2. As expected, the costs are significantly lower for the DMPC approaches than for the decentralized approaches. This is mainly caused by the larger RCIS for the DMPC setups. However, both DMPC approaches perform equally – which is unexpected. This is caused by the fact that the coupling is caused by the second state of the first subsystem. The costs caused by the coupling can be minimized by steering the state of the first subsystem to the origin. This is already required by the costs of the first subsystem. Hence, the minimization of the global cost function does not change the goal of the first subsystem and the performance is almost the same. A completely different result is achieved if the local and the global goals differ (see the simulation results for the hot stamping line in Chapter 13). Furthermore, the results show that CoE and EoC perform very similar for the system under consideration. All MPC problems have been feasible for all simulation runs. Hence, the remainder of this section only considers the EoC approach.

While the computation times are the same for the decentralized and non-

Table 11.2.: Simulation results of the decentralized MPC as well as both DMPC architectures for both cost formulations (CoE and EOC).

		costs			time in ms		
		decent.	non-co.	coop.	decent.	non-co.	coop.
CoE	subsys 1	0.271	0.185	0.185	2.6	2.6	3.2
	subsys 2	1.615	0.867	0.869	2.5	2.4	2.4
	Σ	1.886	1.052	1.054			
EoC	subsys 1	0.271	0.185	0.185	3.2	3.2	4.3
	subsys 2	1.615	0.869	0.871	3.1	3.0	3.0
	Σ	1.886	1.054	1.056			

cooperative DMPC, the computation time is higher for the cooperative DMPC in subsystem one. The additional computation time is due to the construction of the cost prediction matrices $W^n[k]$ and $q^n[k]$ according to Theorem 11.4 and 11.5.

All in all, the simulations demonstrate the effectiveness of the proposed RCIS design approaches and the advantages of a distributed architectures over a decentralized architecture. For this example, the usage of a cooperative DMPC is not necessary, since no performance improvement is achieved and the computation time is higher than for the non-cooperative DMPC.

Due to the conservatism and complexity of the RCIS design, it is of interest whether it is sufficient to determine the RCIS independently for all subsystems according to Section 8.1 (neglecting the couplings). To this end, three simulations are performed for the decentralized MPC and DMPC approaches employing RCIS that are determined independently for both subsystems:

I. Setup with 200 different initial states ($x_0^n \in \mathcal{E}_i^n \ \forall \, i, n \in \{1, 2\}$) and 15 simulation runs each. The reference for the control is the origin.

II. Setup with 200 simulation runs with $x^1[0] = [0.9, 1.4]^\top$ and $x^2[0] = [-0.7, 1.2]^\top$. The reference for the control is the origin.

III. Setup with 200 different initial states ($x_0^n \in \mathcal{E}_i^n \ \forall \, i, n \in \{1, 2\}$) and 15 simulation runs each. The reference for the control is $u_r^1 = 2.5$ and $u_r^2 = -2.5$ and $x_r^1 = x_r^2 = \mathbf{0}$. These references push the system states to the edge of the feasible sets.

For all setups $N^n = 4$, $Q^n = I$, and $R^n = 10$ is used. The number of infeasible problems that occurred during these simulations are shown in Table 11.3. For the setups II. and III., the decentralized MPC setup results in a significant number of infeasible problems. In consequence, the constraints have been violated for some simulation runs. Hence, a decentralized MPC employing independently determined RCIS does not guarantee robustness. In contrast, both DMPC approaches have been feasible for all simulation runs in all setups. These results indicate that independently determined RCIS can be used in combination with the DMPC approaches though

Table 11.3.: Number of infeasible runs for the decentralized MPC as well as both DMPC architectures with independently determined RCIS.

infeasible runs		decent. MPC	non-coop. DMPC	coop. DMPC
Simulation I.	subsystem 1	0	0	0
	subsystem 2	0	0	0
Simulation II.	subsystem 1	0	0	0
	subsystem 2	165	0	0
Simulation III.	subsystem 1	0	0	0
	subsystem 2	1962	0	0

robustness is not guaranteed in theory. Hence, a DMPC employing independently determined RCIS is a considerable alternative to the RCIS design approaches presented in this chapter – in particular for applications with large system dimensions.

Multiple Subsystems

In order to investigate the limits of the decentralized MPC and DMPC approaches, distributed JMS with more than two subsystems are considered. Here, the JMS (11.54) is used as a basis and copies of the second subsystem are appended to the last subsystem. The RCIS for a distributed JMS with three subsystems, determined according to Theorem 11.3, are shown in Fig. 11.3. RCIS for a decentralized setup cannot be found, demonstrating the reduced conservatism of the RCIS design procedure for DMPC. However, the resulting RCIS decrease in volume significantly
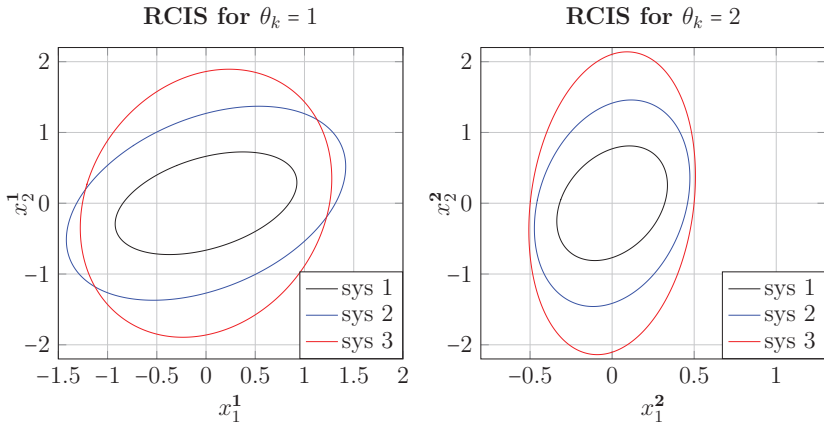
Figure 11.3.: RCIS $\mathcal{E}\left(\mathcal{Q}_i^n\right)$ for all three subsystems for the local first (left) and second (right) Markov state for DMPC according to Theorem 11.3.

Table 11.4.: Simulation results for the distributed JMS with four subsystems controlled by a decentralized MPC and a non-cooperative DMPC.

	decentralized MPC				non-coop. DMPC			
	sys 1	sys 2	sys 3	sys 4	sys 1	sys 2	sys 3	sys 4
inf. runs	0	99	69	86	0	0	0	0
comp. time in ms	2.9	2.9	2.9	2.9	2.8	2.8	2.8	2.8

from the third system to the first system. This is due to the fact, that the overall disturbances accumulate from the first to the last system. Hence, the volume of the RCIS decreases with an increasing number of subsystems. This is why, for the distributed test system with four subsystems, no RCIS can be determined anymore. All in all, the RCIS design is applicable to more than two subsystems and the distributed approaches exhibit a reduced conservatism compared to the decentralized approach. However, the design procedure is restrictive and only applicable to distributed systems with few subsystems.

Nevertheless, a decentralized MPC and a non-cooperative DMPC are considered for the distributed JMS with four subsystems. To this end, RCIS are designed independently for all subsystems ignoring the couplings. Subsystem 3 and 4 and the corresponding MPCs are parameterized as subsystem 2 in Section 11.3.1. The initial sates are $x^1[0] = [0.9 \ 1.4]^\top$ and $x^3[0] = x^4[0] = x^2[0] = [-0.7 \ 1.2]^\top$. The number of infeasible problems and average computation times for 100 simulation runs are shown in Table 11.4. Again, the decentralized setup does not stabilize the system for all simulations if the RCIS are designed independently. However, for the DMPC setup, all simulation runs are feasible and all subsystems are robustly stabilized. These results support the conclusion that a DMPC approach employing RCIS, that are determined independently, robustly stabilizes the JMS in many cases.

11.4. Summary

In this chapter, design procedures for a decentralized MPC architecture as well as non-cooperative and cooperative DMPC architectures have been presented. For both, decentralized and distributed architectures, specialized RCIS design approaches have been proposed in order to determine RCIS. The simulations confirm the effectiveness of the design approaches. However, the SDP problems are quite conservative and its solution is computationally costly. Thus, these approaches are only suitable for small to medium scale processes.

The robust QCQP-based MPC formulations have been presented for all three architectures. In order to minimize a global cost function, cost prediction procedures are proposed for both MPC cost definitions. While the decentralized MPC and non-cooperative DMPC exhibit nearly the same computational effort, the computational

burden of the cooperative DMPC is increased by the involved cost predictions.

The simulation results demonstrate the validity of the MPC approaches and show the advantages of DMPC over decentralized MPC (larger RCIS and lower costs). However, for the example considered, the cooperative DMPC performs as good as the non-cooperative DMPC. This effect is investigated in more detail in the following chapters with the models of the hot stamping process.

The simulations show that recursive feasibility is not guaranteed for decentralized MPC if independently determined RCIS are employed. The DMPC approaches, however, have been feasible for all simulations performed. Hence, in case that the distributed RCIS design fails (due to the conservatism or the system size), DMPC employing independently designed RCIS should be considered.

Part V.

Application to the Hot Stamping Process

12. Investigation of Isolated MPC for a Roller Hearth Furnace

In this part, the applicability of the MPC and DMPC approaches to the models of an existing hot stamping line is evaluated[1]. The modeling of the production process has been presented in Chapter 4. In Chapter 12, the different MPC approaches for a single nonlinear JMS are tested with the model of a roller hearth furnace (RHF). See Section 4.2.1 for details on the model. The RHF is employed to demonstrate the properties of the MPC as it is the subsystem with the most process variables (inputs), states, and Markov states. In addition, the chamber furnace is, in principle, a smaller version of the RHF, and the control of the transportation time of the robot is trivial. The control of a reduced model of the press has been investigated in [116]. In order to illustrate the properties of the MPC approaches, two scenarios are considered: First, a setup is considered where the goal of the MPC is to keep the product properties and furnace states as close as possible to a given reference. Second, an economic MPC setup is considered. The MPC is supposed to control the process variables such that the energy consumption of the furnace is minimized. The distributed control of the overall production line is considered in Chapter 13.

12.1. MPC Setup and Design of Robust Control Invariant Sets

As described in Section 5.2, the JMS modeling the RHF is transformed into an augmented system description by combining the state and output vector. The resulting JMS has $n_x = 34$ states, $n_u = 11$ inputs, $n_w = 12$ disturbances, and $n_\theta = 4$ Markov states. The state references and constraints are shown in Table 12.1. The localization of the corresponding quantities is illustrated in Fig. 12.1. The constraints for the furnace temperatures $T_{f,m}$ result from the technical limitations of the furnace. In a realistic setup, the furnace time t_f shall not deviate more than 1.5 % from the reference to ensure a constant cycle time. The blank temperatures in the furnace $T_{b,m}$ do not have to be constrained. The critical maximum temperatures for the coating and the base material cannot be reached, since the furnace temperatures

---

[1]For confidentiality reasons, not all given parameters, bounds, and references match the real process parameters exactly. However, the magnitudes and relations are realistic. For the same reason, the transition probabilities, i.e., breakdown and repair rates, are not presented.
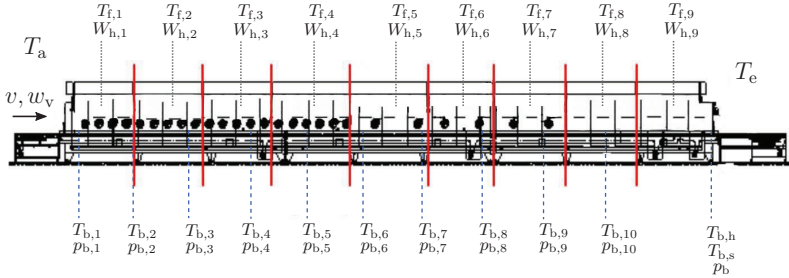
Figure 12.1.: Sketch of the RHF and the localization of the states, inputs, and disturbances. The furnace-related quantities are shown on the top, while the blank-related quantities are shown on the bottom.

Table 12.1.: Bounds, references, and cost weights for the states of the RHF (all temperatures $T_{(\cdot)}$ in °C, furnace time t_f and its disturbance ν_t in s, and probabilities $p_{(\cdot)}$ dimensionless).

	$T_{f,1}$	$T_{f,2}$	$T_{f,3}$	$T_{f,4}$	$T_{f,5}$	$T_{f,6}$	$T_{f,7}$	$T_{f,8}$	$T_{f,9}$	t_f	$T_{b,1}$	$T_{b,2}$	$T_{b,3}$
max.					980					341	-	-	-
x_r	860	870	880	880	880	890	890	890	940	336	101	428	589
min.					700					331	-	-	-
$Q_{l,l}$					1					10		0.1	

	$T_{b,4}$	$T_{b,5}$	$T_{b,6}$	$T_{b,7}$	$T_{b,8}$	$T_{b,9}$	$T_{b,10}$	$p_{b,1:10}$	ν_t	$T_{b,h}$	$T_{b,s}$	p_b
max.	-	-	-	-	-	-	-	-	-	980	730	-
x_r	679	747	795	833	863	879	887	1	0	920	650	1
min.	-	-	-	-	-	-	-	-	-	900	-	-
$Q_{l,l}$			0.1						0	0	10	10	0

are bounded according to Table 12.1. Finally, the temperatures in the hard and the soft part of the blanks (after the partial hardening process) have to be bounded such that the desired hardness values can be achieved. See page 13 for details. To this end, the lower bound for the temperature in the hard part $T_{b,h}$ is set to ensure that the base material has a martensitic matrix. The upper bound for the temperature in the soft part is set to the A_{c1} temperature of the base material minus 20 °C safety margin. This guarantees a lower hardness in the considered area.

The state costs are defined by a diagonal matrix Q. The weighting factors for the states are shown in Table 12.1. Important factors for the production process are the final part temperatures and the cycle times determined by the furnace time. Hence, the deviation of these quantities from their references is penalized most.

All temperatures are initialized randomly with values from the interval of ±10 °C around the reference. The furnace is assumed to be empty at the beginning of each

Table 12.2.: Bounds, references, and cost weights for the inputs of the RHF (heating powers $W_{\text{h},n}$ in kW, transportation velocity v^{-1} in s/m, and partial hardening time $t_{\text{ph},1}$ in s).

	$W_{\text{h},1}$	$W_{\text{h},2}$	$W_{\text{h},3}$	$W_{\text{h},4}$	$W_{\text{h},5}$	$W_{\text{h},6}$	$W_{\text{h},7}$	$W_{\text{h},8}$	$W_{\text{h},9}$	v^{-1}	$t_{\text{ph},1}$
max.	480	320	260	192	192	192	192	90	90	20	30
u_{r}	125	79	42	65	46	57	13	0	39	16.7	25
min.	0									13.4	20
$R_{l,l}$	0.01									1	

simulation, and the Markov state is initialized in normal operation mode ($\theta_0 = 1$).

The temperatures $T_{\text{b,h}}$ and $T_{\text{b,s}}$ of the exiting blanks are set to 0 if no blanks exit the furnace (in particular for the reversing mode ($\theta_k = 2$) and the standby mode ($\theta_k = 4$)). See Section 4.2.1 for details. Hence, the prediction of the expected values $\bar{T}_{\text{b,h}}$ and $\bar{T}_{\text{b,s}}$ can differ significantly from the actual temperatures of an exiting blank. To formulate the constraints $T_{\text{b,h}} \geq 900\ °\text{C}$ and $T_{\text{b,s}} \leq 730\ °\text{C}$ for the whole prediction horizon, the probability that a blank pair exits the furnace has to be considered. Employing the corresponding probability that a blank pair exits the furnace, the constraints can be formulated as follows:

$$\bar{T}_{\text{b,h}}[j] \geq p_{\text{b}}[j] \cdot 900\ °\text{C} \qquad \text{and} \qquad \bar{T}_{\text{b,s}}[j] \leq p_{\text{b}}[j] \cdot 730\ °\text{C}. \qquad (12.1)$$

Note that this formulation introduces time-variant constraints. The issue also has an effect on the CoE cost function, since the difference between the expected values $\bar{T}_{\text{b,h}}$ and $\bar{T}_{\text{b,s}}$ and fixed reference values is minimized. To mitigate this problem, a reference $\bar{\boldsymbol{x}}_{\text{r}}[k]$ considering the probabilities of blank exits can be employed:

$$\bar{x}_{\text{r},32}[j] = p_{\text{b}}[j] \cdot 920\ °\text{C} \qquad \text{and} \qquad \bar{x}_{\text{r},33}[j] = p_{\text{b}}[j] \cdot 650\ °\text{C}. \qquad (12.2)$$

Hence, the state reference $\bar{\boldsymbol{x}}_{\text{r}}[k]$ is time-variant.

The references and constraints for the inputs are shown in Table 12.2. The bounds result from the technical limitations of the RHF. Since an optimal output and state reference tracking is desired, the input cost weights are relatively small.

The plant model mismatch processes $\nu_{\text{T},m}$ for the furnace segment temperatures $T_{\text{f},m}$ are simulated by truncated normal distributions with bounds and standard deviations according to Table 4.1 on page 57. Furthermore, the following disturbance process is considered:

$$w[k] = \begin{bmatrix} T_{\text{a}}[k] & w_{\text{v}}[k] & T_{\text{e}}[k] \end{bmatrix} \sim \begin{cases} k \leq 15: & \mathcal{N}\left([25\ \ 0.05\ \ 50]^\top, \text{diag}(0.1, 0.1, 0.1)\right) \\ k > 15: & \mathcal{N}\left([25\ \ -0.05\ \ 30]^\top, \text{diag}(0.1, 0.1, 0.1)\right) \end{cases}. \qquad (12.3)$$

RCIS Design

For the implementation of the robust MPC approaches, the RCIS and a Lyapunov function have to be determined for the RHF. Strictly speaking, it is not possible to

determine the RCIS according to Section 8.1 for two reasons:

1. The given reference x_r is not a steady state for the second Markov state. If this mode is active, the blank temperatures $T_{\mathrm{b},m}[k]$ converge to the corresponding furnace segment temperatures. Hence, it is not possible to find an input sequence that ensures PMSS with respect to the specified reference x_r.

2. The over-approximation of only the system matrices $\tilde{A}_i[k]$ would lead to a matrix polytope $\mathbb{A}_i$ with up to 2^{34} vertex matrices $A_{i,[l]}$. This obviously results in an SDP problem that cannot be solved.

However, if the focus is on guaranteeing constraint satisfaction and recursive feasibility, the concepts developed to derive RCIS can still be applied under reasonable assumptions and relaxations. The first problem can be mitigated, by ignoring the PMSS constraints if the system is in the second Markov state. Furthermore, the invariance constraints are switched off, when the blank temperatures $T_{\mathrm{b},m}$ have been driven too far away from the reference (in $\theta_k = 2$). This does not endanger recursive feasibility, since the blank temperatures cannot exceed the furnace temperatures. For both, the same upper bounds hold, which can always be satisfied by switching off the heating. In order to solve the second problem, the number of polytope vertices is reduced significantly by only considering worst-case scenarios (see below). In addition, a reduced JMS without probability states is employed.

Error Approximation: The first step to determine the RCIS is to determine an ellipsoidal over-approximation of the one-step prediction errors $e_i[k]$. These errors are caused by the linearization errors, model uncertainties, and disturbance inputs $w[k]$. To reduce the conservatism of the error set calculation, Monte Carlo simulations are employed instead of the Lagrange remainder over-approximation and interval arithmetics. The result is a realistic box approximation $\lfloor e_i[k] \rfloor$ of the overall one-step prediction error. The simulations show that the errors are mainly determined by the model errors of the furnace temperatures (cf. Table 4.1). The error bounds for the blank temperatures are shown in Table 12.3.

Finally, the error $\lfloor e_i[k] \rfloor$ is over-approximated by the error ellipsoids $\mathcal{E}(E_i)$. Here, minimum-volume ellipsoids that contain the box $\lfloor e_i[k] \rfloor$ are determined by solving an SDP. The resulting over-approximation is rather conservative. The maximal admissible values of the components of the disturbance are $\sqrt{22} \approx 4.7$ times the values shown in Table 12.3 (22 is the dimension of the state vector of the reduced JMS without probability states; see below). To reduce the conservatism, the ellipsoids

Table 12.3.: Bounds of the one-step pred. error for the blank temperatures in °C.

	$T_{\mathrm{b},1}$	$T_{\mathrm{b},2}$	$T_{\mathrm{b},3}$	$T_{\mathrm{b},4}$	$T_{\mathrm{b},5}$	$T_{\mathrm{b},6}$	$T_{\mathrm{b},7}$	$T_{\mathrm{b},8}$	$T_{\mathrm{b},9}$	$T_{\mathrm{b},10}$	$T_{\mathrm{b,h}}$	$T_{\mathrm{b,s}}$
max.	3.8	1.0	1.5	1.8	1.1	1.5	1.6	1.2	1.2	1.2	1.2	1.8
min.	-3.8	-1.1	-0.7	-1.0	-1.0	-0.7	-1.5	-1.3	-1.0	-1.4	-1.9	-1.6

are contracted by that factor[2]. In the final calculation of the invariance constraints by Minkowski difference approximations, the non-contracted error ellipsoid is used.

Polytopic System Approximation: The nonlinear system dynamics is over-approximated by matrix polytopes $\mathbb{A}_i$, $\mathbb{B}_i$, and $\mathbb{G}_i$ for all Markov states $i \in \Theta$. Here, the vertex matrices are determined for worst-case situations only[3]. To keep the problem dimensions as small as possible, the system matrices are approximated by polytopes with 8 vertices. These vertices are defined by the permuted combinations of the following extreme situations:

- All temperatures (in the state and disturbance vector) have the maximum value of $x_r + 30\,°C$ and no blanks are in the furnace ($p_{b,m} = 0$) or all temperatures have the minimum value of $x_r - 30\,°C$ and $p_{b,m} = 1$ for all $m \in \{1, \ldots, 10\}$.

- Minimum or maximum transportation velocity $v[k]$ according to Table 12.2.

- Minimum or maximum partial hardening time $t_{ph,1}[k]$ according to Table 12.2.

Since the system equations are linear in the remaining quantities, these do not influence the linearization.

The system equations for $\theta_k = 1$ and $\theta_k = 3$ only differ in the dynamics of the probability $p_{b,1}$ and the furnace segment temperature $T_{b,1}$. As a result, the differential inclusion $\mathbb{A}_1, \mathbb{B}_1$, and $\mathbb{G}_1$ of the dynamics for the first Markov state also includes the dynamics of $\theta_k = 3$. Thus, the dynamics of the third Markov state can be combined with that of the first Markov state. In addition, all states that cannot be influenced by the inputs, i.e., the blank position probabilities and the disturbance model, are left out. This can be done since the effect of these quantities on the remaining states is considered by the error over-approximation and determination of the matrix polytopes. Hence, a reduced JMS with $n_x = 22$ states, $n_u = 11$ inputs, $n_w = 3$ disturbances, and $n_\theta = 3$ Markov states results from this procedure.

For this setup, the resulting disturbance set over-approximation is still too large to solve the SDP according to Theorem 7.3. Hence, the approach presented in Remark 7.4, in which the error set is scaled down, is employed. The shape matrices $\mathcal{E}(x_r, \breve{\mathcal{Q}}_{i,m})$ and the Lyapunov matrices $\mathscr{P}_i$ for the original system are derived by adding zero columns and rows for the neglected states. The RCIS and the Lyapunov matrices for the third Markov state are the same as for the first Markov state.

This procedure demonstrates the difficulties in determining RCIS for real systems. However, employing the knowledge of the system and reducing the error sets, finally provides a suitable RCIS.

---

[2]In theory, this excludes possible disturbances and does not guarantee robustness anymore. However, the conservatism is reduced significantly and the following simulations show that the approximation is reasonable.

[3]This approximation is not valid for $\theta_k = 2$. However, as discussed in the beginning of this section, this issue can be ignored.

12.2. MPC for Reference Tracking

In this section, the MPC approaches are used for reference tracking. The main goal is to keep the furnace and blank temperatures as close as possible to their references and satisfy the constraints. To this end, the influence of the continuous disturbances, e.g., ambient temperature and transportation speed deviations, and the abrupt disturbances (production line stops) have to be compensated. For the setup presented in the previous section, both MPC approaches (CoE and EoC) are tested. To investigate the influence of the RCIS, the MPC formulations (according to Theorem 8.1 on page 123) are simulated with and without the quadratic invariance and stability constraints. All four formulations are simulated for 100 Markov state trajectories with 30 time steps each employing a prediction horizon length of $N = 9$, $N = 11$, and $N = 13$. In addition, the RHF is simulated using a hysteresis controller for the heating powers $W_{\mathrm{h},n}$. The hystereses are parametrized with values between ±5 °C and ±10 °C. Furthermore, the transportation speed $v[k]$ and partial hardening time $t_{\mathrm{ph},1}[k]$ are fixed at their references. This type of controllers is the standard setup for existing furnaces. Other standard controllers, such as PID controllers, are not considered since these can not guarantee constraint satisfaction.

The different control approaches are compared by costs, computation times, number of constraint violations, and infeasible problems. To evaluate the control performance, the average costs are calculated for all approaches:

$$J = \frac{1}{100} \sum_{m=1}^{100} \sum_{k=1}^{30} \left(\|x[k] - x_{\mathrm{r}}\|_{Q_i}^2 + \|u[k-1] - u_{\mathrm{r}}\|_R^2 \right), \qquad (12.4)$$

where Q_i and R are defined as in Table 12.1 and 12.2 but for $i = 2$ and $i = 4$ the output cost weights are set to zero. This prevents the costs to be dominated by the difference of $T_{\mathrm{b,h}}[k] = 0$ and $T_{\mathrm{b,s}}[k] = 0$ and the reference for the case that no blanks exit the furnace. The simulation results are shown in Table 12.4.

The optimization problems have been feasible and the constraints have been satisfied for all setups at all times. The costs J of the different MPC approaches are very similar. The costs for the robust approaches are increased slightly due to the additional quadratic constraints. As expected from the results in Section 6.3,

Table 12.4.: Average costs and computation times for the different approaches.

N	approach	CoE	CoE rob.	EoC	EoC rob.	Hysteresis
9	J in $\cdot 10^5$	2.173	2.195	2.169	2.192	2.676
	comp. time in ms	19	54	42	75	-
11	J in $\cdot 10^5$	2.173	2.195	2.169	2.191	2.676
	comp. time in ms	37	70	69	106	-
13	J in $\cdot 10^5$	2.171	2.194	2.167	2.191	2.676
	comp. time in ms	32	91	86	146	-

the costs of the EoC approach are lower than for the CoE approach. However, the difference is only marginal. In addition, the costs J according to (12.4) decrease only insignificantly from $N = 9$ to $N = 13$. Hence, larger prediction horizons are not necessary. The costs of all MPC approaches are about 19 % lower than the costs of the common hysteresis controller. This effect is illustrated in the plots in Fig. 12.2.

The computation times of the CoE MPC are significantly smaller than for the EoC MPC (44 % in average). This coincides with the results presented in Section 6.3. In addition, the computation time for the non-robust formulation (employing a QP formulation) is about 49 % lower than for the robust formulation (employing a QCQP formulation). However, for all cases, the average computation time per cycle is below 0.15 seconds. This is quite low compared to the cycle time of 36 seconds. Hence, all formulations are fast enough for an online implementation. This illustrates the efficiency of the prediction algorithms presented in Chapter 6 in combination with online linearization.

The mean values and envelopes of the temperatures of the exiting blanks $T_{b,h}[k]$ and $T_{b,s}[k]$, the transportation velocity $v^{-1}[k]$, and the partial hardening time $t_{ph,1}[k]$ are shown in Fig. 12.2 for both non-robust MPC formulations and $N = 9$. In addition, the results for the hysteresis controller are shown. The mean values and envelopes of the blank temperatures $T_{b,h}[k]$ and $T_{b,s}[k]$ are virtually the same for both MPC approaches. This illustrates the very similar costs. The temperatures are zero for the first 10 time steps, since it takes 10 production cycles until the first blanks reach the end of the furnace. The jump in the results for $T_{b,s}[k]$ at $k = 15$ is due to the jump in the disturbance $w_v[k]$. The effect can also be seen in the partial hardening time $t_{ph,1}[k]$. Both temperatures are kept very close to the defined reference. This illustrates the effectiveness of the proposed MPC. The small offset from the reference is due to the fact that the references x_r and u_r are not exactly a steady state. In comparison, the standard hysteresis controller results in similar mean values of the temperatures, but the variances are significantly higher than for the MPC approaches. The disturbance change at $k = 15$ has a stronger impact on the temperature $T_{b,s}[k]$ for the reference controller. In addition, the frequent starts and stops of the burners lead to an inefficient burning process and thermal wear of the burners. This illustrates some of the advantages of the MPC approaches compared to the hysteresis controller. However, the bigger advantages are demonstrated in Chapter 13 in the context of distributed control for the whole production line.

The plots of the input values show a slight difference between the EoC and CoE MPC. The CoE approach shows a more aggressive behavior for v^{-1} at $k = 9$ shortly before the first blanks exit the furnace. Obviously, the additional minimization of the variance of states in the EoC approach suppresses this effect. This is a favorable behavior in terms of actuator wear. The values of the partial hardening time illustrate the direct reaction to a jump in the disturbance.

The effect of the Markov state trajectory on the blank temperatures and on the inputs is illustrated in Fig. 12.3. The Markov state trajectory is shown at the bottom. During the two production line halts, the blank temperature at the beginning of
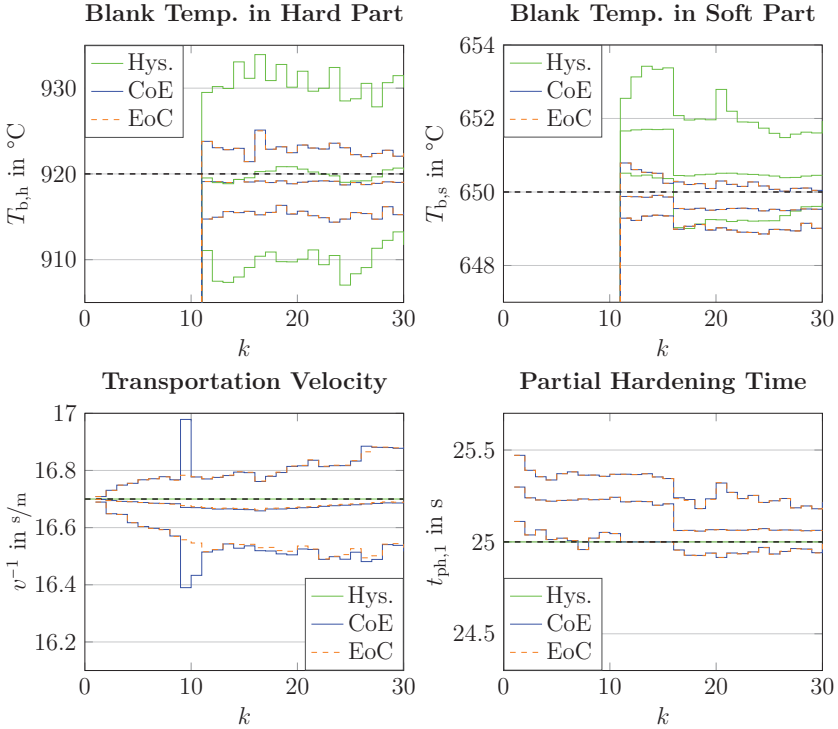
Figure 12.2.: Mean values and envelopes of the temperatures of the exiting blanks $T_{b,h}[k]$ and $T_{b,s}[k]$ and inputs for the non-robust MPC formulations and the hysteresis controller.

the furnace $T_{b,1}$ increases in each time step towards the local furnace temperature (plot on the top). To mitigate this deviation from the reference value, the heating ($W_{h,1}[k]$ and $W_{h,2}[k]$) is switched off in the corresponding segments (third plot). The temperature of the exiting blanks ($T_{b,h}[k]$) is stabilized close to the reference despite the production line halts (second plot).

Remark 12.1. *If the original reference x_r according to Table 12.1 is employed for the blank temperatures instead of (12.2) for the CoE MPC, the results change significantly. In this case, the temperatures $T_{b,h}$ and $T_{b,s}$ are pushed to their upper limits. This is caused by the fact that these temperatures are set to zero for the second and the fourth Markov state. Hence, the expected values of the predicted temperatures $\bar{T}_{b,h}[j]$ and $\bar{T}_{b,s}[j]$ are lower than the actual temperatures. In this case, the costs quadruple for some configurations. Hence, the CoE approach has to be used with the adapted references $\bar{x}_r[k]$ for the expected value of the states.*
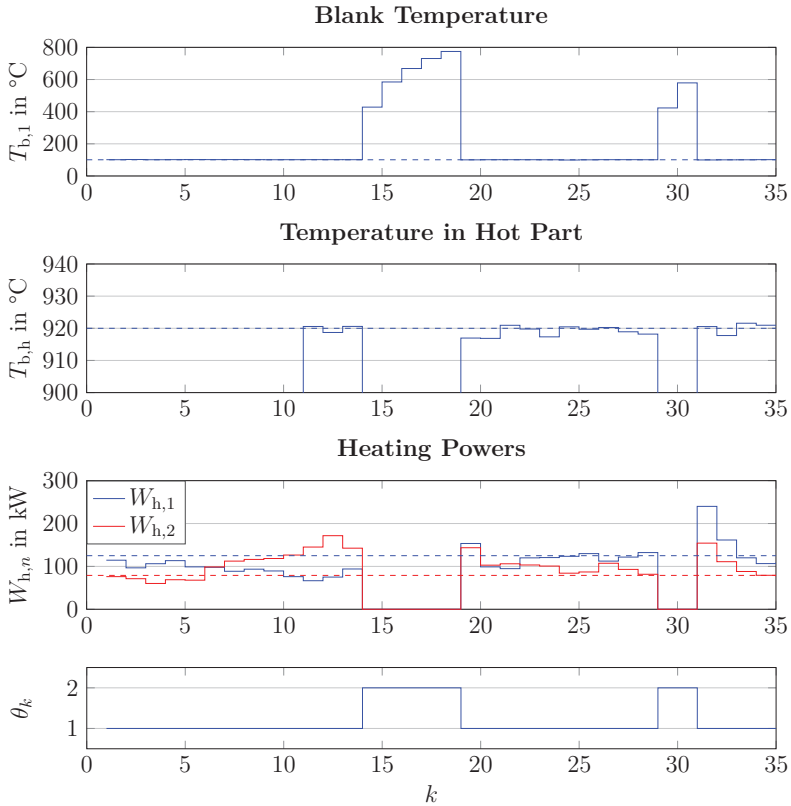
Figure 12.3.: Blank temperatures $T_{b,1}[k]$ and $T_{b,h}[k]$ as well as the heating powers $W_{h,1}[k]$ and $W_{h,2}[k]$ in the first two furnace segments for a specific Markov state trajectory (bottom). The corresponding reference values are shown as dashed lines.

The simulations illustrate that the MPC approaches can be used for reference control of the RHF. Since the RHF is stabilized and all constraints are satisfied, the relaxations and assumptions made for the RCIS design prove to be reasonable.

12.3. MPC with Minimal Energy Consumption

The MPC approaches presented in Chapter 8 can also be used to determine economically optimized operation points. This is illustrated with an MPC configuration,

Table 12.5.: Costs, Computation times and number of infeasible problems for CoE-based and EoC-based MPC with and without invariance constraints.

approach	CoE		EoC	
invariance constraints	no	yes	no	yes
costs J in $\cdot 10^7$	5.03	15.34	5.02	15.32
computation time in ms	24	81	60	116
infeasible prob.	1290	0	1288	0

Table 12.6.: Minimum values of furnace and blank temperatures of all simulation runs in °C.

	$T_{f,1}$	$T_{f,2}$	$T_{f,3}$	$T_{f,4}$	$T_{f,5}$	$T_{f,6}$	$T_{f,7}$	$T_{f,8}$	$T_{f,9}$	$T_{b,h}$	$T_{b,s}$
CoE	704	756	805	812	832	847	861	866	899	**877**	622
EoC	704	756	805	812	832	847	861	866	899	**877**	622
CoE rob.	794	838	862	864	860	868	877	881	922	**901**	631
EoC rob.	794	838	862	864	860	868	877	881	922	**901**	631

aiming at minimizing the energy consumption of the RHF. Hence, the MPC is supposed to steer the RHF to an operation point with minimized furnace temperatures and to guarantee that the part properties always satisfy the specifications, i.e., the constraints. By this configuration the process is operated at the boundary of the process window and the part properties may be steered close to the bounds of the specifications. To investigate the properties of this economic MPC setup, the same configuration as presented in the previous section is employed. Only the input reference u_r and input costs R are changed in order to minimize the energy consumption. The references for the heating powers $W_{h,m}$ are set to 0 and the corresponding input cost weights $R_{m,m}$ are increased to 100. This setup pushes the states to the edge of the feasible set. Hence, the stability constraint (8.13e), which forces a convergence to x_r, is not used for the economic MPC setup. Only the invariance constraints (8.13d) are employed. Again, the CoE and EoC MPCs with and without quadratic constraints (robust and non-robust version) are considered.

The four formulations are simulated with $N = 11$ for 100 times and 50 time steps each. The results are shown in Table 12.5. As for the reference tracking setup, the CoE and EoC approaches perform very similar. However, there is a different behavior for the robust and non-robust approaches. While both non-robust formulations result in a large number of infeasible problems and constraint violations, the robust approaches, employing the invariance constraints, are recursively feasible and satisfy the constraints. This causes the difference in the costs. The computation times are similar to the reference tracking setup.

The mean values and envelopes for the blank temperature $T_{b,h}$ resulting from the control with the EoC MPC are shown in Fig. 12.4. The minimal values of the
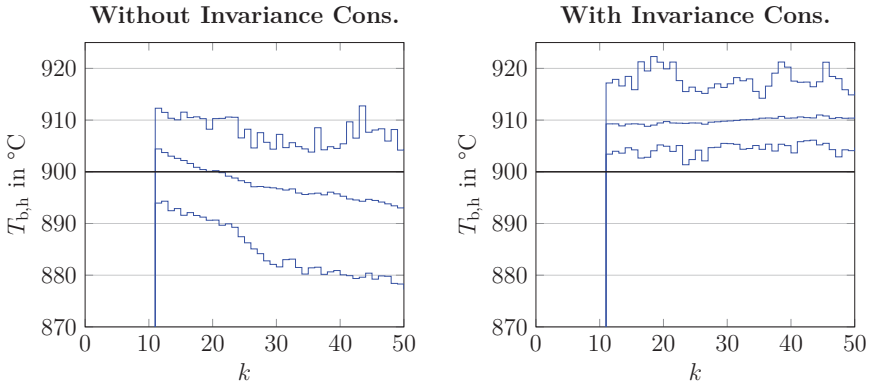
Figure 12.4.: Mean values and enveloping curves of the temperatures of the exiting blanks $T_{b,h}[k]$ for the EoC-based MPC with and without invariance constraints.

furnace temperatures $T_{f,m}$ and the temperatures $T_{b,s}$ as well as $T_{b,h}$ of all simulations are shown in Table 12.6.

As intended, the MPC lowers the furnace temperatures to reduce the energy consumption. In the non-robust case, the temperatures are reduced too far and the MPC problem becomes infeasible. This is prevented by the usage of the quadratic invariance constraints. Hence, the furnace temperatures are reduced just enough to keep $T_{b,h}$ above the defined bound. This illustrates the effectiveness of the robust MPC formulations employing the RCIS.

12.4. Conclusions

In summary, the simulations show that the MPC approaches proposed for a single nonlinear JMS can be applied to real production units, such as the RHF. Both, reference tracking and economic setups, can be realized. However, the determination of RCIS for real systems is quite challenging and sometimes impossible. Reasonable relaxations are necessary to derive RCIS for the RHF. Once the RCIS are determined, the usage of the MPC approaches is straightforward. The control performance is nearly the same for both cost functions, but the computation time for the CoE MPC is considerably lower than for the EoC MPC. However, the absolute computation time is below 0.15 seconds for all configurations considered. This is low enough for an online implementation.

The robust MPC approaches perform well for both reference tracking and economic optimization. For reference tracking, the blank temperatures are stabilized

close to the reference and the variance of the blank temperatures is reduced significantly compared to a commonly used hysteresis controller. In the economic setup, the temperatures are lowered as much as possible considering the constraints. This minimizes the energy consumption in normal operation. In contrast, the non-robust formulations (without quadratic constraints) perform slightly better for reference tracking satisfying all constraints, but they fail completely in the economic setup. The temperatures are lowered too much and the MPC becomes infeasible. This illustrates the effectiveness of the designed RCIS despite the employed relaxations.

All in all, both cost function formulations can be employed due to the low computation times. If computational capacity is an issue, the CoE approaches should be employed since the performance difference is neglectable. For reference tracking, the non-robust MPCs are recommended due to better performance and lower computation times. However, if constraint satisfaction is crucial, the robust approaches may be employed resulting in a moderate performance degradation. For economic MPC setups or references that are close to the constraints, only the robust formulations should be employed.

13. Distributed MPC for a Hot Stamping Line

In the previous chapter, the effectiveness of the proposed MPC approaches for a single production unit has been demonstrated. However, the effects of couplings with other production units, of the plant model mismatch, and of the corresponding error models have not been considered. Hence, in this chapter, the interplay of MPCs in multistage production setups subject to model errors caused by, e.g., wear, are considered. The DMPC approaches, proposed in Chapter 11, are tested with the model of a complete hot stamping production line presented in Chapter 4. The effects of cooperation between the local controllers and the usage of error models are investigated in detail. First, the properties of the DMPC approaches are tested without plant model mismatch, but with the known uncertainties. Second, the approaches are tested with 48 different plant model mismatch configurations.

13.1. Problem Setup

In this chapter, a hot stamping production line consisting of two RHFs, a chamber furnace, a transfer robot, and a press is considered. For details on the models and the process, see Chapter 4. A cycle time of about t_c = 18 seconds is considered. The blanks are delivered alternatingly to the chamber furnace by the two RHFs, which are operated at double cycle time. A detailed timing diagram, including the processing times, measurement times, and MPC executions, is presented in Fig. 13.1 on page 186. The sampling instants of the subsystems, i.e., the times when the MPC procedures are executed, are indicated by bold bars. The green arrows indicate the times, when the processing is finished in one production unit, and the products are transferred to the subsequent production unit. In parallel, the predicted output trajectories $\bar{\boldsymbol{y}}^{\boldsymbol{n}}[0]$ are transmitted. The input trajectory $\boldsymbol{u}^{\boldsymbol{n}}[0]$, the state trajectory $\bar{\boldsymbol{x}}^{\boldsymbol{n}}[0]$, and the Markov state θ_k^n are transmitted to all predecessors (indicated by the red arrows), once the MPC procedure has been finished. The times, when the blank temperatures are measured, are indicated by blue squares. As soon as the measurements have been performed, the error models are updated, the disturbances are predicted, and the corresponding trajectories $\bar{\boldsymbol{w}}^{\boldsymbol{n}}[0]$ are transmitted to all predecessors (indicated by the blue arrows).

The timing diagram shows, that the blank temperatures can only be measured at three points: after the blanks leave the chamber furnace, after the blanks are inserted
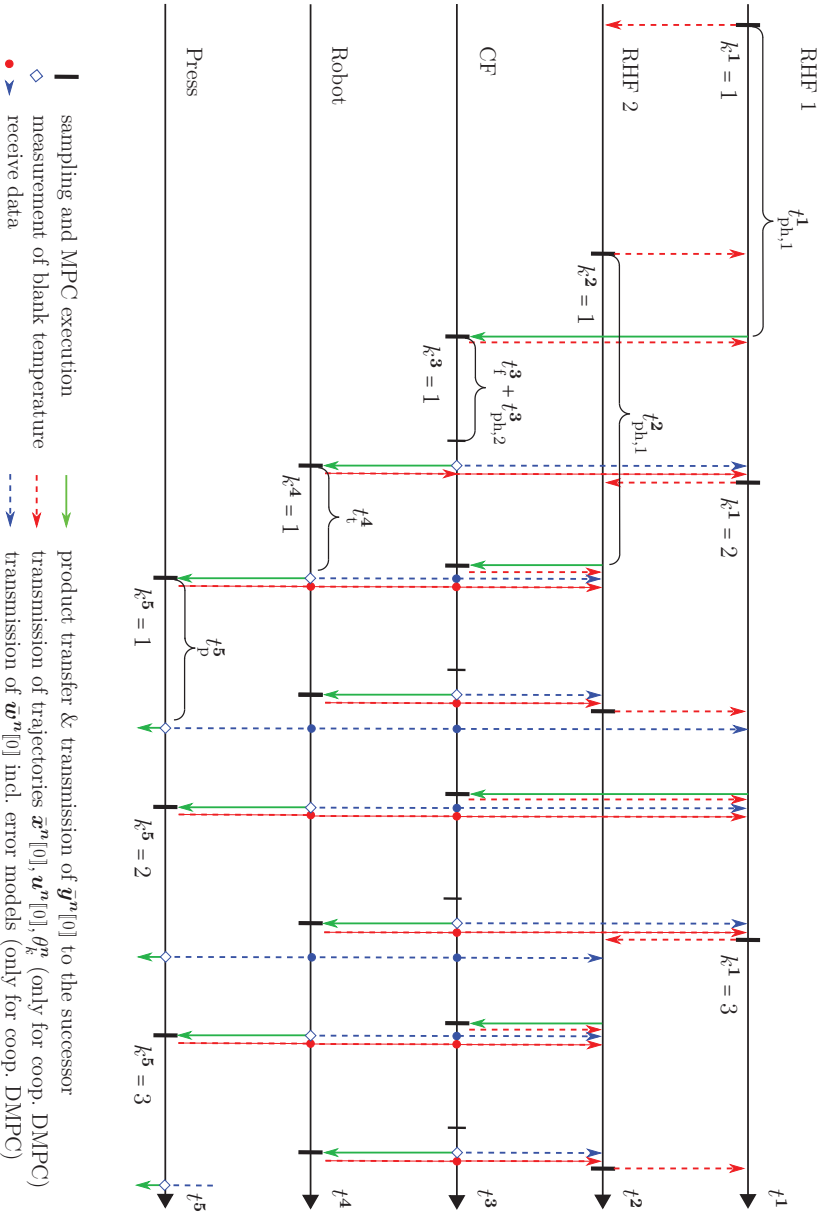
Figure 13.1.: Timing and communication diagram of the distributed control system for the hot stamping line.
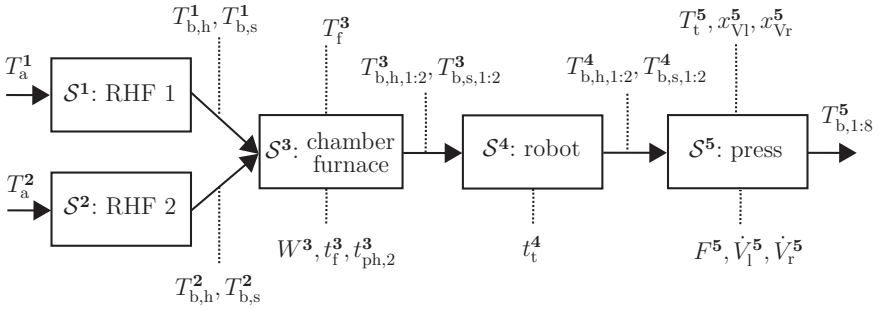
Figure 13.2.: Sketch of the hot stamping process and the localization of the corresponding quantities. The outputs are shown on the arrows, the states are shown over the blocks, and the inputs are shown below the blocks. For the sake of a clear presentation, the quantities of the RHFs and the probabilities p_{b}^{n} are omitted. For details on the RHFs, see Fig. 12.1.

to the press, and after the press has reopened. Hence, plant model mismatches for the blank temperature dynamics can only be detected for the chamber furnace, the robot, and the press. The timing diagram also illustrates the asynchronous sampling of the subsystems. It can be seen that there is a delay of more than two production cycles between the MPC execution of the first and the fifth subsystem. This illustrates the cause for the lags $l^{n,m}$ and their effects on the implementation and communication of a DMPC discussed in Section 11.2.3.

This demonstrates the complexity for an implementation of a DMPC caused by the asynchronous sampling of the subsystems. This DMPC setting is tested by simulations. The details on the parameterization of the local MPCs and the simulation setup are presented below.

Parameterization of the Local MPCs

For all subsystems, the augmented system description according to Section 5.2 is employed, i.e., the output vectors are appended to the state vectors. The cost weights, constraints, and references of both furnaces are defined as in Section 12.1. Only the input reference u_{r}^{2} for the second RHF is slightly different due to a different state of the thermal insulation. The values are shown in Table 13.1.

Table 13.1.: Input reference u_{r}^{2} for the second RHF (heating powers $W_{\mathrm{h},n}$ in kW, transportation velocity v^{-1} in s/m, and hardening time $t_{\mathrm{ph},1}$ in s).

$W_{\mathrm{h},1}$	$W_{\mathrm{h},2}$	$W_{\mathrm{h},3}$	$W_{\mathrm{h},4}$	$W_{\mathrm{h},5}$	$W_{\mathrm{h},6}$	$W_{\mathrm{h},7}$	$W_{\mathrm{h},8}$	$W_{\mathrm{h},9}$	v^{-1}	$t_{\mathrm{ph},1}$
110	95	65	55	50	50	15	5	50	16.7	25

Table 13.2.: Bounds, references, and cost weights for inputs and states of the chamber furnace, the robot, and the press (all temperatures $T_{(\cdot)}$ in °C, all times $t_{(\cdot)}$ in s, heating power W_h in kW, cooling agent flow rates $\dot{V}_{(\cdot)}$ and $x_{V(\cdot)}$ in $^1/_{\text{min}}$, force F in 10^3 kN, and probabilities $p_{(\cdot)}$ dimensionless).

inputs	chamber furnace				robot	press		
	W_h^3	t_f^3	$t_{\text{ph},2}^3$	$t_f^3 + t_{\text{ph},2}^3$	t_t^4	$\dot{V}_l^5$	$\dot{V}_r^5$	F^5
max.	90	8	4	9	10	20	20	1
u_r	38	6	2.5	-	9	15	15	0.85
min.	0	5	2	-	7	0	0	0.6
$R_{l,l}$	1	1	1	-	50	10	10	1000

states	chamber furnace				robot			press			
	T_f^3	$T_{\text{b,h}}^3$	$T_{\text{b,s}}^3$	p_b^3	$T_{\text{b,h}}^4$	$T_{\text{b,s}}^4$	p_b^4	$T_{\text{t},:}^5$	$x_{\text{Vl}}^5, x_{\text{Vr}}^5$	$T_{\text{b,1,2,5,6}}^5$	$T_{\text{b,3,4,7,8}}^5$
max.	980	980	730	-	-	670	-	200	20	230	
x_r	930	920	650	-	760	610	-	40	15	170	130
min.	900	900	-	-	730	-	-	-	0	-	
$Q_{l,l}$	5	50	50	0	50	50	0	0.3	0	3	

The references, bounds, and cost weights for the chamber furnace, the transfer robot, and the press are shown in Table 13.2. The localization of the corresponding quantities is shown in Fig. 13.2. The weights of the local cost functions γ^n are set to one. Note that the bounds and references for the press are that of a test tool. Here, the pressure force is normalized. The input bounds of the chamber furnace and the transfer robot result from technical limitations. The pressure force F and the transfer time t_t have relatively high cost weights, since these inputs should only deviate from the reference if necessary.

The references and bounds for the blank temperatures of the chamber furnace are defined such that a martensitic phase transition is induced in the hard part of the blanks ($T_{\text{b,h}}$) and is prevented in the soft part of the blanks ($T_{\text{b,s}}$). To this end, the A_{c1} temperature of the base material (about 750 °C [98]) minus 20 °C safety margin is used; see page 13 for details.

To guarantee that the phase transformations are finished after opening the pressing tool, the blank temperatures have to be below the martensite finish temperature M_f = 230 °C [97]. Since the blank temperatures have a main influence on the product properties, high weights are used for the corresponding states.

Analogously to the RHF, the blank temperatures are set to zero in all subsystems, if no blanks are processed. Hence, the issues described for the RHF in Section 12.1 arises for all subsystems. Therefore, the corresponding constraints and reference values for the CoE approach are formulated analogously to (12.1) and (12.2).

Remark 13.1. *The subsystems are only coupled if products are processed. Hence, the costs in the downstream production units cannot be influenced if no products are processed. In this case ($\theta_k^n = 2$), only the local costs are minimized.*

Initialization of the Subsystems

All subsystems are initialized to normal operation ($\theta_0^n = 1$ for all $n \in \{1, \ldots, 5\}$). The RHFs are fully loaded with blanks at the beginning of each simulation. The blank temperatures are initialized randomly with values from the interval ±10 °C around the reference. All furnace temperatures (also for the chamber furnace) are initialized randomly with values from the interval ±15 °C around the reference. The pressing tool is at room temperature (25 °C) at the beginning of the simulation. The cooling agent flows are set to 90 % of the maximum value. The robot has no states that have to be initialized.

Disturbance Processes

The disturbance $w^n[k]$ comprises the product-related disturbances $w_p^n[k]$ and the local disturbances $w_d^n[k]$. The local disturbances also contain the plant model mismatch processes $\nu^n[k]$. For the RHFs, the model errors of the furnace temperatures $\nu_T[k]$ are defined as truncated normal distributions (cf. Section 12.1). The remaining disturbances have the following distributions:

$$
\begin{aligned}
w_d^1[k] &= \begin{bmatrix} T_a^1[k] & w_v^1[k] & T_e^1[k] \end{bmatrix} \sim \mathcal{N}\left([25 \ \ 0.05 \ \ 50]^\top, \mathrm{diag}(0.1, 0.1, 0.1)\right), \\
w_d^2[k] &= \begin{bmatrix} T_a^2[k] & w_v^2[k] & T_e^2[k] \end{bmatrix} \sim \mathcal{N}\left([25 \ \ 0.05 \ \ 50]^\top, \mathrm{diag}(0.1, 0.1, 0.1)\right), \\
w_d^3[k] &= \begin{bmatrix} T_a^3[k] & \nu_T^3[k] & \nu_y^3[k] \end{bmatrix} \sim \mathcal{N}\left([75 \ \ 0 \ \ 0 \ \ 0 \ \ 0]^\top, \mathrm{diag}(0.1, 0.8^2, I_4)\right), \quad (13.1) \\
w_d^4[k] &= \begin{bmatrix} T_a^4[k] & \nu_y^4[k] \end{bmatrix} \sim \mathcal{N}\left([40 \ \ 0 \ \ 0 \ \ 0 \ \ 0]^\top, \mathrm{diag}(0.1, 4I_4)\right), \\
w_d^5[k] &= \begin{bmatrix} T_a^5[k] & T_c^5[k] & \nu_x^5[k] & \nu_y^5[k] \end{bmatrix} \sim \mathcal{N}\left([25 \ \ 27 \ \ \mathbf{0}_{1\times24}]^\top, \mathrm{diag}(0.1, 0.1, I_{16}, 4I_8)\right).
\end{aligned}
$$

The plant model mismatch processes of the transfer robot and the press are bounded to ±5 °C. For the measurable blank temperatures (after chamber furnace, transfer robot, and press) and for the temperatures of the pressing tool, second order ARIMA models described in Sections 3.2.2 and 4.2 are employed to predict the expected value of the disturbances $\bar{w}^n[\![j]\!]$.

RCIS Design

The difficulties to design RCIS for the RHF have been described in Section 12.1. Since the design approach for RCIS for a distributed setup is more conservative than the RCIS design for one subsystem, it is not possible to determine RCIS according to Section 11.2. However, the simulations in Chapter 11 demonstrated that a DMPC setup in combination with independently determined RCIS may also guarantee recursive feasibility. Hence, for the control of the hot stamping line, independently determined RCIS are employed:

- The RCIS determined in Chapter 12 are used for both RHFs.

- The RCIS design presented in Chapter 12 also applies to the chamber furnace, since it is a small RHF with just one temperature segment. As the only state of the chamber furnace is the furnace temperature T_f, the resulting RCIS is a box-constraint for the furnace temperature. This constraint is nearly the same as the tightened state constraint for the furnace temperature T_f. Thus, the use of additional invariance constraints is not necessary for the chamber furnace.

- The JMS of the transfer robot has no states, i.e., no RCIS is needed.

- For the press, the calculation of a RCIS is possible employing similar relaxations as for the RHF. However, the resulting sets are relatively small resulting in a very conservative formulation of the MPC. This results in a significant degradation of the control performance. Hence, no RCIS are used for the MPC of the press.

To allow the subsystems to deviate from their reference in order to minimize the overall cost criteria, no stability (convergence) constraints are used. In order to guarantee state constraint satisfaction, the state constraints of all subsystems are tightened with the local error sets. To this end, the one-step prediction error, caused by model uncertainties, linearization errors, and disturbances, is determined by Monte Carlo simulations. The resulting error sets are not shown, since they do not provide a significant insight into the underlying problem. Note that these error sets do not consider the persistent plant model mismatches employed in the simulations presented later in this chapter.

13.2. Simulation Results

To demonstrate the effects of the error models and the differences between the DMPC approaches, three MPC configurations are considered:

1. Non-cooperative DMPC without disturbance models.

2. Non-cooperative DMPC as in 1., but with disturbance models for the prediction of the plant model mismatch.

3. Cooperative DMPC with disturbance models.

In all simulations the states are assumed to be measurable. The prediction horizons of the MPCs of both RHFs are $N^1 = N^2 = 11$, since a blank pair needs 10 time steps to pass the furnace completely. In addition, the simulations in Chapter 12 showed that a larger horizon length is not necessary. The horizons of the remaining MPCs are $N^3 = N^4 = N^5 = 15$. Due to the different prediction horizon lengths and the lags between the MPC executions (see Fig. 13.1), not all quantities needed for

the implementation of the DMPC are available. The missing elements at the end of the trajectories are filled up with a copy of the last known element, i.e., the trajectories are assumed to be constant for the last time steps. The selected horizon lengths establish a good compromise between computation time and the amount of unavailable information.

The different configurations are simulated 100 times with both the CoE and EoC cost function. In each simulation, 30 production cycles of the RHFs and 60 production cycles of the remaining subsystems are considered. In addition, the system is simulated with the currently used control setup: hysteresis controllers are used for the furnace temperatures (heating powers). The remaining input values are set to the corresponding reference. This setup is denoted as *reference controller*. The average costs for all simulation runs are shown in Table 13.3. The results document that all MPC approaches result in similar costs and perform significantly better than the reference controller. The costs for the EoC approaches are nearly the same as for the CoE approaches. While the costs for the different configurations are similar, it is unexpected that the performance decreases from configuration one to three. This is caused by the fact that the plant model mismatch processes are noise processes with zero mean. Hence, the optimal prediction is a zero trajectory. The second order ARIMA models, however, predict values different from zero based on the noisy signals. The cooperative MPC tries to counteract these wrong error predictions by cooperative actions and degrades the performance slightly. If the error models are not used, the cooperative DMPC is marginally better than the non-cooperative DMPC. The effect is quite small, since the whole process can be stabilized at the reference if each subsystem keeps its state close to the reference. This does not require cooperation between the subsystems. However, if persistent disturbances affect the subsystems, the results are different. This is illustrated later in this section.

The MPC problems have been feasible and the constraints have been satisfied at all times. For the reference controller, the upper bound of the furnace time $t_{\mathrm{f}}^{\mathbf{2}}$ of the

Table 13.3.: Comparison of average costs in the subsystems for the different architectures.

cost function	EoC			CoE			
architecture	non-coop.		coop.	non-coop.		coop.	ref.
distur. model	no	yes		no	yes		
RHF 1	1.44	1.44	1.48	1.45	1.45	1.49	1.84
RHF 2	1.42	1.42	1.45	1.42	1.42	1.46	1.88
CF	0.43	0.45	0.48	0.44	0.46	0.47	4.27
robot	0.33	0.40	0.39	0.33	0.40	0.40	1.10
press	1.35	1.35	1.35	1.35	1.35	1.35	3.08
Σ	4.97	5.06	5.15	4.99	5.08	5.17	12.17

Table 13.4.: Computation times in ms of the local MPCs in the different architectures.

cost function	EoC		CoE	
architecture	non-coop.	coop.	non-coop.	coop.
RHF 1	107	305	69	89
RHF 2	106	303	67	92
CF	34	95	8	19
robot	29	66	6	13
press	47	47	12	12

second RHF is violated slightly. Due to the large number of states, it is impractical to plot all simulation results. Hence, the results are illustrated with one example – the blank temperature in the hard part $T_{\mathrm{b,h}}^{n}[k]$. The results of the cooperative DMPC with EoC cost function and the reference controller[1] are shown in Fig. 13.3. It can bee seen that the mean values for the furnaces and the robot are close to the references for both the DMPC and the reference controller. The resulting envelopes (and variances) are similar for the robot. In contrast, the envelopes of the states of the furnaces are significantly smaller for the DMPC setups. For the press, a substantial difference between the reference controller and the DMPC results. This is most likely caused by the fact that the influence of the production line stops on the bank temperatures is larger than for the other subsystems. The mean temperature deviates about 10 °C from its reference for the reference controller, while the difference is about 3 °C for the DMPC controller. In addition, the variance is reduced. An offset to the reference cannot be avoided completely since the press is underactuated (three inputs for 16 tool and 8 blank temperatures). However, the offsets are reduced significantly by the DMPC.

The average computation times of the local MPCs for the DMPC architectures[2] are shown in Table 13.4. As for the previous simulations, the computation times of the EoC approaches are significantly larger than the computation times for the CoE approaches (3.7 times in average). For all but the last subsystem, the computation time of the cooperative DMPC is larger than that of the non-cooperative DMPC. This is caused by the calculation of the prediction matrices for the costs in downstream subsystems. The additional computation time decreases from subsystem to subsystem, since the prediction matrices have to be calculated for all downstream processes. This effect is much larger for the EoC cost function. However, the overall computation times are still below 0.4 seconds for all subsystems. This is low enough

---

[1]The other DMPC configurations are not shown, since the plots are very similar. The second RHF is not shown, since the results are similar to the first one.

[2]There is no differentiation whether the disturbance models are used or not, since the disturbance prediction is performed independently of the MPC execution and consumes very little computation time.
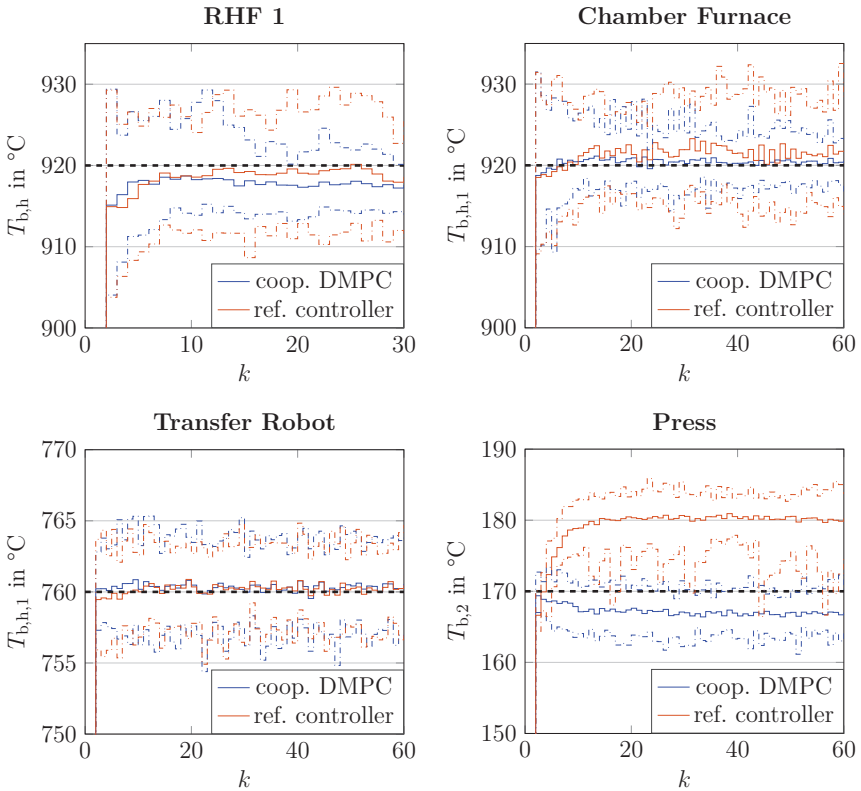
Figure 13.3.: Mean values and envelopes of the blank temperatures $T_{b,h}^n[k]$ for the first RHF, the chamber furnace, the transfer robot, and the press controlled by the cooperative DMPC and the reference controller. The reference values are shown by the dashed lines.

for an implementation on the real production line.

All in all, the simulations show that both non-cooperative and cooperative DMPC approaches can be used for the control of the whole production line. The control performance in terms of reference tracking and variation of the part properties is increased compared to the reference controller. The disturbance models do not increase the control performance for the considered setup. Instead, the performance slightly degrades. However, the question remains if this is also true if persistent unknown disturbances (plant model mismatch) affect the subsystems. This is the case, e.g., if the burners or the tools wear out, or if the cooling power deteriorates.
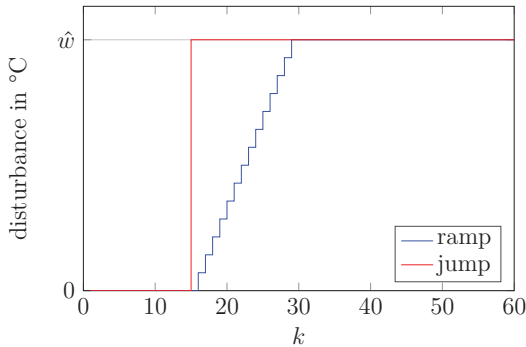
Figure 13.4.: Trajectories of the additive persistent disturbances.

This issue is considered in detail by 48 simulations with different persistent disturbances affecting the subsystems. Each simulation has the same setup as for the simulation presented above, but one or more output or state equations are subject to an additional disturbance. The disturbance is unknown to the MPCs. To simulate both continuous effects, such as wear, or abrupt effects, such as burner defects, both ramp and jump disturbances are considered[3]. Here, ramps from zero to an amplitude $\hat{w}$ between the 15th and the 30th production cycle and jumps to the same amplitude $\hat{w}$ at the 15th production cycle are employed. The disturbances are illustrated in Fig. 13.4. In Table 13.5, the disturbance parameterizations for the 48 simulations are shown. It shows which quantities are affected by the disturbance, which type of disturbance is considered (ramp or jump), and the amplitude of the disturbance. For example, in the first simulation, an abrupt disturbance with a temperature jump of 15 °C is added to both blank temperatures of the first RHF. In the first 44 simulations, various disturbance setups affecting the blank temperatures or the tool temperatures of one subsystem are considered. The last four simulations consider persistent disturbances in two subsystems. For the sake of brevity, only disturbances affecting the blank temperatures and tool temperatures are considered. The amplitude is set to 15 °C which is a value possibly encountered in the serial process and which can still be handled considering the constraints.

Remark 13.2. *The nonlinearities (the kinks and jumps) in these disturbance definitions can not be modeled exactly by the linear ARIMA models. In combination with the online adaption of the parameters, this may lead to a temporary instability of the disturbance models. This is in particular the case for the abrupt disturban-*

---

[3]In theory, the abrupt disturbances can be also considered by additional Markov states. However, the sheer number of disturbance combinations (arising, for example, from the possible defects of 64 different burners in one RHF) renders this impossible. Hence, only the significant disturbances can be considered by different Markov states.

ces directly after the jump has occurred. This effect may result in significantly too large values of the predicted plant model mismatch $\breve{\nu}[\![j]\!]$ and problems in the MPC execution. Hence, the predicted values of the plant model mismatch are bounded to realistic values that are not exceeded in the real production process (± 20 °C) in order to guarantee that the disturbance predictions are bounded. This does not impair the robustness guarantees of the approach since the true disturbances will be within the used bounds. In addition, this effect occurs only in very few time steps after the jump. Thus, the control performance is influenced only in very few time steps and the effect has only a small negative effect on the overall control performance (as can be seen from the simulation results).

Table 13.5.: Disturbance parameterizations of the different simulations.

no.	aff. quantities	type	ampl. $\hat{w}$	no.	dist. subj.	type	ampl. $\hat{w}$
1	$T_{b,h}^{1}, T_{b,s}^{1}$	jump	15 °C	25	$T_{b,h,1:2}^{3}$	jump	15 °C
2	$T_{b,h}^{1}, T_{b,s}^{1}$	ramp	15 °C	26	$T_{b,h,1:2}^{3}$	ramp	15 °C
3	$T_{b,h}^{1}, T_{b,s}^{1}$	jump	-15 °C	27	$T_{b,h,1:2}^{3}$	jump	-15 °C
4	$T_{b,h}^{1}, T_{b,s}^{1}$	ramp	-15 °C	28	$T_{b,h,1:2}^{3}$	ramp	-15 °C
5	$T_{b,s}^{1}$	jump	15 °C	29	$T_{b,h,1}^{3}, T_{b,s,1}^{3}$	jump	15 °C
6	$T_{b,s}^{1}$	ramp	15 °C	30	$T_{b,h,1}^{3}, T_{b,s,1}^{3}$	ramp	15 °C
7	$T_{b,s}^{1}$	jump	-15 °C	31	$T_{b,h,2}^{3}, T_{b,s,2}^{3}$	jump	-15 °C
8	$T_{b,s}^{1}$	ramp	-15 °C	32	$T_{b,h,2}^{3}, T_{b,s,2}^{3}$	ramp	-15 °C
9	$T_{b,h}^{1}$	jump	15 °C	33	$T_{b,h,1:2}^{4}, T_{b,s,1:2}^{4}$	jump	15 °C
10	$T_{b,h}^{1}$	ramp	15 °C	34	$T_{b,h,1:2}^{4}, T_{b,s,1:2}^{4}$	ramp	15 °C
11	$T_{b,h}^{1}$	jump	-15 °C	35	$T_{b,h,1:2}^{4}, T_{b,s,1:2}^{4}$	jump	-15 °C
12	$T_{b,h}^{1}$	ramp	-15 °C	36	$T_{b,h,1:2}^{4}, T_{b,s,1:2}^{4}$	ramp	-15 °C
13	$T_{b,h}^{2}, T_{b,s}^{2}$	ramp	15 °C	37	$T_{t,1:16}^{5}$	ramp	5 °C
14	$T_{b,h}^{2}, T_{b,s}^{2}$	jump	-15 °C	38	$T_{t,1:8}^{5}$	ramp	5 °C
15	$T_{b,s}^{2}$	jump	15 °C	39	$T_{t,9:12}^{5}$	ramp	5 °C
16	$T_{b,h}^{2}$	ramp	-15 °C	40	$T_{b,1:8}^{5}$	ramp	15 °C
17	$T_{b,h,1:2}^{3}, T_{b,s,1:2}^{3}$	jump	15 °C	41	$T_{b,1:4}^{5}$	ramp	15 °C
18	$T_{b,h,1:2}^{3}, T_{b,s,1:2}^{3}$	ramp	15 °C	42	$T_{b,1}^{5}$	ramp	15 °C
19	$T_{b,h,1:2}^{3}, T_{b,s,1:2}^{3}$	jump	-15 °C	43	$T_{b,3}^{5}$	ramp	15 °C
20	$T_{b,h,1:2}^{3}, T_{b,s,1:2}^{3}$	ramp	-15 °C	44	$T_{b,2:3}^{5}, T_{b,6:7}^{5}$	ramp	15 °C
21	$T_{b,s,1:2}^{3}$	jump	15 °C	45	no. 2 & no. 15		
22	$T_{b,s,1:2}^{3}$	ramp	15 °C	46	no. 2 & no. 22		
23	$T_{b,s,1:2}^{3}$	jump	-15 °C	47	no. 22 & no. 36		
24	$T_{b,s,1:2}^{3}$	ramp	-15 °C	48	no. 10 & no. 38		

Table 13.6.: Comparison of average costs, number of infeasible problems, and simulation runs with constraint violations for all 48 simulations.

cost function	EoC			CoE			ref.
architecture	non-coop.		coop.	non-coop.		coop.	
distur. model	no	yes		no	yes		
avg. costs J in $\cdot 10^5$	13.31	8.30	6.98	13.32	8.31	6.97	21.79
infeas probs.	0	0	0	0	0	0	-
cons. viol.	9	6	4	9	6	4	74

For the sake of brevity, not all simulation results are discussed in detail. In this section, an overview over the results is provided and some examples are shown to illustrate the effects of the different control setups. A complete list of the resulting average costs of all simulations can be found in Appendix D. The average costs for the different MPC configurations as well as the reference controller for all 48 simulations are shown in Table 13.6. In addition, the number of infeasible problems and the number of subsystem simulations with constraint violations are shown. The computation times are very similar to that in Table 13.4 and are, hence, not shown.

Again, the results are almost the same for both cost functions. However, the results of the different MPC configurations are different. The usage of the disturbance models decreases the costs of the non-cooperative DMPC approaches by 38 %. The cooperative DMPC reduces the costs further by 16 %. This result is completely different from that of the first simulation presented in this section. The disturbance models improve the state prediction significantly, if persistent disturbances are present. The effect of the disturbances can be reduced further if the subsystems cooperate. This is illustrated by some examples later. In addition, the costs of the reference controller have increased considerably.

The optimization problems have been feasible at all times. The constraints are violated in some simulations, since only the local disturbances and linearization errors (and not the plant model mismatches $\nu[k]$ specified in Table 13.5) have been considered for the state constraint tightening. Hence, this problem only occurs after significant plant model mismatch processes start to affect the system. As soon as the disturbance models describe the plant model mismatch accurately, no violations occur anymore. The number of constraint violations is reduced by employing the disturbance models and cooperation. This also illustrates the positive influence of the disturbance models and the cooperation on the control performance. By comparing these results with the reference controller, the advantages of the cooperative DMPC become obvious. The number of constraints violations is reduced by 95 %. These constraint violations do not necessarily lead to a violation of product specifications but they result in an increased variance of the product properties.

The effects of the cooperation are illustrated with two examples. First, the simulation number 1 is considered. The mean values of the blank temperatures $T_{b,h}^n[k]$
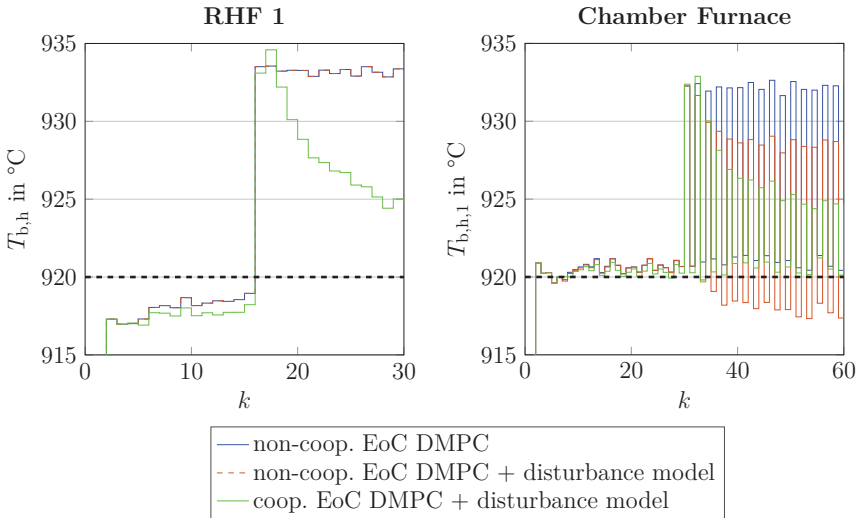
Figure 13.5.: Mean values of the blank temperatures $T_{b,h}^1[k]$ and $T_{b,h,1}^3[k]$ for the first RHF and the chamber furnace for the EoC DMPCs and the first simulation. The reference values are shown by the dashed lines.

in the hard part after the first RHF and after the chamber furnace are shown in Fig. 13.5. Only the EoC DMPCs are considered. The influence of the abrupt disturbance can be seen: The blank temperature of the RHF increases by 15 °C at $k^1 = 15$. This results in an increased temperature after the chamber furnace for every second part. Since the blank temperatures cannot be measured after the RHF, both non-cooperative DMPC configurations do not detect the disturbance and control the RHF as if there where no disturbances. After the chamber furnace, the blank temperature is measured and the disturbance model predicts the plant model mismatch resulting from the temperature of the incoming blanks. Thus, the MPC is able to reduce the effect of the disturbance by reducing the furnace temperature, but the influence of the disturbance cannot be suppressed completely. In the cooperative DMPC setup, the local MPC of the RHF tries to minimize the costs in the own subsystem and the downstream subsystems. Hence, the local MPC reduces the furnace temperatures of the RHF in order to reduce the blank temperature after the chamber furnace. This can be seen from the decrease in the blank temperature of the RHF. It reduces the tracking error in the RHF and the chamber furnace. The larger the cost weight γ^3 of the chamber furnace is, the stronger the RHF reducers the furnace temperatures. In this setup, the cooperation of the local MPCs compensates the missing measurement of the blank temperatures in the RHF. This reduces the overall costs by 23 %. The whole procedure does not affect the second RHF,
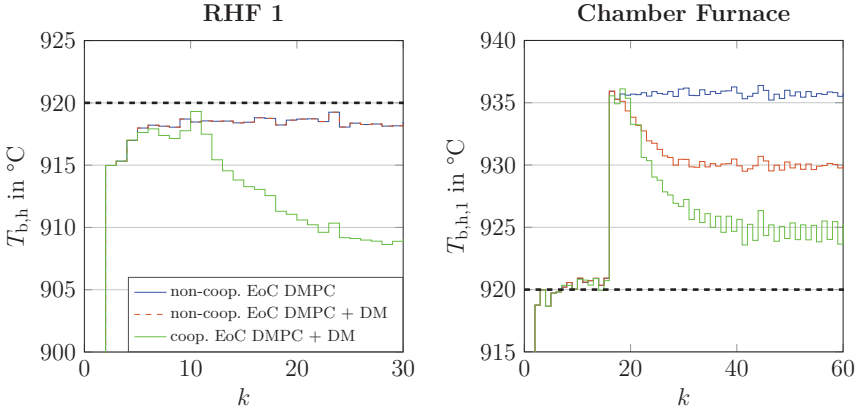
Figure 13.6.: Mean values of the blank temperatures $T_{\mathrm{b,h}}^{\mathbf{1}}[k]$ and $T_{\mathrm{b,h,1}}^{\mathbf{3}}[k]$ for the first RHF and the chamber furnace for the EoC DMPCs and the 17th simulation. The reference values are shown by the dashed lines.

since the source of the disturbances is the first RHF.

The second example considers simulation number 17, where an additive abrupt disturbance affects the blank temperatures of the chamber furnace. The results are shown in Fig. 13.6. The influence of the disturbance can be seen in the plot of the blank temperature of the chamber furnace. The behavior is similar to the example considered previously: The DMPC setup without disturbance models does not react at all on the disturbance. In the non-cooperative DMPC setup with disturbance models, the local MPC of the chamber furnace reduces the furnace temperature to counteract the disturbance. However, due to the short processing time of the chamber furnace, the impact on the final blank temperature is moderate and a significant offset from the reference remains. In the cooperative DMPC setup, the temperatures of the RHF are reduced, i.e., the MPC deviates from its own reference in order to "help" the chamber furnace to reduce the effect of the disturbance. The MPC of the second RHF reacts in the same way. This cooperation reduces the overall costs by 37 %. This effect is even stronger if no invariance constraints are used by the MPCs of the RHFs. In this case, the furnace temperatures of the RHFs are reduced further, such that the disturbance is compensated almost completely. This results from the large cost weights for the blank temperatures $T_{\mathrm{b,h}}^{\mathbf{3}}[k]$ and $T_{\mathrm{b,s}}^{\mathbf{3}}[k]$ of the chamber furnace. While this reduces the overall costs, the furnace temperatures of the RHF are reduced too much (as in the economic MPC setup considered in the previous section). In consequence, the optimization problems of the RHFs become infeasible sometimes. Hence, the cooperative DMPC setup should be used with invariance constraints for the RHFs.

The two examples illustrate the positive effects of the disturbance models and the cooperation of the local MPCs. The impact of the cooperation depends on the cost parameterization and the possibility to influence the affected quantities. For example, the influence of the furnace temperatures or the transportation time on the tool temperature of the press and the final blank temperatures is very low. Hence, for disturbances affecting the press, a cooperative DMPC does not achieve a better performance than a non-cooperative MPC. The resulting costs behave as for the nominal case simulated in the beginning of this chapter. The blank temperature in the soft part of the blank can be influenced by the partial hardening time much more than the temperature in the hard parts by the furnace temperature. Hence, disturbances affecting $T_{\mathrm{b,s}}^{\boldsymbol{n}}[k]$ can be compensated more effectively. For the examples 45 - 48 (considering persistent disturbances in two subsystems), the results are basically an accumulation of the effects for the single disturbance. If the cooperation does not improve the performance for the setups with the single disturbances, it does not improve the performance for the combination of the disturbances (see simulation 48), and vice versa (see simulation 46). All in all, whether cooperation improves the overall control performance strongly depends on where a disturbance / plant model mismatch occurs. Especially for disturbances in the furnaces, cooperation improves the performance significantly. In none of the simulations, the performance was degraded significantly by the use of the disturbance models or the cooperation. Hence, for the control of the production line, a cooperative DMPC with disturbance models should be used.

13.3. Summarizing Assessment

For the design of the DMPC, two main challenges arise: The timing and communication diagram (Fig. 13.1) illustrated the complexity of the implementation of an cooperative DMPC due to the asynchronous sampling of the subsystems. In addition, the design of RCIS for a DMPC setup according to Section 11.2.1 is too conservative for the production line under consideration. Hence, the RCIS have to be determined independently for the subsystems. For the press, even these sets are too restrictive. Hence, invariance constraints are only considered for the furnaces. This illustrates the main shortcoming of the proposed approaches – the conservatism of the offline part, i.e., of the computation of RCIS.

Despite the difficulties in the offline part, the simulations without plant model mismatch show that the performance of the resulting DMPC setup is good. The states are stabilized close to the references by all MPC setups and the constraints are satisfied at all times. For all DMPC setups, the costs are about 58 % lower than for the reference control used in current production lines. However, there is no improvement of the performance achieved by the cooperative DMPC and the error models compared to the simple non-cooperative DMPC, since the optimal error prediction is zero and cooperation is not needed to stabilize the system at the

reference. If persistent plant model mismatches occur, the results are completely different. This was illustrated by 48 simulations with different persistent plant model mismatches. The costs are reduced by 38 % in average by using the error models. The costs are reduced further by 16 % if a cooperative DMPC is used instead of the non-cooperative formulation. The cooperative DMPC reduces the costs by 68 % compared to the reference control. More importantly, the number of constraint violations was reduced by 95 %. The remaining constraint violations are due to the fact that the plant model mismatch has not been considered in the constraint tightening. In addition, all MPC problems have been feasible at all times if the RCIS are used. If they are not used, some simulations have resulted in infeasible MPC problems. Hence, the RCIS are needed for the DMPC setup, but it is sufficient to design the RCIS independently. The average computation times of the MPCs have been below 0.4 seconds, which is low enough for an online implementation. Since the computation times of the EoC MPCs are about 3.7 times larger than for the CoE MPCs, the CoE cost function should be used.

All in all, the simulations show the effectiveness of the DMPC approaches. To control the production line, a robust cooperative DMPC employing error models and the CoE cost function is recommended. This guarantees a robust operation of the production line according to the given specifications. In addition, the costs are reduced significantly compared to the reference control.

Part VI.

Conclusions and Outlook

14. Conclusions and Outlook

Industrial manufacturing processes are often run with a set of constant parameters although continuous and abrupt disturbances influence the process. To reduce quality variations, scrape rate, and resource consumption resulting from this operation, a (robust) closed-loop control of the process variables becomes more and more important. To this end, a modeling and control approach for large scale manufacturing systems that are subject to abrupt faults and continuous disturbances has been presented in this thesis. In this chapter, the approaches and results are summarized and discussed. Finally, an outlook on future research topics is presented.

14.1. Summary and Discussion of the Results

The literature review has revealed that there is a lack of a modeling and control framework that focuses on the properties of the products considering abrupt faults or disturbances of the manufacturing system (and their probabilities of occurrence), such as component defects. To fill this gap, an approach combining the modeling of the production units by jump Markov systems (JMS) with concepts from the stream of variation framework and robust distributed model predictive control (DMPC) has been presented. The whole procedure has been applied to the model of an industrial hot stamping line.

Modeling Setup

Large scale manufacturing systems consist of several production units (subsystems). To account for abrupt disturbances, each production unit is modeled by a JMS, in which the Markov chain models the occurrences of faults (cf. Section 3.2). The corresponding transition probabilities are calculated based on fault histograms. The process variables are regarded as inputs of the continuous dynamics. The properties of the incoming products are considered as disturbances, and the outputs describe the properties of the processed products. A plant model mismatch, e.g., caused by tool wear, is modeled by ARIMA models for which the parameters are estimated online. The production units are coupled by the properties of the processed products, i.e., the output of one subsystem defines the disturbances of the downstream subsystems (cf. Section 3.3). By this approach, the propagation of deviations through the process is modeled (similar to concepts from the stream of variation approach [60]). A special feature of the modeling framework proposed is that the subsystems are

sampled when products are received for processing, i.e., when the controller is supposed to determine the process variables for the processing of this product. This results in an asynchronous sampling of the subsystems. This modeling framework can be used to describe the dynamics of the product properties as a function of the process variables considering possible abrupt faults and continuous disturbances. This is crucial for the design and implementation of a robust MPC. Since general nonlinear dynamics and time-variant transition probabilities are considered, a broad class of manufacturing processes can be modeled by this framework.

Control Approaches

To control the manufacturing system, different robust MPC setups for coupled JMS have been proposed in this thesis. First, the control of a single production unit has been considered, and then extended to a distributed MPC architecture. The different concepts are discussed and compared in this section.

Existing approaches for MPC of JMS and DMPC of distributed JMS are characterized by high computation times that render an application to large scale manufacturing systems impossible. In addition, general reference trajectories and time-variant behavior are not considered (cf. Sections 5.1 and 10.1). Hence, in a first step, efficient MPC formulations for time-variant JMLS that consider general linear constraints and reference trajectories have been developed (cf. Chapter 6). Two different probabilistic cost function formulations have been considered: a cost function employing the expected value of quadratic state and input costs (EoC), which is commonly used in literature, and a cost function defining quadratic costs on the expected values of the state and input trajectories (CoE). The EoC cost function can be regarded as an extension of the CoE cost function that also punishes the variance of the deviation from the reference $\mathrm{Var}(x[\![j]\!] - x_{\mathrm{r}}[\![j]\!])$ (cf. Theorem 5.1). For both cost functions, efficient recursive algorithms have been developed that can be used to derive quadratic programming formulations for the MPC (Sections 6.1 and 6.2). By these algorithms, the exponential complexity for the cost prediction is reduced to a low polynomial complexity. A comparison of the computation times demonstrates the effectiveness of the algorithms (Section 6.3): for 100 states, 30 inputs, and 15 Markov states, the computation time of the EoC-based MPC is about 580 ms and 250 ms for the CoE-based MPC. Both computation times are several magnitudes lower than for common semidefinite programming-based approaches. The higher effort of the EoC approach, however, may lead to a better control performance. In the majority of the simulations, the EoC-based MPC results in lower costs. For both approaches, the performance and stability of the controlled system depends strongly on the cost parameterization and horizon length. With increasing horizon length, the resulting costs decrease until a minimum is reached. Then, the costs start to increase. This is due to the fact that the uncertainties of the prediction increase with the horizon length. When a certain horizon length is reached, the uncertainties begin to dominate the costs. Hence, for many systems, large horizons

are not needed. This effect is stronger for the EoC approach since the variances of the states are considered in the cost function. All in all, no general statement is possible on which approach performs better and on whether the system is stabilized or not. Closed-loop stability depends on the MPC parameterization.

This is insufficient for the robust control of manufacturing systems. Hence, a robust MPC for JMLS that guarantees closed-loop stability (independent of the parameterization of the cost function) has been proposed in Chapter 7. To this end, ellipsoidal robust control invariant sets (RCIS) and a quadratic Lyapunov function are determined offline by solving an semidefinite program. In parallel, a linear state-feedback law is determined offline that keeps the state of the JMLS in the RCIS, stabilizes the JMLS, and satisfies the constraints. To guarantee the same properties for the MPC, quadratic constraints, that guarantee that the state of the JMLS stays in the RCIS and that the value of the Lyapunov function decreases, are added to the quadratic programming. In addition, the state constraints are tightened with the disturbance set. The resulting quadratically constrained quadratic program (QCQP) guarantees closed-loop stability and robust constraint satisfaction. Due to the existence of the auxiliary control law, the optimization problem is recursively feasible. The big advantage of this design approach is that the complex computations (the design of the RCIS) are performed offline. The computational effort for solving the QCQP is relatively low.

The whole procedure can be applied to time-variant JMLS. To this end, the system matrices are over-approximated with matrix polytopes. The RCIS and the Lyapunov function are computed employing concepts from linear parameter varying systems. The formulation of the MPC problem as a QCQP is the same as for the time-invariant case. The conservatism of the RCIS design, however, is larger than for the time-invariant case due to the polytopic over-approximation of the system. Hence, the volume of the RCIS and the feasible set of the MPC may be considerably smaller. However, compared to existing approaches, the conservatism is reduced by allowing the controllers to be time-variant for the design of the RCIS.

This framework can be applied also to nonlinear JMS employing online linearizations (cf. Chapter 8). For this purpose, an ellipsoidal over-approximation of the linearization errors and the disturbances is determined offline. Employing this error set and polytopic linear differential inclusions describing the nonlinear dynamics, the RCIS and a Lyapunov function can be determined as for time-variant JMLS. Using the linearized JMS, the robust MPC can be formulated as for time-variant JMLS employing the larger error set for the construction of the constraints. This approach guarantees closed-loop stability, constraint satisfaction, and recursive feasibility. In addition, the computation times are almost the same as for JMLS, i.e., very low compared to existing MPC approaches for nonlinear JMS, e.g., [18, 104]. On the downside, the conservatism of the RCIS design is relatively high due to the polytopic over-approximation of the system dynamics. Nevertheless, this approach can be used for the control of single production units, as demonstrated in Chapter 12 with the model of a roller hearth furnace. Suitable state estimation concepts for

the considered setup have been discussed in Chapter 9.

For the control of the overall manufacturing system, a decentralized MPC, a non-cooperative DMPC, and a cooperative DMPC architecture have been proposed in Chapter 11. For the decentralized MPC and the DMPC architectures, semidefinite programs have been developed that can be used to determine RCIS for all local MPCs simultaneously considering the coupling of the subsystems. Using these RCIS, the local MPC formulations are recursively feasible and guarantee PMSS for the overall system. The local MPCs can be formulated as simple QCQPs. Hence, the computation times of the local MPCs are still relatively low. The computation time of the cooperative DMPC depends on the number of downstream subsystems and is larger than that of the decentralized MPC or the non-cooperative DMPC. The control performance of the DMPC approaches is significantly better than for the decentralized setup. For the illustrative example, the cooperation does not improve the control performance, since the global goal is the same as the two local goals. However, the simulations of the hot stamping line show that cooperation may improve the performance significantly.

The RCIS design procedure is only applicable to systems with moderate dimensions and few subsystems, due to the complexity of the semidefinite program. Most likely, it is not applicable to many real manufacturing systems. Hence, it has been investigated whether this special design of the RCIS is necessary. All simulations performed suggest that, for the given application, it is sufficient to employ RCIS, which are determined independently, i.e., ignoring the couplings, in the DMPC architectures proposed. This is not the case for the decentralized MPC setup. Thus, the DMPC approaches presented in this thesis (employing RCIS that are determined independently for the subsystems) propose a promising approach for the control of large scale manufacturing systems. The simulations of the hot stamping line confirm this conclusion.

All in all, the advantages of the approaches proposed are the guarantees concerning stability and recursive feasibility, the low computation time of the local MPCs, and the direct consideration of the fault probabilities in the prediction. The bottleneck of the approaches is the design of the RCIS.

Application to the Hot Stamping Process

The modeling and control approaches presented have been applied to the models of a complete hot stamping line. The modeling of the subsystems and the overall process have been presented in Chapter 4. The continuous dynamics are used to model the temperatures of the processed blanks, while the Markov chain models abrupt production line halts. While there are many finite element method models used for process design, the model presented is the first one suitable for an online control of the blank temperatures. In particular, the probabilistic consideration of production line standstills is a novelty in the hot stamping domain.

In Chapter 12, the MPC approaches for isolated JMS have been tested with the model of a roller hearth furnace. The simulations show that the computation of

RCIS for real systems is challenging but possible if reasonable approximations and relaxations are employed. The computation times of the MPC are low enough for an implementation in a real production line. The results show a good performance of the robust MPC for both reference tracking applications and economic setups. In the reference tracking application, the costs are almost 20 % lower than for the reference controller. In the economic setup, the furnace temperatures are reduced as much as possible to reduce energy consumption. Both cost functions (CoE and EoC) perform nearly the same, since the variance in the system dynamics is moderate. Furthermore, the simulations show that the quadratic constraints guarantee recursive feasibility and robust constraint satisfaction. All in all, the simulations demonstrate that the MPC approaches can be applied to the model of a real production unit and may improve its performance significantly.

Finally, both DMPC approaches have been tested with the model of the whole production line (cf. Chapter 13). It is not possible to calculate RCIS for the whole production line, due to the conservatism involved. Hence, RCIS are only employed for the local MPCs of the furnaces. The simulations show that this is sufficient to guarantee recursive feasibility for the considered setup. If no plant model mismatch is present, the different DMPC approaches perform very similar and stabilize the subsystem close to the reference. In this case, cooperation and disturbance models are not needed. The result is completely different, if persistent disturbances / plant model mismatches occur in the subsystems. In this case, the costs are reduced significantly by the use of the disturbance models (-38 %) and the cooperation of the local MPCs (-16 %). The average costs of the cooperative DMPC are about 68 % lower than for the operation with fixed process variables. More importantly, the number of constraint violations is reduced by 95 %[1]. Since the control performance is nearly the same for both cost functions, the results suggest to use a cooperative DMPC with the CoE cost function for the control of the hot stamping line. These results illustrate the potential of the developed modeling and control approaches for applications to large scale production systems.

14.2. Outlook

The modeling and control approaches presented can be extended in different directions. In addition, some open questions remain, that are worth being explored:

- In the models of the subsystems, the transition probabilities of the Markov chains may depend on the time. So, for example, the change of break down rates with time can be considered. A more precise description of wear processes can be achieved by transition probabilities that may also depend on the system state. In some cases, the inputs of the subsystems influence the transition probabilities. Hence, a more general model of the subsystems regarding the transition

---

[1]The few remaining violations are due to the fact that the persistent disturbances are not considered during the constraint tightening.

probabilities should be considered. This extension poses new challenges for the prediction of the costs, as it introduces additional nonlinearities in the dynamics.

- In Section 6.3, the influence of the prediction horizon on closed-loop stability and the control performance has been illustrated. In contrast to deterministic systems, stability is not guaranteed by large prediction horizons. Thus, a detailed investigation of the relation between stability and the horizon length is of great interest. Stability conditions based on the horizon length and cost parameterization could answer the question whether the use of RCIS and stability constraints is necessary for the system under consideration.

- The relatively conservative design of RCIS is the main drawback of the proposed approaches. The conservatism is caused by the fact that ellipsoidal sets are used, on the one hand, and due to the use of linear differential inclusions to approximate the nonlinear system, on the other hand. To mitigate this problem, approaches for the design of polytopic RCIS for nonlinear systems can be considered. This renders the RCIS design more complex, but most likely results in larger RCIS and provides linear constraints.

- In this thesis, decentralized and distributed MPC architectures have been investigated. In particular, the cooperative DMPC approaches exhibit a good control behavior in the presence of persistent disturbances. However, due to the local definition of the reference values for states and inputs, the disturbances are not compensated completely. In order to improve the disturbance compensation, the reference values of the whole production line have to be adapted. In addition, a modification of the cycle time might be necessary to react to large disturbances. Since the cycle time has to be the same for all subsystems, it cannot be changed only by one local MPC. For both, the reference adoption and the cycle time control, a centralized / global entity is needed. Thus, a hierarchical control architecture is required (cf. [28, 113]). The design of such hierarchical MPC setups is still an open issue for JMS.

  In addition, the integration of the proposed control framework in the enterprise control hierarchy (cf. Fig. 1.1) is an interesting topic. In this context, also discrete inputs can be considered by the global controller, such as scheduling or the shut down of single production units to reduce the production rate. For example, one of the RHFs of the hot stamping line could be deactivated, if the number of parts to be produced decreases. An economic optimum between the number of produced parts and energy consumption could be established by a global controller.

- The simulations with the models of the hot stamping line have revealed a high potential in terms of costs and robustness. Hence, the concept should be implemented and tested on a real hot stamping line. It has to be evaluated whether the resulting process optimizations justify the effort for implementing the complex control setup. In addition, the application of the proposed modeling and control framework to other manufacturing systems should be considered.

Appendix A. Proofs

A.1. Proof of Lemma 6.3

This proof is a generalization of the proof in [133]. For the sake of a brief notation, the Markov state at $k+j$ is denoted by θ_j instead of θ_{k+j}. A trajectory of Markov states $(\theta_0 \dots, \theta_j)$ is denoted by $\boldsymbol{\theta}_{k,j}$. Its realization probability is given by:

$$\Pr(\boldsymbol{\theta}_{k,j}) = \mu_{\theta_0}[k] \cdot \prod_{l=0}^{j-1} p_{\theta_l, \theta_{l+1}}[k+l]. \tag{A.1}$$

Let the set of all Markov states trajectories $\boldsymbol{\theta}_{k,j}$ be denoted by $\boldsymbol{\Theta}_{k,j}$. Then, the expectancy of the state costs can be determined by calculating the costs and probabilities for all trajectories $\boldsymbol{\theta}_{k,j} \in \boldsymbol{\Theta}_{k,j}$:

$$\mathrm{E}\left(\|x[j]\|^2_{Q_{\theta_j}} \right) = \sum_{\boldsymbol{\Theta}_{k,j}} \Pr(\boldsymbol{\theta}_{k,j}) \Bigg[x^\top[k] \left(\prod_{c=0}^{j-1} A^\top_{\theta_c}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j} A^\top_{\theta_{j-c}}[j-c] \right) x[k] \tag{A.2a}$$

$$+ 2 \sum_{l=0}^{j-1} x^\top[k] \left(\prod_{c=0}^{j-1} A^\top_{\theta_c}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_l}[l]\, u[l] \tag{A.2b}$$

$$+ 2 \sum_{l=0}^{j-1} x^\top[k] \left(\prod_{c=0}^{j-1} A^\top_{\theta_c}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l-1} A_{\theta_{j-c}}[j-c] \right) G_{\theta_l}[l]\, \bar{w}[l] \tag{A.2c}$$

$$+ 2 \sum_{l_1=0}^{j-1} \sum_{l_2=0}^{j-1} \bar{w}^\top[l_1]\, G^\top_{\theta_{l_1}}[l_1] \left(\prod_{c=l_1+1}^{j-1} A^\top_{\theta_c}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l_2-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_{l_2}}[l_2]\, u[l_2] \tag{A.2d}$$

$$+ \sum_{l_1=0}^{j-1} \sum_{l_2=0}^{j-1} u^\top[l_1]\, B^\top_{\theta_{l_1}}[l_1] \left(\prod_{c=l_1+1}^{j-1} A^\top_{\theta_c}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l_2-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_{l_2}}[l_2]\, u[l_2] \tag{A.2e}$$

$$+ \sum_{l_1=0}^{j-1} \sum_{l_2=0}^{j-1} \bar{w}^\top[l_1]\, G^\top_{\theta_{l_1}}[l_1] \left(\prod_{c=l_1+1}^{j-1} A^\top_{\theta_c}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l_2-1} A_{\theta_{j-c}}[j-c] \right) G_{\theta_{l_2}}[l_2]\, \bar{w}[l_2] \Bigg] + \Psi', \tag{A.2f}$$

where Ψ' describes the costs caused by the noise $\tilde{w}[l]$. Aggregating the costs in (A.2a) related to the initial state and Ψ' results in Ψ in Eq. (6.40). To shorten the proof, the principle of transforming the terms (A.2b) - (A.2f) is demonstrated only for the terms (A.2b) and (A.2d) here. Due to the definition of the realization probabilities in (A.1) and the definition of the linear operator $\mathcal{T}_i(\cdot)$ according to

(2.54), it follows that:

$$
\begin{aligned}
J' &= \sum_{\boldsymbol{\Theta}_{k,j}} \Pr(\boldsymbol{\theta}_{k,j}) \Bigg[2 \sum_{l=0}^{j-1} x^{\top}[k] \left(\prod_{c=0}^{j-1} A_{\theta_c}^{\top}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_l}[l]\, u[l] \quad\quad\quad\text{(A.3)} \\
&\quad + 2 \sum_{l_1=0}^{j-1} \sum_{l_2=0}^{j-1} \bar{w}^{\top}[l_1] G_{\theta_{l_1}}^{\top}[l_1] \left(\prod_{c=l_1+1}^{j-1} A_{\theta_c}^{\top}[c] \right) Q_{\theta_j} \left(\prod_{c=1}^{j-l_2-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_{l_2}}[l_2]\, u[l_2] \Bigg] \\
&= \sum_{\boldsymbol{\Theta}_{k,j-1}} \Pr(\boldsymbol{\theta}_{k,j-1}) \Bigg[2 \sum_{l=0}^{j-1} x^{\top}[k] \left(\prod_{c=0}^{j-1} A_{\theta_c}^{\top}[c] \right) \mathcal{T}_{\theta_{j-1}}(Q, j-1) \left(\prod_{c=1}^{j-l-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_l}[l]\, u[l] \\
&\quad + 2 \sum_{l_1=0}^{j-1} \sum_{l_2=0}^{j-1} \bar{w}^{\top}[l_1] G_{\theta_{l_1}}^{\top}[l_1] \left(\prod_{c=l_1+1}^{j-1} A_{\theta_c}^{\top}[c] \right) \mathcal{T}_{\theta_{j-1}}(Q, j-1) \left(\prod_{c=1}^{j-l_2-1} A_{\theta_{j-c}}[j-c] \right) B_{\theta_{l_2}}[l_2]\, u[l_2] \Bigg].
\end{aligned}
$$

These terms can be formulated as functions of the trajectories $\boldsymbol{u}[k]$ and $\bar{\boldsymbol{w}}[k]$. To this end, the following abbreviation is introduced:

$$
\boldsymbol{A}_{m,n} := \prod_{l=0}^{m-n} A_{\theta_{m-l}}[m-l] = A_{\theta_m}[m] \cdot A_{\theta_{m-1}}[m-1] \cdot \ldots \cdot A_{\theta_n}[n], \quad m, n \in \mathbb{N}, \ m > n. \quad \text{(A.4)}
$$

Using this definition, it follows that:

$$
\begin{aligned}
J' &= \sum_{\boldsymbol{\Theta}_{k,j-1}} \Pr(\boldsymbol{\theta}_{k,j-1}) \Bigg(2\, x^{\top}[k]\, \boldsymbol{A}_{j-1,0}^{\top} \left[\mathbf{1}_{1\times j} \otimes \mathcal{T}_{\theta_{j-1}}(Q, j-1) \right] \begin{bmatrix} \boldsymbol{A}_{j-1,1} B_{\theta_0}[0]\, u[0] \\ \boldsymbol{A}_{j-1,2} B_{\theta_1}[1]\, u[1] \\ \vdots \\ A_{\theta_{j-1}}[j-1] B_{\theta_{j-2}}[j-2]\, u[j-2] \\ B_{\theta_{j-1}}[j-1]\, u[j-1] \end{bmatrix} \\
&\quad + 2 \begin{bmatrix} \boldsymbol{A}_{j-1,1} G_{\theta_0}[0]\, \bar{w}[0] \\ \boldsymbol{A}_{j-1,2} G_{\theta_1}[1]\, \bar{w}[1] \\ \vdots \\ A_{\theta_{j-1}}[j-1] G_{\theta_{j-2}}[j-2]\, \bar{w}[j-2] \\ G_{\theta_{j-1}}[j-1]\, \bar{w}[j-1] \end{bmatrix}^{\top} \left[\mathbf{1}_{j\times j} \otimes \mathcal{T}_{\theta_{j-1}}(Q, j-1) \right] \begin{bmatrix} \boldsymbol{A}_{j-1,1} B_{\theta_0}[0]\, u[0] \\ \boldsymbol{A}_{j-1,2} B_{\theta_1}[1]\, u[1] \\ \vdots \\ A_{\theta_{j-1}}[j-1] B_{\theta_{j-2}}[j-2]\, u[j-2] \\ B_{\theta_{j-1}}[j-1]\, u[j-1] \end{bmatrix} \Bigg) \\
&= \sum_{\boldsymbol{\Theta}_{k,j-1}} \Pr(\boldsymbol{\theta}_{k,j-1}) \Bigg(2\, x^{\top}[k]\, \boldsymbol{A}_{j-1,0}^{\top} \left[\mathbf{1}_{1\times j} \otimes \mathcal{T}_{\theta_{j-1}}(Q, j-1) \right] \begin{bmatrix} I_{j-1} \otimes A_{\theta_{j-1}}[j-1] & \mathbf{0} \\ \mathbf{0} & B_{\theta_{j-1}}[j-1] \end{bmatrix} \cdots \\
&\quad \cdots \begin{bmatrix} I_{j-2} \otimes A_{\theta_{j-2}}[j-2] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_{\theta_{j-2}}[j-2] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{n_u} \end{bmatrix} \cdots \begin{bmatrix} A_{\theta_1}[1] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & B_{\theta_1}[1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{(j-2)\cdot n_u} \end{bmatrix} \begin{bmatrix} B_{\theta_0}[0] & \mathbf{0} \\ \mathbf{0} & I_{(j-1) n_u} \end{bmatrix} \begin{bmatrix} u[0] \\ \vdots \\ u[j-1] \end{bmatrix} \\
&\quad + 2 \begin{bmatrix} \bar{w}[0] \\ \cdots \\ \bar{w}[j-1] \end{bmatrix}^{\top} \begin{bmatrix} G_{\theta_0}^{\top}[0] & \mathbf{0} \\ \mathbf{0} & I_{(j-1) n_w} \end{bmatrix} \begin{bmatrix} A_{\theta_1}^{\top}[1] & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & G_{\theta_1}^{\top}[1] & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{(j-2) n_w} \end{bmatrix} \cdots \begin{bmatrix} I_{j-1} \otimes A_{\theta_{j-1}}^{\top}[j-1] & \mathbf{0} \\ \mathbf{0} & G_{\theta_{j-1}}^{\top}[j-1] \end{bmatrix} \cdots \\
&\quad \cdots \left[\mathbf{1}_{j\times j} \otimes \mathcal{T}_{\theta_{j-1}}(Q, j-1) \right] \begin{bmatrix} I_{j-1} \otimes A_{\theta_{j-1}}[j-1] & \mathbf{0} \\ \mathbf{0} & B_{\theta_{j-1}}[j-1] \end{bmatrix} \cdots \begin{bmatrix} B_{\theta_0}[0] & \mathbf{0} \\ \mathbf{0} & I_{(j-1) n_u} \end{bmatrix} \begin{bmatrix} u[0] \\ \vdots \\ u[j-1] \end{bmatrix} \Bigg).
\end{aligned}
$$

The presented reformulation can be applied analogously to the terms (A.2c), (A.2e), and (A.2f). Using the prediction matrices introduced in (6.41), the cost

terms can be formulated as follows:

$$
\begin{aligned}
\mathrm{E}\left(\|x[j]\|_{Q_{\theta_j}}^2\right) = \sum_{\Theta_{k,j-1}} \Pr(\boldsymbol{\theta}_{k,j-1}) \Big(& 2x^\top[k]\, \boldsymbol{A}_{j-1,0}^\top\, \boldsymbol{Q}_{\mathrm{xu},\theta_{j-1}}(j)\, \boldsymbol{B}_{\theta_{j-1}}'[j] \cdot \ldots \cdot \boldsymbol{B}_{\theta_0}'[1]\, \boldsymbol{u}[k] \\
& + 2x^\top[k]\, \boldsymbol{A}_{j-1,0}^\top\, \boldsymbol{Q}_{\mathrm{xw},\theta_{j-1}}(j)\, \boldsymbol{G}_{\theta_{j-1}}'[j] \cdot \ldots \cdot \boldsymbol{G}_{\theta_0}'[1]\, \bar{\boldsymbol{w}}[k] \\
& + 2\bar{\boldsymbol{w}}^\top[k]\, \big(\boldsymbol{G}_{\theta_0}'[1]\big)^\top \cdot \ldots \cdot \big(\boldsymbol{G}_{\theta_{j-1}}'[j]\big)^\top \boldsymbol{Q}_{\mathrm{wu},\theta_{j-1}}(j)\, \boldsymbol{B}_{\theta_{j-1}}'[j] \cdot \ldots \cdot \boldsymbol{B}_{\theta_0}'[1]\, \boldsymbol{u}[k] \\
& + \boldsymbol{u}^\top[k]\, \big(\boldsymbol{B}_{\theta_0}'[1]\big)^\top \cdot \ldots \cdot \big(\boldsymbol{B}_{\theta_{j-1}}'[j]\big)^\top \boldsymbol{Q}_{\mathrm{uu},\theta_{j-1}}(j)\, \boldsymbol{B}_{\theta_{j-1}}'[j] \cdot \ldots \cdot \boldsymbol{B}_{\theta_0}'[1]\, \boldsymbol{u}[k] \\
& + \bar{\boldsymbol{w}}^\top[k]\, \big(\boldsymbol{G}_{\theta_0}'[1]\big)^\top \cdot \ldots \cdot \big(\boldsymbol{G}_{\theta_{j-1}}'[j]\big)^\top \boldsymbol{Q}_{\mathrm{ww},\theta_{j-1}}(j)\, \boldsymbol{G}_{\theta_{j-1}}'[j] \cdot \ldots \cdot \boldsymbol{G}_{\theta_0}'[1]\, \bar{\boldsymbol{w}}[k] \Big) + \Psi.
\end{aligned}
\tag{A.5}
$$

Equation (A.5) can be transformed into the form presented in (6.40). In this formulation, the prediction matrices are still determined by a sum over all possible Markov trajectories. By reordering the sums, the recursions in Lemma 6.3 can be derived. Since, the procedure is the same for all cost terms, it is only presented for the costs described by $\Phi_{\mathrm{wu}}[j]$:

$$
\begin{aligned}
\Phi_{\mathrm{wu}}[j] &= \sum_{\theta_0=1}^{n_\theta} \cdots \sum_{\theta_{j-1}=1}^{n_\theta} \Pr(\boldsymbol{\theta}_{k,j-1}) \big(\boldsymbol{G}_{\theta_0}'[1]\big)^\top \cdot \ldots \cdot \big(\boldsymbol{G}_{\theta_{j-1}}'[j]\big)^\top \boldsymbol{Q}_{\mathrm{wu},\theta_{j-1}}(j)\, \boldsymbol{B}_{\theta_{j-1}}'[j] \cdot \ldots \cdot \boldsymbol{B}_{\theta_0}'[1] \\
&= \sum_{\theta_0=1}^{n_\theta} \cdots \sum_{\theta_{j-2}=1}^{n_\theta} \Pr(\boldsymbol{\theta}_{k,j-2}) \big(\boldsymbol{G}_{\theta_0}'[1]\big)^\top \cdot \ldots \cdot \big(\boldsymbol{G}_{\theta_{j-2}}'[j-1]\big)^\top \\
&\qquad \cdots \left(\sum_{\theta_{j-1}=1}^{n_\theta} p_{\theta_{j-2},\theta_{j-1}} \underbrace{\big(\boldsymbol{G}_{\theta_{j-1}}'[j]\big)^\top \boldsymbol{Q}_{\mathrm{wu},\theta_{j-1}}(j)\, \boldsymbol{B}_{\theta_{j-1}}'[j]}_{=:\, \phi_{\mathrm{wu},\theta_{j-1}}^{(1)}}\right) \boldsymbol{B}_{\theta_{j-2}}'[j-1] \cdot \ldots \cdot \boldsymbol{B}_{\theta_0}'[1] \\
&= \sum_{\theta_0=1}^{n_\theta} \cdots \sum_{\theta_{j-2}=1}^{n_\theta} \Pr(\boldsymbol{\theta}_{k,j-2}) \big(\boldsymbol{G}_{\theta_0}'[1]\big)^\top \cdot \ldots \cdot \underbrace{\big(\boldsymbol{G}_{\theta_{j-2}}'[j-1]\big)^\top \mathcal{T}_{\theta_{j-2}}\left(\phi_{\mathrm{wu}}^{(1)}, j-2\right) \boldsymbol{B}_{\theta_{j-2}}'[j-1] \cdot \ldots \cdot \boldsymbol{B}_{\theta_0}'[1]}_{=:\, \phi_{\mathrm{wu},\theta_{j-2}}^{(2)}} \\
&\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&= \sum_{\theta_0=1}^{n_\theta} \mu_{\theta_0}[k] \underbrace{\big(\boldsymbol{G}_{\theta_0}'[1]\big)^\top \mathcal{T}_{\theta_0}\left(\phi_{\mathrm{wu}}^{(j-1)}, 0\right) \boldsymbol{B}_{\theta_0}'[1]}_{=:\, \phi_{\mathrm{wu},\theta_0}^{(j)}} = \sum_{\theta_0=1}^{n_\theta} \mu_{\theta_0}[k]\, \phi_{\mathrm{wu},\theta_0}^{(j)}.
\end{aligned}
\tag{A.6}
$$

These equations coincide with the initialization, recursion, and matrix calculation as defined in Lemma 6.3. The procedure for the remaining cost terms defined by $\Phi_{\mathrm{uu}}[j], \Phi_{\mathrm{xu}}[j], \Phi_{\mathrm{xw}}[j]$, and $\Phi_{\mathrm{ww}}[j]$ can be shown analogously. $\qquad\square$

A.2. Proof of Theorem 11.4

The first summands in (11.38) and (11.39) describe the local costs of subsystem $n \in \mathbb{N}_{n_\mathrm{s}}$ and coincide with the definitions in Theorem 6.2 for a single JMLS. The remaining summands describe the costs J^m of the succeeding subsystems. The state

and input costs of these subsystems, employing the CoE definition (cf. (5.8) on page 78), are given by:

$$\sum_{m \in \text{Succ}(n,k_{\text{b}})} \gamma^m \left(\|\bar{\boldsymbol{x}}^m[\![n]\!] - \boldsymbol{x}_{\text{r}}^m[\![n]\!]\|_{\boldsymbol{Q}^m}^2 + \|\boldsymbol{u}^m[\![n]\!] - \bar{\boldsymbol{u}}_{\text{r}}^m[\![n]\!]\|_{\boldsymbol{R}^m}^2 \right). \tag{A.7}$$

The predicted input costs of the succeeding subsystems (second summand in (A.7)) cannot be influenced directly by the local inputs $\boldsymbol{u}^n[k^n]$. Hence, these terms belong to $\Psi[k^n]$ and are ignored. It remains to determine the state costs. To this end, $\bar{\boldsymbol{x}}^m[\![n]\!]$ is formulated as a function of $\boldsymbol{u}^n[k^n]$.

By the definition of the prediction equations in (6.20) and Equations (11.40) and (11.41), it follows that:

$$\bar{\boldsymbol{x}}^n[\![1]\!] = \boldsymbol{A}^n[k^n]\, x^n[k^n] + \boldsymbol{B}^n[k^n]\, \boldsymbol{u}^n[k^n] + \boldsymbol{G}^n[k^n]\bar{\boldsymbol{w}}^n[k^n] \tag{A.8}$$

$$= \boldsymbol{f}^{n,n}[k^n] + \boldsymbol{B}^{n,n}[k^n]\, \boldsymbol{u}^n[k^n]. \tag{A.9}$$

The predicted state trajectory of the direct successor $m_2 = \text{succ}(n, k_{\text{b}})$ is:

$$\bar{\boldsymbol{x}}^{m_2}[\![n]\!] = \boldsymbol{A}^{m_2}[k^n]\, x^{m_2}[\![n]\!] + \boldsymbol{B}^{m_2}[k^n]\, \boldsymbol{u}^{m_2}[\![n]\!] + \boldsymbol{G}^{m_2}[k^n] \begin{bmatrix} \bar{\boldsymbol{w}}_{\text{p}}^{m_2}[\![n]\!] \\ \bar{\boldsymbol{w}}_{\text{d}}^{m_2}[\![n]\!] \end{bmatrix}. \tag{A.10}$$

From the coupling definition (10.2), it follows that:

$$\bar{\boldsymbol{w}}_{\text{p}}^{m_2}[\![n]\!] = \boldsymbol{T}_{\text{y}}^n\bar{\boldsymbol{x}}^n[\![1]\!] = \boldsymbol{T}_{\text{y}}^n \left(\boldsymbol{B}^{n,n}[k]\, \boldsymbol{u}^n[k] + \boldsymbol{f}^{n,n}[k] \right). \tag{A.11}$$

Inserting (A.11) into (A.10) and using the recursion equations (11.42) and (11.43), one obtains:

$$\bar{\boldsymbol{x}}^{m_2}[\![n]\!] = \boldsymbol{f}^{n,m_2}[k^n] + \boldsymbol{G}_{\text{p}}^{m_2}[k^n]\, \boldsymbol{T}_{\text{y}}^n \boldsymbol{B}^{n,n}[k^n]\boldsymbol{u}^n[k^n] \tag{A.12}$$

$$= \boldsymbol{f}^{n,m_2}[k^n] + \boldsymbol{B}^{n,m_2}[k^n]\boldsymbol{u}^n[k^n]. \tag{A.13}$$

By induction, it can be shown that the predicted trajectories of all downstream subsystems $m \in \text{Succ}(n, k_{\text{b}})$ are given by:

$$\bar{\boldsymbol{x}}^m[\![n]\!] = \boldsymbol{f}^{n,m}[k^n] + \boldsymbol{B}^{n,m}[k^n]\boldsymbol{u}^n[k^n] \tag{A.14}$$

if the vectors $\boldsymbol{f}^{n,m}[k^n]$ and matrices $\boldsymbol{B}^{n,m}[k^n]$ are determined according to (11.42) and (11.43). Inserting the state predictions $\bar{\boldsymbol{x}}^m[\![n]\!]$ according to (A.14) into the state costs terms in (A.7), one obtains the following state cost terms:

$$\gamma^m \left\| \boldsymbol{B}^{n,m}[k^n]\boldsymbol{u}^n[k^n] + \boldsymbol{f}^{n,m}[k^n] - \boldsymbol{x}_{\text{r}}^m[\![n]\!] \right\|_{\boldsymbol{Q}^m}^2. \tag{A.15}$$

Expanding the summands results in (11.45) and (11.46). □

A.3. Proof of Theorem 11.5

The local costs of the n-th subsystem are given by the first summand in (11.48) and the summand in (11.49). These terms are the same as for the isolated case in Theorem 6.3 and the corresponding proof applies. The remaining sums in (11.48) and (11.50) describe the expected values of the state costs $\mathrm{E}\big(\|\boldsymbol{x}^m[n] - \boldsymbol{x}_r^m[n]\|_{\boldsymbol{Q}^m(\boldsymbol{\theta}_{k,N}^m)}^2\big)$ of the downstream subsystems. According to Lemmata 6.2 and 6.3 and considering that only costs caused by $\bar{\boldsymbol{w}}_p^m[n]$ can be influenced by $\boldsymbol{u}^n[k^n]$, it holds that:

$$\mathrm{E}\big(\|\boldsymbol{x}^m[n] - \boldsymbol{x}_r^m[n]\|_{\boldsymbol{Q}^m(\boldsymbol{\theta}_{k,N}^m)}^2\big) = \sum_{j=1}^{N^m}\Big[\big[(\bar{\boldsymbol{w}}_p^m[n])^\top \quad 2(\bar{\boldsymbol{w}}_d^m[n])^\top\big]\Phi_{\mathrm{ww}_p}^m[j]\,\bar{\boldsymbol{w}}_p^m[n] + \dots \quad \text{(A.16)}$$

$$\dots + 2\left(x^m[n]\right)^\top \Phi_{\mathrm{xw}_p}^m[j]\bar{\boldsymbol{w}}_p^m[n] + 2\left(\boldsymbol{u}^m[n]\right)^\top \Phi_{\mathrm{uw}_p}^m[j]\bar{\boldsymbol{w}}_p^m[n] - 2\Phi_{\mathrm{x_r w_p}}[j]\bar{\boldsymbol{w}}_p^m[n]\Big] + \Psi'$$

with $\Phi_{\mathrm{uw}_p}^m[j] = \big(\Phi_{\mathrm{w_p u}}^m[j]\big)^\top$. Inserting the prediction equation (A.14) for $\bar{\boldsymbol{x}}^m[n]$ into the coupling condition (10.2), one obtains that:

$$\bar{\boldsymbol{w}}_p^m[n] = \boldsymbol{T}_y^{m_1}\left(\boldsymbol{f}^{n,m_1}[k^n] + \boldsymbol{B}^{n,m_1}[k^n]\boldsymbol{u}^n[k^n]\right), \tag{A.17}$$

where $m_1 = \mathrm{pre}(m, k_\mathrm{b})$. Inserting this term into (A.16) results in the equations (11.51) and (11.52) and completes the proof. $\qquad\square$

Appendix B. Markov State Dependent Inputs

The approaches presented in Chapter 6 determine inputs $u[j]$ that are independent of the Markov state θ_{k+j}. The fact, that the Markov state is measured, is not exploited in the prediction. This causes the problems mentioned in Section 6.3 concerning control performance and stationary accuracy. This section presents how inputs, that depend on the Markov state, can be considered. Therefore, for all prediction steps $0 \le j < N$ and all Markov states $i \in \Theta$, an input $u_i[j]$ is determined. Thus, an augmented optimization variable is defined:

$$\boldsymbol{u}'[k] := \begin{bmatrix} u_1[0] & \cdots & u_{n_\theta}[0] & \cdots & u_1[N-1] & \cdots & u_{n_\theta}[N-1] \end{bmatrix}. \tag{B.1}$$

Due to this definition, the future inputs $u[j] = u_{\theta_{k+j}}[j]$ are not deterministic anymore. Hence, the MPC problems have to be adapted. The CoE MPC is defined by:

$$\min_{\boldsymbol{u}'[k]} \quad \sum_{j=1}^{N}\|\bar{x}[j] - x_\mathrm{r}[j]\|_Q^2 + \sum_{j=0}^{N-1}\mathrm{E}\left(\|u_{\theta_{k+j}}[j] - u_{\mathrm{r},\theta_{k+j}}[j]\|_R^2\right) \tag{B.2a}$$

$$\text{s.t.} \quad H_{\text{x}}'\bar{x}[j] + H_{\text{u}}'u_i[j] \le h'[k+j] \qquad \forall j \in \mathbb{N}_{N-1}^0, i \in \Theta. \tag{B.2b}$$

The EoC MPC is defined by:

$$\min_{\boldsymbol{u}'[k]} \quad \mathrm{E}\left(\sum_{j=1}^{N} \|x[j] - x_{\text{r}}[j]\|_{Q_{\theta_{k+j}}}^2 + \sum_{j=0}^{N-1} \|u_{\theta_{k+j}}[j] - u_{\text{r},\theta_{k+j}}[j]\|_{R_{\theta_{k+j}}}^2\right) \tag{B.3a}$$

$$\text{s.t.} \quad H_{\text{x}}'\bar{x}[j] + H_{\text{u}}'u_i[j] \le h'[k+j] \qquad \forall j \in \mathbb{N}_{N-1}^0, i \in \Theta. \tag{B.3b}$$

To solve these problems, an extended JMLS $\mathcal{S}_1'$ is introduced:

Definition B.1. *Let $\mathcal{S}_1'$ be a JMLS where $B_i'[k] = \begin{bmatrix} \boldsymbol{0}_{n_{\text{x}}\times(i-1)n_{\text{u}}} & B_i[k] & \boldsymbol{0}_{n_{\text{x}}\times(n_\theta-i)n_{\text{u}}} \end{bmatrix}$ and $u'[k] := \begin{bmatrix} u_1[k] & \cdots & u_{n_\theta}[k] \end{bmatrix}$. All other components are as defined in (6.1).*

By applying the CoE and EoC approaches presented in Chapter 6 to the extended system, the desired MPC formulations result:

Theorem B.1. *If the CoE approach according to Theorem 6.2 is applied to the extended JMS $\mathcal{S}_1'$ with the following adaptions:*

$$\boldsymbol{R}'[k] := \mathrm{diag}\left(\mu_1[0]R, \cdots, \mu_{n_\theta}[0]R, \cdots, \mu_1[N-1]R, \cdots, \mu_{n_\theta}[N-1]R\right) \tag{B.4a}$$

$$\boldsymbol{u}_{\text{r}}[k] := \begin{bmatrix} u_{\text{r},1}[0] & \cdots & u_{\text{r},n_\theta}[0] & \cdots & u_{\text{r},1}[N-1] & \cdots & u_{\text{r},n_\theta}[N-1] \end{bmatrix} \tag{B.4b}$$

and constraints defined by $H_{\text{x}}', H_{\text{u}}'$, and $h'[k]$, the solution $\boldsymbol{u}[k]$ also solves the optimization problem (B.2).

Proof. The dynamics of $\mathcal{S}_1'$ and $\mathcal{S}_1$ are identical but for the inputs. The definition of $B_i'[k]$ and $u'[k]$ imply that $B_i'[k]u'[j] = B_i[k]u_i[j]$. Hence, the prediction equations for the expected values of the states, the corresponding costs, and the constraints do not have to be changed. Since the inputs and the input reference depend on the same Markov state, it holds that:

$$\mathrm{E}\left(\|u_{\theta_{k+j}}[j] - u_{\text{r},\theta_{k+j}}[j]\|_R^2\right) = \sum_{i=1}^{n_\theta} \mu_i[j] \|u_i[j] - u_{\text{r},i}[j]\|_R^2 \tag{B.5}$$

$$= \left(u'[j] - u_{\text{r}}'[j]\right)^\top \mathrm{diag}\left(\mu_1[j]R, \cdots, \mu_{n_\theta}[j]R\right)\left(u'[j] - u_{\text{r}}'[j]\right),$$

where $u_{\text{r}}'[j] := \begin{bmatrix} u_{\text{r},1}[j] & \cdots & u_{\text{r},n_\theta}[j] \end{bmatrix}$. Hence, the equivalence of the expected input costs and the overall optimization problem follows. $\square$

Theorem B.2. *If the EoC approach according to Theorem 6.3 is applied to the extended JMS $\mathcal{S}_1'$ with:*

$$\boldsymbol{R}' := \mathrm{diag}\left(\mu_1[0]R_1, \cdots, \mu_{n_\theta}[0]R_{n_\theta}, \cdots, \mu_1[N-1]R_1, \cdots, \mu_{n_\theta}[N-1]R_{n_\theta}\right) \tag{B.6}$$

and $\boldsymbol{u}_{\text{r}}[k]$ defined as in (B.4b) *as well as constraints defined by $H_{\text{x}}', H_{\text{u}}'$, and $h'[k]$, the solution $\boldsymbol{u}[k]$ also solves the optimization problem B.3.*

Proof. The reasoning concerning the state costs in the proof of Theorem B.1 applies also to this theorem. Replacing R by R_{θ_k} results into:

$$\mathrm{E}\left(\left\|u_{\theta_{k+j}}[j] - u_{\mathrm{r},\theta_{k+j}}[j]\right\|^2_{R_{\theta_{k+j}}}\right) = \sum_{i=1}^{n_\theta} \mu_i[j]\left\|u_i[j] - u_{\mathrm{r},i}[j]\right\|^2_{R_i} \tag{B.7}$$

$$= \left(u'[j] - u'_{\mathrm{r}}[j]\right)^\top \mathrm{diag}\left(\mu_1[j]R_1, \cdots, \mu_{n_\theta}[j]R_{n_\theta}\right)\left(u'[j] - u'_{\mathrm{r}}[j]\right).$$

Hence, the equivalence of costs and constraints is established. □

These theorems show that only minor adjustments to the proposed MPC approaches are necessary in order to consider input trajectories that depend on the Markov state. However, the dimensions of the prediction equations and the optimization problem increase significantly.

Example B.1. As pointed out in Section 6.3, Markov state dependent inputs can be used to control a system to a general state reference without offset even if the input reference depends on the Markov state. To illustrate this, the simulation presented in Example 6.4 on page 99 was repeated with the approaches proposed in this section. The results can be found in Fig. B.1. The simulation results show that both approaches converge to the reference. In addition, the convergence is faster than for the approaches proposed in the Sections 6.1 and 6.2 - especially for the EoC approach, which now performs better than the CoE approach. This illustrates that the variance of the predicted states is reduced significantly by considering inputs that depend on the Markov state. △
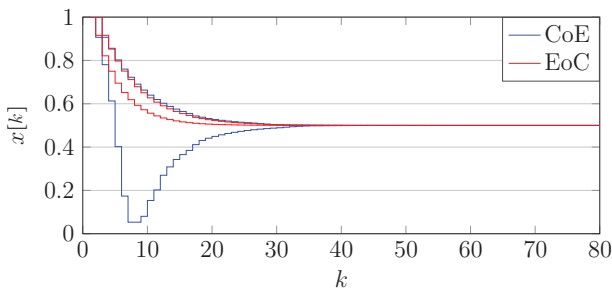

Figure B.1.: State trajectory envelopes of the JMS (6.55) controlled by both MPC approaches presented in this section.

Appendix C. Transition Probabilities of a Hot Stamping Line

The following tables show how the transition probabilities (TP) of the RHFs, the chamber furnace, the robot, and the press can be calculated.

Table C.1.: Transition probabilities for the Markov chain $\mathcal{M}^1$ of the RHFs.

probability	method of calculation
$p_{1,1}^1[k]$	$p_{1,1}[k] = 1 - p_{1,2}^1[k] - p_{1,3}^1[k]$
$p_{1,2}^1[k]$, $p_{3,2}^1[k]$	This transition occurs if the production line stops due to a fault on any subsystem, and a blank pair is at the last position of the furnace. According to (3.12) the overall fault rate λ_f can be calculated. If a blank pair arrives at the last position, the furnace stops with probability λ_f. The TP can be stated by $p_{1,2}^1[k] = p_{b,9}[k] \cdot \lambda_f$. Future values of $p_{1,2}^1[k]$ can be calculated with $p_{b,1:8}$.
$p_{1,3}^1[k], p_{2,3}^1[k]$	The transition to the furnace emptying mode is selected by the operator. The TP can be calculated from recordings of the PLC.
$p_{2,1}^1[k]$	In normal operation, a mean repair rate $p_{2,1}^1 = \lambda_r$ based on the overall MTTR is used. Once a specific fault has occurred, the repair rate of the production unit causing the fault is used $p_{2,1}^1 = \lambda_r^n[k]$.
$p_{2,2}^1[k]$	$p_{2,2}^1[k] = 1 - p_{2,1}^1[k] - p_{2,3}^1[k]$
$p_{3,1}^1[k]$	The transition to the normal mode is selected by the operator. The TP can be calculated from recordings of the PLC.
$p_{3,3}^1[k]$	$p_{3,3}^1[k] = 1 - p_{3,1}^1[k] - p_{3,2}^1[k] - p_{3,4}^1[k]$.
$p_{3,4}^1[k]$	For the sake of simplicity, $p_{3,4}^1[k] = 0$ until the transition occurs. This approximation can be used, since both $\theta^1 = 3$ and $\theta^1 = 4$ use similar dynamics. The real probabilities can be calculated by simulation of all possible trajectories including the transition to the standby mode, once the furnace is empty.
$p_{4,1}^1[k]$	The TP can be calculated from recordings of the PLC.
$p_{4,4}^1[k]$	$p_{4,4}^1[k] = 1 - p_{4,1}^1[k]$.

The TP of the chamber furnace can be calculated based in the probabilities that a blank pair exits the furnace.

Table C.2.: Transition probabilities for the Markov chain $\mathcal{M}^3$ of the chamber furnace.

probability	method of calculation
$p_{1,1}^3[k]$, $p_{2,1}^3[k]$	If a blank exits the RHF, this transition is true. Hence, the transition probabilities equals the probability of a blank exit predicted by the output equation of $\mathcal{S}^1$ or $\mathcal{S}^2$ regardless of the current state. The TPs can be calculated as follows: $p_{1,1}^3[k] = p_{2,1}^3[k] = p_b^n[\kappa_n(k+1)]$, $n = \mathrm{pre}(\mathcal{S}^3, k+1)$.
$p_{1,2}^3[k]$, $p_{2,2}^3[k]$	$p_{1,2}^3[k] = p_{2,2}^3[k] = 1 - p_b^n[\kappa_n(k+1)]$, $n = \mathrm{pre}(\mathcal{S}^3, k+1)$.

By this configuration, the transition probabilities become unconditioned and the probability distribution follows the probability that a blank is transferred to the chamber furnace. The transition probabilities for the Markov chain $\mathcal{M}^4$ can be constructed similar considering that blanks are only supplied by one subsystem.

Table C.3.: Transition probabilities for the Markov chain $\mathcal{M}^4$ of the robot.

probability	method of calculation
$p_{1,1}^4[k]$, $p_{2,1}^4[k]$	If a blank exits the chamber furnace, this transition is true. Hence, the transition probability equals the probability of a blank exit predicted by the output equation of $\mathcal{S}^3$ regardless of the current state, i.e., $p_{1,1}^4[k] = p_{2,1}^4[k] = p_b^3[\kappa_3(k+1)]$.
$p_{1,2}^4[k]$, $p_{2,2}^4[k]$	$p_{1,2}^4[k] = p_{2,2}^4[k] = 1 - p_b^3[\kappa_3(k+1)]$.

Again, the transition probabilities are unconditioned, and the probability distribution bases on the probability that a blank is transferred with the robot. The transition probabilities for the Markov chain $\mathcal{M}^5$ can be constructed as for $\mathcal{M}^4$.

Appendix D. Simulation Results for the Hot Stamping Line

This appendix shows an overview of the average costs for the simulations of the whole hot stamping line subject to persistent disturbances according to Table 13.5.

Table D.1.: Comparison of average costs J in $\cdot 10^5$ of the different setups for all 48 simulations according to Table 13.5.

cost function	EoC			CoE			ref.
architecture	non-coop.		coop.	non-coop.		coop.	
distur. model	no	yes		no	yes		
1	11.64	8.61	6.62	11.68	8.63	6.59	23.74
2	7.52	6.29	5.50	7.55	6.31	5.49	17.11
3	12.88	7.86	7.10	12.87	7.89	7.08	14.66
4	8.02	5.98	5.69	8.01	5.97	5.68	12.31
5	9.07	6.92	5.57	9.09	6.94	5.55	20.61
6	6.44	5.51	5.08	6.44	5.52	5.08	15.89
7	10.55	6.40	5.90	10.56	6.42	5.90	13.00
8	7.15	5.41	5.31	7.16	5.41	5.30	11.91
9	7.51	7.10	6.17	7.55	7.13	6.15	15.19
10	5.92	5.84	5.39	5.94	5.86	5.38	13.41
11	7.31	6.75	5.99	7.29	6.74	6.05	13.79
12	5.81	5.65	5.42	5.80	5.64	5.44	12.69
13	7.56	6.32	5.46	7.58	6.34	5.43	16.91
14	12.76	7.66	6.95	12.73	7.63	6.95	15.18
15	9.16	7.01	5.59	9.17	7.00	5.58	20.60
16	5.78	5.62	5.44	5.76	5.61	5.45	12.91
17	30.02	16.54	10.39	30.09	16.60	10.35	51.90
18	24.58	13.22	8.14	24.66	13.29	8.15	43.97
19	32.44	10.90	9.97	32.35	10.84	9.92	23.60
20	26.54	8.66	7.75	26.47	8.63	7.64	20.45
21	19.17	11.58	6.66	19.17	11.59	6.67	39.78
22	16.03	9.66	5.70	16.04	9.66	5.70	34.18
23	23.76	8.57	6.92	23.75	8.59	6.91	16.02
24	19.71	7.21	5.89	19.71	7.21	5.82	14.55
25	15.86	11.81	8.76	15.96	11.88	8.71	24.61
26	13.49	10.05	7.42	13.57	10.11	7.38	22.13
27	13.79	9.02	7.63	13.69	8.96	7.66	19.57
28	11.82	7.88	6.80	11.74	7.84	6.83	17.80
29	17.33	13.51	12.87	17.38	13.53	12.85	31.81
30	14.76	11.60	10.93	14.81	11.63	10.93	28.41
31	17.47	13.62	12.89	17.53	13.65	12.87	31.69

32	14.72	11.58	10.92	14.75	11.59	10.89	27.91
33	22.87	11.11	9.57	22.87	11.13	9.57	36.63
34	18.98	9.22	8.03	18.98	9.24	8.02	31.70
35	24.43	9.24	8.14	24.45	9.27	8.12	24.92
36	20.27	7.95	7.02	20.28	7.95	7.00	21.88
37	6.03	5.98	6.08	6.03	5.97	6.06	16.91
38	5.78	5.83	5.95	5.78	5.83	5.93	14.91
39	5.11	5.18	5.21	5.11	5.18	5.19	13.05
40	6.59	5.51	5.57	6.60	5.52	5.56	16.70
41	5.96	5.57	5.64	5.98	5.59	5.63	14.56
42	5.49	5.56	5.66	5.50	5.57	5.64	13.20
43	5.20	5.28	5.37	5.19	5.26	5.35	12.78
44	4.98	4.70	4.81	4.99	4.71	4.78	13.63
45	11.56	7.99	5.87	11.59	8.02	5.85	25.66
46	23.45	14.26	6.07	23.47	14.29	6.07	43.98
47	19.70	8.72	7.35	19.70	8.73	7.33	32.37
48	5.87	5.92	5.89	5.89	5.93	5.89	14.93

List of Symbols

Functions

$\kappa_{\boldsymbol{n}} : \mathbb{N}^0 \to \mathbb{N}^0$ mapping of the base cycle counter k_{b} to the corresponding time index $k^{\boldsymbol{n}}$ of the n-th subsystem

$\rho : \mathbb{R}^n \times \mathcal{E} \to \mathbb{R}_{\geq 0}$ support function of the ellipsoid $\mathcal{E}$

$f_{\chi^2} : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ probability density function of the χ^2-distribution

$F_{\chi^2} : \mathbb{R}_{\geq 0} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$ cumulative density function of the χ^2-distribution

$f_i : \Xi \times \mathbb{Z} \to \mathbb{X}$ continuous state dynamics for Markov state i

$g_i : \Xi \times \mathbb{Z} \to \mathbb{Y}$ output equations for Markov state i

$h_i : \Xi \to \mathbb{R}^{n_z}$ measurement output equations for Markov state i

$J : \mathbb{X}^N \times \mathbb{U}^N \times \mathbb{Y}^N \times \Theta_{k,N} \times \mathbb{R} \to \mathbb{R}$ costs as a function of the input trajectory, state trajectory, output trajectory, Markov state trajectory, and the horizon N

$V : \mathbb{X} \times \Theta \to \mathbb{R}_{\geq 0}$ Markov state dependent Lyapunov function

General

$\star$ transposed entries in a symmetric matrix

$*$ non-zero entries in a matrix

$\bar{\cdot}$ expected value, quantity related to an expected value

$\check{\cdot}$ quantities related to state estimation

$\hat{\cdot}$ quantity of an augmented system

$\tilde{\cdot}, \acute{\cdot}$ auxiliary variables

$(\cdot)'$ auxiliary variables

$(\cdot)_\delta$ quantity of a 'delta system' (see page 98)

$(\cdot)_{\mathrm{c}}$	continuous or cycle
$(\cdot)_i$	quantity or function for Markov state i
$(\cdot)_{\mathrm{r}}$	reference, steady state
$(\cdot)^{\boldsymbol{m}}$	quantity of the m-th subsystem
$(\cdot)^{(m)}$	quantity at the m-th step of an iteration or recursion
$(\cdot)[k]$	quantity at discrete-time k
$(\cdot)(t)$	quantity at continuous time t
$(\cdot)\llbracket j \rrbracket$	quantity predicted by j steps at time k
$(\cdot)\llbracket j,l \rrbracket$	matrix that describes the influence of a quantity at time $k + l$ on a predicted quantity at $k + j$
$\mathfrak{E}[k_{\mathrm{b}}]$	set of edges of the graph $\mathfrak{G}[k_{\mathrm{b}}]$
$\mathfrak{G}[k]$	time dependent directed graph describing the product flow
i	index - used to indicate the Markov state
j	index - often for a prediction step
k	(discrete) time index
l, m, n	general indices
$\mathcal{M}$	Markov chain
$n_{(\cdot)}$	number or dimension of indicated quantity
n_{s}	number of subsystems
$\mathcal{N}(l, \Sigma)$	normal distribution with mean l and covariance matrix Σ
$\mathfrak{N}$	set of nodes of the graph $\mathfrak{G}[k_{\mathrm{b}}]$
$\mathcal{S}$	dynamical system
t	(continuous) time
Δt	sampling time

Operators

$\ominus, \oplus$	Minkowski difference and Minkowski sum

$\otimes$	Kronecker product
$\lfloor \cdot \rceil$	operator that returns an over-approximation of a set in form of a hyperbox
$\lfloor \cdot \rfloor$	floor function (rounding to lower integer)
$\|\cdot\|, \|\cdot\|_M$	euclidean vector norm, weighted norm: $\|x\|_M^2 := x^\top M x$
$(\cdot)^\top$	transpose of a vector or matrix
$\mathbb{1}_{\{\cdot\}}$	indicator function
$\nabla(\cdot)$	gradient
$\lambda_{\min}(\cdot), \lambda_{\max}(\cdot)$	smallest eigenvalue of a matrix, largest eigenvalue if a matrix
$\lambda_{\min}(\cdot,\cdot), \lambda_{\max}(\cdot,\cdot)$	smallest and larges generalized eigenvalue of two matrices
$\mathrm{Cov}(\cdot)$	covariance
$\det(\cdot)$	determinant of a matrix
$\mathrm{diag}(\cdot)$	returns a diagonal or block diagonal matrix from the arguments
$\mathrm{E}(\cdot)$	expectancy operator
$\mathcal{O}(\cdot)$	Big O notation for computational complexity
$\Pr(\cdot)$	probability operator
$\mathrm{pre}(n, k_\mathrm{b})$	index set of predecessors of subsystem $\mathcal{S}^n$
$\mathrm{succ}(n, k_\mathrm{b})$	index set of successors of subsystem $\mathcal{S}^n$
$\mathrm{Succ}(n, k_\mathrm{b})$	index set of all downstream subsystems of subsystem $\mathcal{S}^n$
$\mathcal{T}_i(\cdot), \mathcal{T}_i(\cdot,\cdot)$	linear matrix operator for a set of matrices for Markov state i
$\mathrm{tr}(\cdot)$	trace operator
$\mathcal{V}_i(\cdot), \mathcal{V}_i(\cdot,\cdot)$	linear matrix operator for a set of matrices for Markov state i
$\mathrm{Var}(\cdot)$	variance

Physical Quantities

$\varepsilon(\cdot)$	emissivity [in -]
$\lambda_{(\cdot)}$	thermal conductivity [in $\mathrm{W/m^2 \cdot K}$]

List of Symbols

σ	Stefan-Boltzmann constant [5.6704 $\mathrm{W/m^2 \cdot K^4}$]
$\mathrm{Ac_1}$, $\mathrm{Ac_3}$	austenitization temperatures [in °C]
$c(\cdot), c_n$	parameter in heat transfer equations [in $\mathrm{m^2 \cdot K/J}$]
d_b	distance between two consecutive blanks in the RHF [in m]
F	pressure force [in N]
$\mathcal{F}(\cdot)$, $\mathcal{F}_\mathrm{a}(\cdot)$	solutions of initial value problems of the blank temperature [in °C]
l_f	furnace length [in m]
$\mathrm{M_f}$	martensite finish temperature [in °C]
p_b	probability of a blank transfer to a subsequent production unit
$p_{\mathrm{b},m}$	probability whether a blank pair is processed at the m-th position of the RHF
$\dot{Q}_{\mathrm{b},n}$	heat flow rate from n-th furnace segment to the blanks [in kW]
s, s_e	position in furnace, position for blank exiting [in m]
$s_{\mathrm{b},n}$	position of n-th blank at the sampling instants [in m]
$t_\mathrm{c}, t_\mathrm{c,b}$	cycle time, base cycle time (of the fastest subsystem) [in s]
$t_\mathrm{f}, t_\mathrm{p}, t_\mathrm{ph}, t_\mathrm{t}$	time in furnace, pressing time, partial hardening time, transfer time [in s]
T_a	ambient temperature [in °C]
$T_\mathrm{b}, T_\mathrm{b,h}, T_\mathrm{b,s}$	blank temperature, blank temperature in the hard part, blank temperature in the soft part [in °C]
T_e	temperature at the furnace exit [in °C]
$T_{\mathrm{f},n}, T_\mathrm{f}(s)$	furnace segment temperatures, temperature profile [in °C]
$T_{\mathrm{t},n}$	tool temperatures [in °C]
$\dot{V}$	cooling agent flow [in $\mathrm{m^3/s}$]
v, v_e	transportation velocity, exiting velocity of a furnace [in $\mathrm{m/s}$]
W_h	heating power [in kW]

Scalars and Constants

α	constant used in stability conditions, or scaling parameter
β	confidence level
$\gamma^{\boldsymbol{n}}$	weighting factor for the local cost criteria
$\epsilon, \varepsilon, \varrho$	scaling parameters
θ_k	Markov state at time k
$\boldsymbol{\theta}_{k,l}$	Markov state sequence of length $l+1$: $(\theta_k; \ldots; \theta_{k+l})$
λ	eigenvalue or decrease rate
$\lambda_{\mathrm{f}}, \lambda_{\mathrm{r}}$	failure rate, repair rate
τ	integration variable
$\varphi_{l,j}$	model parameters of the disturbance models
$\mathfrak{a}_{i,l}, \mathfrak{b}_{i,l}, \mathfrak{g}_{i,l}, \mathfrak{p}_l$	weighting factors for l-th polytope vertex of $\mathbb{A}_i, \mathbb{B}_i, \mathbb{G}_i$, and $\mathbb{P}$
N, N_{e}	finite prediction horizon, finite estimation horizon
o	offset of the subsystem time index $k^{\boldsymbol{m}}$
$p_{i,m}$	probability for a transition of the Markov state from i to m

Sets and Spaces

Θ	set of Markov states
$\boldsymbol{\Theta}_{k,l}$	set of all Markov state sequences from time k to $k+l$
Ξ	set of the augmented system vector $\xi[k]$: $\Xi := \mathbb{X} \times \mathbb{U} \times \mathbb{W}$
ψ	set of matrices ψ_i
$\mathbb{A}_i, \mathbb{B}_i, \mathbb{G}_i$	matrix polytopes that contain all possible values of A_i, B_i, G_i
$\mathbb{E}$	error set
$\mathcal{E}, \mathcal{E}(\Lambda), \mathcal{E}(c, \Lambda)$	ellipsoid, ellipsoid with center c and shape matrix Λ
$\mathscr{E}, \acute{\mathscr{E}}$	sets of shape matrices $\mathscr{E}_i$ and $\acute{\mathscr{E}}_i$
$\mathscr{G}$	set of auxiliary matrix variables $\mathscr{G}_i$ or $\mathscr{G}_{i,[l]}$

$\mathcal{I}$	index set
$\mathbb{N}, \mathbb{N}^0$	set of natural numbers, set of natural numbers including 0
$\mathbb{N}_n, \mathbb{N}_n^0, \mathbb{N}_{n_1:n_2}$	natural numbers from 1 to n, from 0 to n, and from n_1 to n_2
$\mathscr{P}$	set of Markov state i dependent Lyapunov matrices $\mathscr{P}_i$
$\mathscr{Q}$	set of Markov state i dependent inverse Lyapunov matrices $\mathscr{Q}_i$
$\mathbb{R}$	set of real numbers
$\mathbb{S}_n, \mathbb{S}_n^{>0}, \mathbb{S}_n^{\geq 0}$	sets of symmetric, symmetric positive definite, and positive semi-definite $n \times n$ matrices
$\mathbb{U}$	input space
$\mathbb{V}$	measurement noise set
$\mathbb{W}$	set of possible disturbances $w[k]$
$\mathbb{X}$	state space
$\mathbb{Y}$	output space
$\mathscr{Y}$	set of auxiliary matrix variables $\mathscr{Y}_i$ or $\mathscr{Y}_{i,[l]}$

Vectors and Matrices

$\mathbf{0}, \mathbf{0}_n, \mathbf{0}_{n \times m}$	zero matrices of appropriate or indicated dimension (a single index references the square $n \times n$-matrix)
$\mathbf{1}, \mathbf{1}_n, \mathbf{1}_{n \times m}$	matrices with all elements equal to 1 of appropriate or indicated dimension (a single index references the square $n \times n$-matrix)
Λ	shape matrix of an ellipsoid
$\mu[k]$	probability distribution of a Markov chain at time k
$\nu[k]$	plant model mismatch, measurement noise
$\xi[k]$	augmented system vector containing $x[k], u[k], w[k]$
Σ	covariance matrix
$\phi_{(\cdot, \cdot)}^{(m)}$	auxiliary iteration matrices
$\Phi_{(\cdot, \cdot)}$	auxiliary matrices for cost prediction
Ψ	cost terms that are independent of the optimization variable

ψ_i	auxiliary variable
A_i, B_i, G_i	system matrices of a lin. dynamical system for Markov state i
$A_{i,[l]}, B_{i,[l]}, G_{i,[l]}$	l-th spanning vertex of the polytopes $\mathbb{A}_i, \mathbb{B}_i, \mathbb{G}_i$ for A_i, B_i, G_i
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{G}$	matrices that define the predicted state sequence $\bar{\boldsymbol{x}}[1]$
$\boldsymbol{A}_\mathrm{c}, \boldsymbol{B}_\mathrm{c}, \boldsymbol{G}_\mathrm{c}$	matrices that define the predicted state sequence $\bar{\boldsymbol{x}}[k]$
$\boldsymbol{B}_i'[j], \boldsymbol{G}_i'[j]$	auxiliary matrices
b	auxiliary vector
c	general variable
C_i, D_i, F_i	matrices defining the outputs for Markov state i
$\mathscr{E}_i^{\boldsymbol{m}}, \acute{\mathscr{E}}_i^{\boldsymbol{m}}$	shape matrices of error and output set of the m-th subsystem
$e[k], \check{e}[k]$	prediction and estimation error
E_i	shape matrix of an ellipsoidal error set
$\boldsymbol{f}$	auxiliary vector
$\mathscr{G}_i, \mathscr{G}_{i,[l]}$	auxiliary matrix variable for Markov state i
$H_{(\cdot)}, \boldsymbol{H}_{(\cdot)}, h_{(\cdot)}, \boldsymbol{h}_{(\cdot)}$	matrices and vectors defining linear constraints on states, inputs, and outputs or the corresponding trajectories
I, I_n	identity matrix of appropriate dimension and of dimension $n \times n$
$K_i, \hat{K}_i$	state feedback matrix and disturbance feedback matrix for Markov state i
L	matrix defining a decrease rate for the Lyapunov function $V(\cdot)$
L_k	Kalman filter gain matrix
$\mathcal{L}(\cdot)$	auxiliary matrix function
M	auxiliary matrix
$\mathscr{M}$	auxiliary matrix variable
P	transition probability matrix
$\mathscr{P}_i$	Lyapunov matrices
$P_{[l]}$	vertices spanning a polytope containing the matrix $P[k]$

Q, Q_i	state cost weights, possibly depending on the Markov state i
$q[k]$	vector defining the linear part of the cost function of a QP
$\boldsymbol{Q}$	cost weighting matrix for the state sequence $\boldsymbol{x}[k]$
$Q_{(\cdot)}(j), \boldsymbol{Q}_{(\cdot)}(j)$	auxiliary matrices for the calculation of $W[k]$
$\mathscr{Q}_i$	shape matrix of $\mathcal{E}_i$ and inverse of Lyapunov matrices $\mathscr{P}_i$
$\breve{\mathscr{Q}}_{i,m}, \breve{\mathscr{Q}}_{y,m}$	shape matrices of tightened RCIS and output set
R, R_i	input cost weights, possibly depending on the Markov state i
$\boldsymbol{R}$	cost weighting matrix for the input sequence $\boldsymbol{u}[k]$
S_i	output cost weights, possibly depending on the Markov state i
$\mathscr{T}$	transformation matrix
$T_\mathrm{y}, \boldsymbol{T}_\mathrm{y}$	matrices that "extract" all outputs from the augmented state vector by $y[k] = T_\mathrm{y}\hat{x}[k]$
$u[k]$	input at time k
$\boldsymbol{u}[k]$	input sequence for the horizon N: $\boldsymbol{u}[k] := \begin{bmatrix} u^\top[0] & \cdots & u^\top[N-1] \end{bmatrix}^\top$
$u_i[k]$	input as a function of the Markov state i at time k
$v[k]$	affine input for a closed-loop prediction at time k
$W[k]$	matrix defining the quadratic part of the cost function of a QP
$w[k], \tilde{w}[k]$	disturbance at time k, zero mean disturbance at time k
$w_\mathrm{d}[k], w_\mathrm{p}[k]$	local disturbance and product-related disturbance at time k
$\boldsymbol{w}[k]$	disturbance sequence $\begin{bmatrix} w^\top[0] & \cdots & w^\top[N-1] \end{bmatrix}^\top$ for the horizon N
$x[k]$	system state at time k
$\boldsymbol{x}[k]$	state sequence for the horizon N: $\boldsymbol{x}[k] := \begin{bmatrix} x^\top[0] & \cdots & x^\top[N-1] \end{bmatrix}^\top$
$\bar{x}_i[j]$	conditional expectancy of the state predicted by j time steps
$y[k]$	output at time k
$\boldsymbol{y}[k]$	output sequence for the horizon N: $\boldsymbol{y}[k] := \begin{bmatrix} y^\top[0] & \cdots & y^\top[N-1] \end{bmatrix}^\top$
$\mathscr{Y}_i, \mathscr{Y}_{i,[l]}$	auxiliary matrix variable for Markov state i
$z[k]$	measurements at time k
$\mathscr{Z}$	matrix variable, inverse of L

List of Abbreviations

ARIMA   autoregressive integrated moving average

BMI   bi-linear matrix inequality

CF   chamber furnace

CoE   costs of expected value

DMPC   distributed model predictive control

EKF   extended Kalman filters

EoC   expected value of costs

FEM   finite element method

JML   jump Markov linear

JMLS   jump Markov linear system

JMS   jump Markov system

LMI   linear matrix inequality

LPV   linear parameter varying

LTI   linear time-invariant

MILP   mixed-integer linear programming

MPC   model predictive control

MQB   Modularer Querbaukasten

MSS   mean square stability

MTBF   mean time between failures

MTTR   mean time to repair

PDF   probability density function

PLC   programmable logic controller

PMSS	practical mean square stability, practically mean square stable
QP	quadratic program
QCQP	quadratically constrained quadratic program
RCIS	robust control invariant set
RHE	receding horizon estimation
RHF	roller hearth furnace
SDP	semidefinite programming
SoV	stream of variation
TP	transition probabilities
ZOH	zero-order-hold

References

[1] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*. Dover Publications, 1964.

[2] R. Akella and P. R. Kumar, "Optimal control of production rate in a failure prone manufacturing system," *Tr. on Automatic Control*, vol. 31, no. 2, pp. 116–126, 1986.

[3] A. Alessandri, M. Baglietto, and G. Battistelli, "Receding-horizon estimation for switching discrete-time linear systems," *Tr. on Automatic Control*, vol. 50, no. 11, pp. 1736–1748, 2005.

[4] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753 – 1765, 2008.

[5] B. D. O. Anderson and J. B. Moore, *Optimal filtering*. Dover Publications, 2005.

[6] C. Andrieu, M. Davy, and A. Doucet, "A particle filtering technique for Jump Markov Systems," in *European Signal Processing Conf.*, 2002, pp. 1–4.

[7] P. J. Antsaklis and A. N. Michel, *Linear Systems*. Birkhäuser, 2006.

[8] L. Asselborn, D. Groß, and O. Stursberg, "Control of uncertain nonlinear systems using ellipsoidal reachability calculus," in *Symp. on Nonlinear Control Systems*, 2013, pp. 50 – 55.

[9] L. Asselborn, M. Jilg, and O. Stursberg, "Control of uncertain hybrid nonlinear systems using particle filters," in *Conf. on Analysis and Design of Hybrid Systems*, 2012, pp. 436–441.

[10] L. Asselborn and O. Stursberg, "On probabilistic control of switched affine systems with state chance constraints," *Accepted for: Nonlinear Analysis: Hybrid Systems*, 2017.

[11] T. I. Bø and T. A. Johansen, "Scenario-based fault-tolerant model predictive control for diesel-electric marine power plant," in *OCEANS - Bergen, MTS/IEEE*, 2013, pp. 1–5.

[12] F. Balduzzi, "Hybrid system model of production work-cells and its optimal control subject to logical constraints," in *Conf. on Emerging Technologies and Factory Automation*, vol. 1, 2001, pp. 455–463.

[13] D. Barcelli, "Decentralized and hierarchical model predictive control of networked systems," Ph.D. dissertation, University of Siena, 2012.

[14] E. Bauer, R. Adams, and D. Eustace, *Beyond Redundancy: How Geographic Redundancy Can Improve Service Availability and Reliability of Computer-Based Systems*. Wiley-IEEE Press, 2011.

[15] D. Bernardini and A. Bemporad, "Stabilizing model predictive control of stochastic constrained linear systems," *Tr. on Automatic Control*, vol. 57, no. 6, pp. 1468–1480, 2012.

[16] G. Böker and J. Lunze, "Stability and performance of switching Kalman filters," *Int. Journal of Control*, vol. 75, no. 16-17, pp. 1269–1281, 2002.

[17] L. Blackmore, A. Bektassov, M. Ono, and B. C. Williams, "Robust, optimal predictive control of jump Markov linear systems using particles," in *Hybrid Systems: Computation and Control*. Springer, 2007, vol. 4416, pp. 104–117.

[18] L. Blackmore, M. Ono, A. Bektassov, and B. C. Williams, "A probabilistic particle-control approximation of chance-constrained stochastic predictive control," *Tr. on Robotics*, vol. 26, no. 3, pp. 502–517, 2010.

[19] W. P. Blair and D. D. Sworder, "Feedback control of a class of linear discrete systems with jump parameters and quadratic cost criteria," *Int. Journal of Control*, vol. 21, no. 5, pp. 833–841, 1975.

[20] M. Blanke, M. Kinnaert, J. Lunze, and M. Staroswiecki, *Diagnosis and Fault-Tolerant Control*. Springer, 2006.

[21] E. K. Boukas, Q. Zhang, and G. Yin, "Robust production and maintenance planning in stochastic manufacturing systems," *Tr. on Automatic Control*, vol. 40, no. 6, pp. 1098–1102, 1995.

[22] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis: Forecasting and Control*. Wiley, 2008.

[23] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994.

[24] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2009.

[25] E. F. Camacho, T. Alamo, and D. M. de la Peña, "Fault-tolerant model predictive control," in *Conf. on Emerging Technologies and Factory Automation*, 2010, pp. 1–8.

[26] J. Campos, C. Seatzu, and X. Xie, Eds., *Formal Methods in Manufacturing*. CRC Press, 2014.

[27] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, 2008.

[28] S. M. O. Cáceres, "Plantwide optimizing control for the continuous bio-ethanol production process," Ph.D. dissertation, TU Berlin, 2010.

[29] J. Cheng and F. Liu, "Feedback predictive control based on periodic invariant set for Markov jump systems," *Circuits, Systems, and Signal Processing*, vol. 34, no. 8, pp. 2681–2693, 2015.

[30] D. Chilin, J. Liu, D. M. de la Peña, P. D. Christofides, and J. F. Davis, "Detection, isolation and handling of actuator faults in distributed model predictive control systems," *Journal of Process Control*, vol. 20, no. 9, pp. 1059 – 1075, 2010.

[31] S. Chitraganti, S. Aberkane, C. Aubrun, and T. Darure, "A receding horizon control of Markov jump linear systems with imperfect state information and probabilistic state constraints," in *Conf. on Control and Fault-Tolerant Systems*, 2016, pp. 611–617.

[32] S. Chitraganti, S. Aberkane, C. Aubrun, G. Valencia-Palomo, and V. Dragan, "On control of discrete-time state-dependent jump linear systems with probabilistic constraints: A receding horizon approach," *Systems & Control Letters*, vol. 74, pp. 81 – 89, 2014.

[33] H. Chizeck and Y. Ji, "Optimal quadratic control of jump linear systems with gaussian noise in discrete-time," in *Conf. on Decision and Control*, vol. 3, 1988, pp. 1989–1993.

[34] Y. Cho, C. Cassandras, and W. Kwon, "Optimal control for steel annealing processes as hybrid systems," *Control Engineering Practice*, vol. 12, no. 10, pp. 1319 – 1328, 2004.

[35] P. D. Christofides, R. Scattolini, D. M. de la Peña, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers & Chemical Engineering*, vol. 51, pp. 21 – 41, 2013.

[36] J. Clobes, "Adaptive control strategies for the production of thermo-mechanically tailored products," Ph.D. dissertation, University of Kassel, 2013.

[37] E. F. Costa and J. B. R. do Val, "Stability of receding horizon control of Markov jump linear systems without jump observations," in *American Control Conf.*, vol. 6, 2000, pp. 4289–4293.

[38] O. L. V. Costa, E. O. A. Filho, E. K. Boukas, and R. P. Marques, "Constrained quadratic state feedback control of discrete-time Markovian jump linear systems," *Automatica*, vol. 35, no. 4, pp. 617 – 626, 1999.

[39] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*. Springer, 2005.

[40] J. do Val and T. Başar, "Receding horizon control of jump linear systems and a macroeconomic policy problem," *Journal of Economic Dynamics and Control*, vol. 23, no. 8, pp. 1099 – 1131, 1999.

[41] A. Doucet, N. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump Markov linear systems," *Tr. on Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

[42] D. Du, B. Jiang, and P. Shi, *Fault Tolerant Control for Switched Linear Systems*. Springer, 2015.

[43] C. Edwards, *Advanced Calculus of Several Variables*. Dover Publications, 1973.

[44] Y. Fang, K. A. Loparo, and X. Feng, "Stability of discrete time jump linear systems," *Journal of Mathematical Systems, Estimation, and Control*, vol. 5, no. 3, pp. 275 – 321, 1995.

[45] M. Farina and R. Scattolini, "Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems," *Automatica*, vol. 48, no. 6, pp. 1088 – 1096, 2012.

[46] A. Fletcher, S. Rangan, V. Goyal, and K. Ramchandran, "Causal and strictly causal estimation for jump linear systems: An LMI analysis," in *Conf. on Information Sciences and Systems*, 2006, pp. 1302–1307.

[47] S. Fujii, K. Urayama, K. Kashima *et al.*, "Machine-based modeling of conveyor-type flowshops with application to scheduling and temperature control in slab reheating furnace," in *Int. Conf. on Control Applications*, 2010, pp. 2059–2064.

[48] A. Goncalves, A. Fioravanti, and J. Geromel, "$\mathcal{H}_\infty$ filtering of discrete-time Markov jump linear systems through linear matrix inequalities," *Tr. on Automatic Control*, vol. 54, no. 6, pp. 1347–1351, 2009.

[49] S. Gottlieb, D. Ketcheson, and C.-W. Shu, "High order strong stability preserving time discretizations," *Journal of Scientific Computing*, vol. 38, no. 3, pp. 251–289, 2009.

[50] M. Grimble, *Robust Industrial Control Systems: Optimal Design Approach for Polynomial Systems*. Wiley, 2006.

[51] Y. Guo and B. Huang, "Moving horizon estimation for switching nonlinear systems," *Automatica*, vol. 49, no. 11, pp. 3270 – 3281, 2013.

[52] S. Hackman and R. Leachman, "A general framework for modeling production," *Management Science*, vol. 35, no. 4, pp. 478–495, 1989.

[53] K. Han and K. Wang, "Coordination and control of batch-based multistage processes," *Journal of Manufacturing Systems*, vol. 32, no. 2, pp. 372 – 381, 2013.

[54] D. Helbing, *Nonlinear Dynamics of Production Systems*. Wiley, 2005, ch. Modeling and Optimization of Production Process: Lessons from Traffic Dynamics, pp. 85–105.

[55] M. Henández-Mejìas, A. Sala, C. Arino, and A. Querol, "Fault-tolerant predictive control for Markov linear systems," in *Conf. on Automation, Quality and Testing, Robotics*, 2014, pp. 1–6.

[56] M. Henández-Mejìas, A. Sala, A. Querol, and C. Ariño, "Multiple-horizon predictive control for Markov/switched linear systems," *IFAC-PapersOnLine*, vol. 48, no. 23, pp. 230 – 235, 2015.

[57] N. Henze, *Stochastik für Einsteiger - Eine Einführung in die faszinierende Welt des Zufalls*. Vieweg + Teubner Verlag, 2012.

[58] B. Hochholdinger, P. Hora, H. Grass, and A. Lipp, "Simulation of the Press Hardening Process and Prediction of the Final Mechanical Material Properties," in *Int. Conf. and Workshop on Numerical Simulation of 3D Sheet Metal Forming Processes*, vol. 1383, no. 1, 2011, pp. 618–625.

[59] J. Hofer and J. Tan, "Extrudate temperature control with disturbance prediction," *Food Control*, vol. 4, no. 1, pp. 17 – 24, 1993.

[60] Q. Huang and J. Shi, "Stream of variation modeling and analysis of serial-parallel multistage manufacturing systems," *Journal of Manufacturing Science and Engineering*, vol. 126, pp. 611–618, 2004.

[61] M. Iqbal and J. Xiao, "A review on filtering techniques for switched control systems," *Journal of Convergence Information Technology*, vol. 8, no. 12, pp. 139–147, 2013.

[62] P. Jarosz, J. Kusiak, S. Małecki *et al.*, "A methodology for optimization in multistage industrial processes: A pilot study," *Mathematical Problems in Engineering*, vol. 2015, p. 10, 2015.

[63] J. Jerez, E. Kerrigan, and G. Constantinides, "A condensed and sparse QP formulation for predictive control," in *Conf. on Decision and Control and European Control Conf.*, 2011, pp. 5217–5222.

[64] M. Jilg, J. Tonne, and O. Stursberg, "Design of distributed $\mathcal{H}_2$-optimized controllers considering stochastic communication link failures," in *American Control Conf.*, 2015, pp. 3540–3545.

[65] H. Karbasian and A. Tekkaya, "A review on hot stamping," *Journal of Materials Processing Technology*, vol. 210, no. 15, pp. 2103 – 2118, 2010.

[66] S. Kargar, K. Salahshoor, and M. Yazdanpanah, "Integrated nonlinear model predictive fault tolerant control and multiple model based fault detection and diagnosis," *Chemical Engineering Research and Design*, vol. 92, no. 2, pp. 340 – 349, 2014.

[67] E. Kerrigan, "Robust constraint satisfaction: Invariant sets and predictive control," Ph.D. dissertation, University of Cambridge, 2000.

[68] M. Kögel and R. Findeisen, "Discrete-time robust model predictive control for continuous-time nonlinear systems," in *American Control Conf.*, 2015, pp. 5091–5096.

[69] B. R. Knudsen, J. H. Brusevold, and B. Foss, "An exact penalty-function approach to proactive fault-tolerant economic MPC," in *European Control Conf.*, 2016, pp. 758 – 763.

[70] M. Kothare, V. Balakrishnan, and M. Morari, "Robust constrained model predictive control using linear matrix inequalities," *Automatica*, vol. 32, no. 10, pp. 1361 – 1397, 1996.

[71] B. Kouvaritakis and M. Cannon, *Model Predictive Control - Classical, Robust and Stochastic*. Springer, 2016.

[72] W. Krzanowski and F. Marriott, *Multivariate Analysis: Distributions, Ordination and Inference*. Wiley Interscience, 1994.

[73] H. Kuiken, *Mathematical Modelling of Industrial Processes*. Springer, 1992, ch. Mathematical Modelling of Industrial Processes, pp. 1–63.

[74] A. B. Kurzhanski and I. Vàlyi, *Ellipsoidal calculus for estimation and control*. Birkhäuser, 1997.

[75] D. Landgrebe, N. Pierschel, M. Alsmann *et al.*, "Intelligent process control for press hardening," in *Int. Conf. on Hot Sheet Metal Forming of High-Performance Steel*, 2015, pp. 419–428.

[76] L. Lao, M. Ellis, and P. D. Christofides, "Proactive fault-tolerant model predictive control: Concept and application," in *American Control Conf.*, 2013, pp. 5140–5145.

[77] J. Lechler, "Grundlegende Untersuchungen zur Beschreibung und Modellierung des Werkstoffverhaltens von presshärtbaren Bor-Manganstählen," Ph.D. dissertation, Universität Erlangen-Nürnberg, 2009.

[78] J. H. Lee and N. Ricker, "Extended Kalman filter based nonlinear model predictive control," in *American Control Conf.*, 1993, pp. 1895–1899.

[79] E. Lefeber, *Nonlinear Dynamics of Production Systems*. Wiley, 2005, ch. Nonlinear Models for Control of Manufacturing Systems, pp. 71–83.

[80] W. Li and Y. Jia, "Rao–blackwellised particle filtering and smoothing for jump Markov nonlinear systems with mode observation," *IET Signal Processing*, vol. 7, no. 4, pp. 327–336, 2013.

[81] Y. Li, W. Zhang, and X. Liu, "Stability of nonlinear stochastic discrete-time systems," *Journal of Applied Mathematics*, vol. 2013, 2013, article ID 356746.

[82] Y. Liu, Y. Yin, F. Liu, and K. L. Teo, "Constrained MPC design of nonlinear Markov jump system with nonhomogeneous process," *Nonlinear Analysis: Hybrid Systems*, vol. 17, no. 0, pp. 1 – 9, 2015.

[83] J. Lu, Y. Xi, D. Li, and L. Cen, "Probabilistic constrained stochastic model predictive control for Markovian jump linear systems with additive disturbance," in *IFAC World Congress*, vol. 19, no. 1, 2014, pp. 10 469–10 474.

[84] J. Lu, D. Li, and Y. Xi, "Constrained model predictive control synthesis for uncertain discrete-time Markovian jump linear systems," *Control Theory Applications, IET*, vol. 7, no. 5, pp. 707–719, 2013.

[85] J. Z. Lu, "Challenging control problems and emerging technologies in enterprise optimization," *Control Engineering Practice*, vol. 11, no. 8, pp. 847 – 858, 2003.

[86] H. Ma and Y. Jia, "$\mathcal{H}_\infty$ control for discrete-time nonlinear Markov jump systems with multiplicative noise and sector constraint," *Int. Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2347–2364, 2014.

[87] S. Ma, J. Xiong, V. A. Ugrinovskii, and I. R. Petersen, "Robust decentralized stabilization of Markovian jump large-scale systems: A neighboring mode dependent control approach," *Automatica*, vol. 49, no. 10, pp. 3105 – 3111, 2013.

[88] J. M. Maciejowski, *Predictive Control with Constraints*. Prentice-Hall, 2002.

[89] J. M. Maciejowski, A. L. Visintini, and J. Lygeros, "NMPC for complex stochastic systems using a Markov chain Monte Carlo approach," in *Assessment and Future Directions of Nonlinear Model Predictive Control*. Springer, 2007, vol. 358, pp. 269–281.

[90] J. M. Maciejowski and X. Yang, "Fault tolerant control using Gaussian processes and model predictive control," in *Conf. on Control and Fault-Tolerant Systems*, 2013, pp. 1–12.

[91] J. M. Maestre and R. R. Negenborn, *Distributed Model Predictive Control Made Easy*. Springer, 2013.

[92] L. Magni and R. Scattolini, "Stabilizing decentralized model predictive control of nonlinear systems," *Automatica*, vol. 42, no. 7, pp. 1231 – 1236, 2006.

[93] D. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967 – 2986, 2014.

[94] M. Medricky, "Sensitivity analysis for prediction of sheet removal temperature within press hardening process and experimental validation," Ph.D. dissertation, Leibniz Universität Hannover, 2012.

[95] M. Merklein, M. Wieland, M. Lechner, S. Bruschi, and A. Ghiotti, "Hot stamping of boron steel sheets with tailored properties: A review," *Journal of Materials Processing Technology*, vol. 228, pp. 11 – 24, 2016.

[96] P. Mhaskar, J. Liu, and P. Christofides, *Fault-Tolerant Process Control - Methods and Applications*. Springer, 2013.

[97] M. Naderi, A. Saeed-Akbari, and W. Bleck, "The effects of non-isothermal deformation on martensitic transformation in 22MnB5 steel," *Materials Science and Engineering: A*, vol. 487, no. 1–2, pp. 445 – 455, 2008.

[98] R. Neugebauer, F. Schieck, S. Polster *et al.*, "Press hardening — an innovative and challenging technology," *Archives of Civil and Mechanical Engineering*, vol. 12, no. 2, pp. 113 – 118, 2012.

[99] R. H. Nyström, R. Franke, I. Harjunkoski, and A. Kroll, "Production campaign planning including grade transition sequencing and dynamic optimization," *Computers & Chemical Engineering*, vol. 29, no. 10, pp. 2163 – 2179, 2005.

[100] R. H. Nyström, I. Harjunkoski, and A. Kroll, "Production optimization for continuously operated processes with optimal operation and scheduling of multiple units," *Computers & Chemical Engineering*, vol. 30, no. 3, pp. 392 – 406, 2006.

[101] M. Ohshima, H. Ohno, I. Hashimoto *et al.*, "Model predictive control with adaptive disturbance prediction and its application to fatty acid distillation column control," *Journal of Process Control*, vol. 5, no. 1, pp. 41 – 48, 1995.

[102] M. D. Oliveira, J. C. Geromel, and J. Bernussou, "Extended $\mathcal{H}_2$ and $\mathcal{H}_\infty$ norm characterizations and controller parametrizations for discrete-time systems," *Int. Journal of Control*, vol. 75, no. 9, pp. 666–679, 2002.

[103] B.-G. Park and W. H. Kwon, "Robust one-step receding horizon control of discrete-time Markovian jump uncertain systems," *Automatica*, vol. 38, no. 7, pp. 1229 – 1235, 2002.

[104] P. Patrinos, P. Sopasakis, H. Sarimveis, and A. Bemporad, "Stochastic model predictive control for constrained discrete-time Markovian switching systems," *Automatica*, vol. 50, no. 10, pp. 2504 – 2514, 2014.

[105] D. L. Pepyne and C. G. Cassandras, "Optimal control of hybrid systems in manufacturing," *Proceedings of the IEEE*, vol. 88, no. 7, pp. 1108–1123, 2000.

[106] M. C. Perez, "Untersuchung und Entwicklung eines thermodynamischen Modells der Temperaturentwicklung in einem Rollenherdofen," Bachelor thesis, Hochschule Hannover, 2017.

[107] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733 – 764, 2003.

[108] P. Åkerström, "Modelling and simulation of hot stamping," Ph.D. dissertation, Luleå University of Technology, 2006.

[109] J. B. Rawlings and B. T. Stewart, "Coordinating multiple optimization-based controllers: New opportunities and challenges," *Journal of Process Control*, vol. 18, no. 9, pp. 839 – 845, 2008.

[110] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design.* Nob Hill Publishing, 2009.

[111] *Understanding Series and Parallel Systems Reliability*, ser. START - Selected Topics in Assurance Related Technologies, vol. 11, no. 5, Reliability Analysis Center, 2004-2005.

[112] A. Rorot, "Prozessoptimierung und Bewertung eines optischen und taktilen Messsystems im Warmumformprozess mit segmentierten Werkzeugen," Master thesis, University of Kassel, 2013.

[113] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723 – 731, 2009.

[114] K. Schaab, J. Hahn, M. Wolkov, and O. Stursberg, "Robust control for voltage and transient stability of power grids relying on wind power," *Control Engineering Practice*, vol. 60, pp. 7–17, 2017.

[115] T. Schönbach and M. Kerausch, "Effective usage of simulation techniques for the layout of hotforming processes," in *Int. Conf. on Hot Sheet Metal Forming of High-Performance Steel*, 2009, pp. 323–330.

[116] L. E. Schneegans, "Prädiktive Regelung für lineare sprungfähige Markov-Systeme am Beispiel eines Warmumformprozesses," Master thesis, University of Kassel, 2016.

[117] G. Seber and A. Lee, *Linear Regression Analysis*. Wiley, 2003.

[118] J. Shi and S. Zhou, "Quality control and improvement for multistage systems: A survey," *IIE Transactions*, vol. 41, no. 9, pp. 744–753, 2009.

[119] P. Shi, E. Boukas, S. Nguang, and X. Guo, "Robust disturbance attenuation for discrete-time active fault tolerant control systems with uncertainties," *Optimal Control Applications and Methods*, vol. 24, no. 2, pp. 85–101, 2003.

[120] P. Shi and F. Li, "A survey on Markovian jump systems: Modeling and design," *Int. Journal of Control, Automation and Systems*, vol. 13, no. 1, pp. 1–16, 2015.

[121] P. Siebert, M. Alsmann, and H.-J. Watermeier, "Influence of different heating technologies on the coating properties of hot-dip aluminized 22MnB5," in *Int. Conference on Hot Sheet Metal Forming of High-Performance Steel*, vol. 3, 2011, pp. 315 – 322.

[122] D. Simon, *Optimal State Estimation: Kalman, H-Infinity, and Nonlinear Approaches*. Wiley, 2006.

[123] Y. Song, S. Liu, and G. Wei, "Constrained robust distributed model predictive control for uncertain discrete-time Markovian jump linear system," *Journal of the Franklin Institute*, vol. 352, no. 1, pp. 73 – 92, 2015.

[124] Y. Song, H. Lou, and S. Liu, "Distributed model predictive control with actuator saturation for Markovian jump linear system," *Journal of Automatica Sinica*, vol. 2, no. 4, pp. 374–381, 2015.

[125] K. Steinhoff *et al.*, "Tailored X – Maßgeschneiderte Produkte brauchen maßgeschneiderte Fertigungsprozesse," in *Umformtechnik im Spannungsfeld zwischen Plastomechanik und Werkstofftechnik*, K. Steinhoff and R. Kopp, Eds. GRIPS Media GmbH, 2008, pp. 217–231.

[126] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Systems & Control Letters*, vol. 59, no. 8, pp. 460 – 469, 2010.

[127] H. Sun and L. Yan, "Robust $\mathcal{H}_\infty$ fuzzy control for nonlinear discrete-time stochastic systems with Markovian jump and parametric uncertainties," *Mathematical Problems in Engineering*, vol. 2014, p. 11, 2014.

[128] M. Suzuki, K. Katsuki, J. I. Imura *et al.*, "Modeling and real-time heating control of a reheating furnace using an advection equation," in *SICE Annual Conf.*, 2011, pp. 842–848.

[129] J. Tonne, "Verteilte Regelung verkoppelter linearer sprungfähiger Markov-Systeme," Master thesis, University of Kassel, 2014.

[130] J. Tonne, J. Clobes, M. Alsmann *et al.*, "Model-based optimization of furnace temperature profiles with regard to economic and ecologic aspects in hot stamping of 22MnB5," in *Int. Conf. on Hot Sheet Metal Forming of High-Performance Steel*, 2013, pp. 177 – 184.

[131] J. Tonne, M. Jilg, and O. Stursberg, "Constrained model predictive control of high dimensional jump Markov linear systems," in *American Control Conf.*, 2015, pp. 2993 – 2998.

[132] J. Tonne and O. Stursberg, "Constrained model predictive control of processes with uncertain structure modeled by jump Markov linear systems," in *Variable-Structure Approaches - Analysis, Simulation, Robust Control and Estimation of Uncertain Dynamic Processes*. Springer, 2016, pp. 335–361.

[133] ——, "Constraint robust model predictive control for jump Markov linear systems with additive disturbances," in *European Control Conf.*, 2016, pp. 1315 – 1321.

[134] ——, "Fast robust model predictive control for nonlinear jump Markov systems," in *20th IFAC World Congress*, 2017, pp. 9752–9758.

[135] A. N. Vargas, J. B. R. do Val, and E. F. Costa, "Receding horizon control of Markov jump linear systems subject to noise and unobserved state chain," in *Conf. on Decision and Control*, vol. 4, 2004, pp. 4381–4386.

[136] ——, "Optimality condition for the receding horizon control of Markov jump linear systems with non-observed chain and linear feedback controls," in *Conf. on Decision and Control and European Control Conf.*, 2005, pp. 7308–7313.

[137] A. N. Vargas, W. Furloni, and J. B. R. do Val, "Control of Markov jump linear systems with state and input constraints: A necessary optimality condition," in *Symposium on System, Structure and Control*, vol. 3, no. 1, 2007, pp. 250–255.

[138] ——, "Second moment constraints and the control problem of Markov jump linear systems," *Numerical Linear Algebra with Applications*, vol. 20, no. 2, pp. 357–368, 2013.

[139] B. Vroemen, H. van Essen, A. van Steenhoven, and J. Kok, "Nonlinear model predictive control of a laboratory gas turbine installation," *ASME. Journal of Engineering for Gas Turbines and Power*, vol. 121, no. 4, pp. 629 – 634, 1999.

[140] B.-C. Wang and J.-F. Zhang, "Distributed output feedback control of Markov jump multi-agent systems," *Automatica*, vol. 49, no. 5, pp. 1397 – 1402, 2013.

[141] L. Wang, B. Zhu, Q. Wang, J. Meng, and Y. Zhang, "Production control and optimization of hot stamping line," in *Int. Conference on Hot Sheet Metal Forming of High-Performance Steel*, vol. 5, 2015, pp. 243–250.

[142] F. Weißel, "Stochastische modell-prädiktive Regelung nichtlinearer Systeme," Ph.D. dissertation, University of Karlsruhe, 2008.

[143] J. Wen and F. Liu, "Receding horizon control for constrained Markovian jump linear systems with bounded disturbance," *Journal of Dynamic Systems, Measurement, and Control*, vol. 133, no. 1, pp. 011 005–1 – 011 005–10, 2011.

[144] J. Wen, L. Peng, and S. Nguang, "Receding horizon control for constrained jump bilinear systems," *Int. Journal of Innovative Computing, Information and Control*, vol. 8, no. 12, pp. 8501–8514, 2012.

[145] Wikipedia, "Multivariate normal distribution," , accessed: 2014-11-07.

[146] Z. Yan and J. Wang, "Stochastic model predictive control of Markov jump linear systems based on a two-layer recurrent neural network," in *Int. Conf. on Information and Automation*, 2013, pp. 564–569.

[147] H. Yang, B. Jiang, and V. Cocquempot, "Fault tolerant control in hybrid systems: A brief survey," in *Symp. on Fault Detection, Supervision and Safety of Technical Processes*, 2009, pp. 229–234.

[148] Y. Yin and F. Liu, "Constrained predictive control of nonlinear stochastic systems," *Journal of Systems Engineering and Electronics*, vol. 21, no. 5, pp. 859–867, 2010.

[149] Y. Yin, Y. Liu, and H. R. Karimi, "A simplified predictive control of constrained Markov jump system with mixed uncertainties," *Abstract and Applied Analysis*, vol. 2014, Special Issue, pp. 1–7, 2014.

[150] Y. Yin, Y. Shi, and F. Liu, "Constrained model predictive control on convex polyhedron stochastic linear parameter varying systems," *Int. Journal of Innovative Computing, Information and Control*, vol. 9, no. 10, pp. 4193 – 4204, 2013.

[151] S. Yu, H. Chen, C. Böhm, and F. Allgöwer, *Enlarging the Terminal Region of NMPC with Parameter-Dependent Terminal Control Law*. Springer, 2009, pp. 69–78.

[152] H. Zhang, J. Wang, and Y. Shi, "Robust sliding-mode control for Markovian jump systems subject to intermittent observations and partially known transition probabilities," *Systems & Control Letters*, vol. 62, no. 12, pp. 1114 – 1124, 2013.

[153] L. Zhang and E.-K. Boukas, "Mode-dependent $\mathcal{H}_\infty$ filtering for discrete-time Markovian jump linear systems with partly unknown transition probabilities," *Automatica*, vol. 45, no. 6, pp. 1462 – 1467, 2009.

[154] L. Zhang, "$\mathcal{H}_\infty$ estimation for discrete-time piecewise homogeneous Markov jump linear systems," *Automatica*, vol. 45, no. 11, pp. 2570 – 2576, 2009.

[155] Y. Zhang and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, no. 2, pp. 229 – 252, 2008.

[156] Y. Zhang and S. Li, "Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes," *Journal of Process Control*, vol. 17, no. 1, pp. 37 – 50, 2007.

[157] S. Zhao and F. Liu, "State estimation in non-linear Markov jump systems with uncertain switching probabilities," *IET Control Theory and Applications*, vol. 6, no. 5, pp. 641 – 650, 2012.

[158] X. Zhong, H. He, H. Zhang, and Z. Wang, "Optimal control for unknown discrete-time nonlinear Markov jump systems using adaptive dynamic programming," *Tr. on Neural Networks and Learning Systems*, vol. 25, no. 12, pp. 2141–2155, 2014.

[159] J. Zhuge and M. Ierapetritou, "Integration of scheduling and control with closed loop implementation," *Industrial & Engineering Chemistry Research*, vol. 51, no. 25, pp. 8550–8565, 2012.

Large scale manufacturing systems are often run with constant process parameters although continuous and abrupt disturbances influence the process. To reduce quality variations and scrap, a closed-loop control of the process variables becomes indispensable. In this thesis, a modeling and control framework for multistage manufacturing systems is developed, in which the systems are subject to abrupt faults, such as component defects, and continuous disturbances. In this context, three main topics are considered: the development of a modeling framework, the design of robust distributed controllers, and the application of both to the models of a real hot stamping line. The focus of all topics is on the control of the product properties considering the available knowledge of faults and disturbances.

9 783737 604482 >