Generation Human Body Motion by the Centralized Networks

Dissertation zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften (Dr. rer. nat.)

Vorgelegt im Fachbereich Mathematics and Natural Sciences der Universität Kassel

Morozov Ivan

Datum der Disputation: 06.09.2023

Prof. Dr. W. Seiler
Prof. Dr. S. Vakulenko
Prof. Dr. G. Regensburger
Prof. Dr. F. Lindner

Datum der Disputation: 06.09.2023

Abstract

We consider continuous time recurrent networks as dynamical models for simulation of human body motions. These networks consist of a few of centers and many satellites. The centers evolve in time as periodical oscillators with different frequencies. The center states define the satellite neuron state by a radial basis function (RBF) network. To simulate different motions we adjust the parameter of the RBF networks.

Our network includes a switching module that allows to turn from a motion to another one. Simulations show that this model allows us to simulate complicated motions consisting of many different dynamical primitives.

Throughout all thesis our main model is to consider continuous time Hopfield-like recurrent networks as dynamical models for neural networks. We are interested in networks that contain n high-degree nodes preferably connected to a large number of N_s weakly connected satellites, a property that we call n/N_s-centrality. If the hub dynamics is slow, we obtain that the large time network dynamics is completely defined by the hub dynamics.

Moreover, such networks are maximally flexible and switchable, in the sense that they can switch from a globally attractive rest state to any structurally stable dynamics when the response time of a special controller hub is changed. In particular, we show that a decrease of the controller hub response time can lead to a sharp variation in the network attractor structure: we can obtain a set of new local attractors, whose number can increase exponentially with N, the total number of nodes of the nework.

These new attractors can be periodic or even chaotic. We provide an algorithm, which allows us to design networks with the desired switching properties, or to learn them from time series, by adjusting the interactions between hubs and satellites. Such switchable networks could be used as models for context dependent adaptation in functional genetics or as models for cognitive functions in neuroscience.

A fundamental difficulty that we encounter for other approaches when we try to identify the model by time-series is that we do not know true model. DMD is free of this difficulty and it is connected with Fourier analysis in a natural way, that allows us, in this thesis, use it to find main frequencies of human motion.

The DMD algorithm based on a singular value decomposition (SVD), has been established as a robust and powerful analysis tool, since SVD is realized in Matlab. The scope of its application was enlarged from fluid dynamics to many fields, such as epidemiology, financial trading, and computer graphics.

Acknowledgments

I am deeply thankful to my first advisor Professor Andreas Weber for his valuable advice for my career development and numbers of guidance during my whole Ph.D. program.

Thank you to my supervisor, Prof. Dr. Werner M. Seiler, for providing guidance and feedback throughout this project.

I have to pay my greatest gratefulness to Professor Sergey Vakulenko for his continuous guidance and abundant discussion for my Ph.D. research program, and especially for his technique support during my Ph.D. research program.

I would like to thank Professor Elena Sashina for her inspiration and discussion on the part of topic thesis.

I am grateful to all of my friends for their friendship in my life. Especially thanks to Vasiliy Tsvetkov for their accompany in St. Petersburg.

Finally, I want to pay my earnest gratitude to my mother, Maria Morozova, my father Sergey Morozov, who is the best parents in the world, for their selfless love and support to me during whole life, regardless of little return.

Contents

List of Figures

Chapter 1

Introduction

Networks of dynamically coupled elements have imposed themselves as models of complex systems in physics, chemistry, biology and engineering (66). The most studied propriety of networks is their topological structure. Structural features of networks are usually defined by the distribution of the number of direct connections a node has, or by various statistical properties of paths and circuits in the network (7). An important structure related property of networks is their scale-freeness (47; 46; 7; 14) often invoked as a paradigm of self-organization and spontaneous emergence of complex collective behaviour. In scale-free networks the fraction $P(k)$ of nodes in the network having k connections to other nodes (i.e. having degree k) can be estimated for large values of k as $P(k) \sim k^{-\gamma}$, where γ is a parameter whose value is typically in the range $2 < \gamma < 3$ (7). In such networks, the degree is extremely heterogeneous. In particular, there are strongly connected nodes that can be named hubs, or centers. The hubs communicate to each other directly, or via a number of weakly connected nodes. The weakly connected nodes that interact mainly with hubs can be called satellites. Scale-free networks have also nodes of intermediate connectivity. Networks that have only two types of nodes, strongly connected hubs and weakly connected satellites are known as bimodal degree networks (90). Because of the presence of a large number of hubs, scale-free or bimodal degree networks can be called centralized. Centralized connectivity has been found by functional imaging of brain activity in neuroscience (19), and also by large scale studies of the protein-protein interactions or of the metabolic networks in functional genetics (47; 46).

The centralized architecture was shown to be important for many emergent properties of networks. For instance, there has been a lot of interest in the resilience of networks with respect to attacks that remove some of their components (8). It was shown that networks with bimodal degree connectivity are resilient to simultaneous targeted and random attacks (90), whereas scale-free networks are robust with respect to random attacks, but sensitive to targeted attacks that are directed against hubs (21; 12). For this reason, the term "robust-yet-fragile" was coined in relation to scale-free networks (17). From a more dynamical perspective, a centralized architecture facilitates communications between hubs, stabilizes hubs by making them insensitive to noise (96; 95) and allows for hub synchronization even in the

absence of satellite synchronization (75; 74; 87).

Using these ideas that were successfully applied to genetics and ecology, we consider continuous time recurrent networks of an analogous topological structure as dynamical models for simulation of human body motions. These networks consist of a few of centers and many satellites. The centers evolve in time as oscillators with different frequencies. The center states define the satellite neuron state by a radial basis function (RBF) network. To simulate different motions we adjust the parameter of the RBF networks. The network also includes a switching module that allows to turn from one particular motion to another one. Due to this structure our model can simulate a large class of different motions with good accuracies, which depend on the oscillator frequencies. For each motion, we adjust these frequencies in order to obtain the best accuracy. Computations show that this model allows to simulate complicated motions consisting of many different dynamical primitives. Note the dynamical primitive method also permits simulate relatively simple motions, however, to describe complicated motions consisting of many different segments (which are themselves some elementary motions), DMP approximate each segment and after we must glue together these approximations.

Our model allows us to simulate sufficiently long motions by only two oscillators. However, in some hard cases (for example, if a motion consists of walking, running, kicking, punching, and knee kicking) we also decompose the whole motion into 2-4 segments and then for each segment we adjust the corresponding oscillator frequency. Then the application of our approximation algorithm allows to obtain automatically an uniform and smooth approximation of the whole motion. In the most of cases, it is sufficient to use $2-3$ oscillators and $100-200$ satellites.

Due to the network switching module we can use nonlinear oscillators and obtain a global network that can simulate a large class of different motions. So, one can assume that the motion of many animals can be controlled by networks of the same universal structure, and those ones consist of a few of oscillating centers and a hundred satellites. Note that the interaction between satellites and centers is organized according to the old Rome principle: "divide and rule", i.e., centers act on satellites and vice versa, but the satellites do not interact. We measure the state of a system (which should be identify) using a set of scalar observables, which are functions defined on state space. The values of these functions evolve in time. If the set of observations is large enough, one can try to find a model for the dynamics of the set of observations, and use this model system instead the original one.

Let us compare the approach based on centralized networks, proposed in this present thesis, and the classical method of Dynamic Motion Primitives (DMP), see, for example, (49) and (86).

To this end let us describe first the DMP method. Using dynamical systems theory (80), we use, in this thesis, systems of coupled nonlinear differential equations. Such approach is motivated by the fact that those systems are capable to form complex coordinated patterns. Among the many different forms of nonlinear systems (e.g., high-dimensional, weakly coupled, strongly coupled, chaotic, Hamiltonian,

dissipative), we use low-dimensional nonlinear systems, following the models phenomena of motor coordination (Kelso, 1995; Thelen and Smith, 1994). In this domain, there exists different approaches. The most known is Dynamical Primitive Model (DMP). The seminal work of Ijspeert et al. (5), (41) introduced the idea of Dynamic Movement Primitives (DMPs), which are primitives of motions. This model was applied for different motions such as bipedal locomotion or reach-and-grasp in arm movement. Moreover, motion primitives were applied to encode complex movements for robots and especially for humanoid robots. For example, rhythmic (e.g. walking) motions can be represented with DMPs as well as . multi-dimensional coordinated behaviors. Moreover, DMPs are used as compact motion representations for many other applications (5), (41), (26), (30), (71). Also rhythmic DMPs are studied (41), (45), (2) and an attention has also been given on Central Pattern Generators (CPGs) in locomotion control. This concept comes from animal locomotion, where CPGs are neural networks (32), (3), (55). From that point of view, rhythmic DMPs can be considered as a variant of a CPG.

Note that for any motions there is a connection between the transient part and the periodic movement and that there can be multiple transients, we would like to have a system of dynamcial equations that is able to simulate diifferent periodic patterns and transitions between them. To this end, we use linear ad nonlinear oscillators.

Nonlinear oscillators can be appleid in different fields of engineering (31), natural phenomena and control (31; 37; 51; 4). An interesting problem is how to design oscillators with desired limit cycle shapes (4).

In this thesis, we use the following basic property of simple nonlinear oscillators: depending on a control parameter, they can generate oscillations with different frequencies. This property allows to show that, to model motions of humanodi robots, it is sufficient to have a few of nonlinear oscillators plus a network of radial basic functions.

The both approaches use the same general representation, which, following (49), we write down as follows (see eqs (1), (2) in (49)):

$$\mathrm{d}s/\mathrm{d}t = Canonical(t, s), \tag{1.0.1}$$

$$\mathrm{d}y/\mathrm{d}t = Transform(t, y) + Perturbation(s). \tag{1.0.2}$$

The first equation is a time dependent dynamical system, the second one describes a transformation of trajectories of that dynamical system to desired trajectories y(t). Note that the term $P(s) = Perturbation(s)$ should be adapted to induce a desired behaviour in the system, i.e., reproduce a give trajectory (49). So, A DMP consists of two parts: the canonical system and the transformation system. While the canonical system is the DMP in time, the transformation system is the link between this DMP state and the robot. The transformation system can be easily adapted to a desired trajectory, i.e. by solving a standard regression problem. The canonical system determines the type of attractor which can be either discrete or periodic (49).

The DMP method uses $P(s)$ to attain the twofold goal: to represent trajectories tending to rest points and periodic trajectories. In fact, roughly speaking dynamics of any dissipative systems reduces to some transient trajectories and motions on local attractors. However, it is not so simple to represent simultaneuously transient dynamics as it was mentioned in (49). To attain this goal, we must use sufficiently sophisticated formulas for $P(s)$, which are based mainly on RBF (radial basis functions) and the fact, that RBF's are universal approximators.

In our centralized network approach, we use the same transformation system (1.0.2). However, we add a new idea in representation of canonical part (1.0.1). It is well known that many motions generated by dissipative systems consist of slow and fast components. Fast components can describe, for example, transient trajectories while slow components correspond to motions on local attractors. To represent such complex dynamics we can nonetheless use systems of oscillators (83).

In particular, in our approach we use usually two oscillators, one of higher frequency and another of low frequency, although one can take 3 and more oscillators for complicated target motions. This idea works effectively: we sharply simplify complicated formulas suggested in (86) and all transformation system takes the feedback form:

$$y = Perturbation(s). \tag{1.0.3}$$

Another important question concerning networks is how to push their dynamics from one region of the phase space to another or from one type of behaviour to another, briefly how to control the network dynamics (59; 88; 65; 23; 82; 73; 48; 72; 29; 102). Several authors used Kalman's results for linear systems to understand how network structure influences network dynamics controllability, and in particular how to choose the control nodes (59; 65; 23). As pointed out by (63; 56) several difficulties occur when one tries to apply these general results to real networks. Even for linear networks, the control of trajectories is nonlocal (88) and shortcuts are rarely allowed. As a result, even small changes of the network state may ask for control signals of large amplitude and energy (100). The control of nonlinear networks is even more difficult and in this case we have no general results. Nonlinear networks can have several co-existing attractors and it is interesting to find out how to push the state of the network from one attractor basin to another. The ability of networks to change attractor under the effect of targeted perturbations can be called switchability. In relation to this, the paper (76) has introduced the terminology "stable yet switchable" (76) meaning that the network remains stable given a context and is able to reach another stable state when a stimulus indicates a change of the context. It was shown, by numerical simulations, that centralized networks with bimodal degree distribution are more prone to (76) behavior than scale-free networks (76). Switchability is important for practical reasons, for instance in drug design. In such applications, one uses pharmaceutical action on nodes to push a network that functions in a pathological attractor (such pathological attractors were discussed in relation to cancer (36) or neurological disorders (85; 27)) to a healthy functioning mode, characterized by a different attractor. Numerical methods

to study switchability of linear (99) and nonlinear (22) networks were discussed in relation with drug design in cancer research. In theoretical biology, network switchability can be important for mathematical theories of genetic adaptation (67). If one looks at organisms as complex systems and model them by networks, then adaptation to changes in the environment can be described as switching the network from one attractor to another one with a higher fitness (67). An important question that is often asked with respect to tuning network dynamics is how many driver nodes are needed to control that dynamics. For linear networks, it was shown that this number is large if we aim to obtain a total control, which allows us to switch the network between any pair of states. This number can be as high as 80% for molecular regulatory networks (60). This fact, as emphasized in (99), contradicts empirical results about cellular reprogramming and about adaptive evolution. Much less nodes are needed if instead of full controlability one wants switching between specific pairs of unexpected and desired states (99). This concept, named "transittability" in (99), is very similar to our switchability, but was studied only for linear systems.

In this thesis, we study dynamical properties of large nonlinear networks with centralized architecture. We consider continuous time versions of the Hopfield model of recurrent neural networks (34) with a large number N of neurons. The Hopfield model is based on the two-states McCullogh and Pitts formal neuron and uses symmetrical weight matrices to specify interactions between neurons. Like to the Hopfield version, we use a thresholding function to describe switching between the two neuron states, active and inactive. However, contrary to the original Hopfield version, we do not impose symmetrical interactions between neurons, in other words our weight matrix is not necessarily symmetric. This model has been successfully used to describe associative memories (34), neural computation (35; 62), disordered systems in statistical physics (89), neural activity (58; 27) and also to investigate space-time dynamics of gene networks in molecular biology (98). The choice of such type of dynamics is motivated by the existence of universal approximation results for multilayered perceptrons (see, for example, (13)). In particular, we have shown elsewhere that networks with Hopfield-type dynamics can approximate any structurally stable dynamics, including reaction-diffusion biochemical networks also largely used in biology (95).

Our aim is to study analytically the ability of a network with centralized architecture to be switchable. We employ a special notion of centrality. Many biological networks exhibit so-called dissortative mixing, i.e., high-degree nodes are preferably connected to low-degree nodes (50). We will consider networks with n strongly connected hubs. We also assume that each hub is under the action of at least N_s weakly connected satellites, that on turn receive actions from all the hubs. For large networks, N_s increases at least as fast as a power of N, $N_s > c_0 N^\theta$ where $c_0 > 0$, $0 < \theta < 1$ are constants and N is the total number of nodes. We call this property n/N_s-centrality. This network architecture ensures a large number of feed-back loops that produce complex dynamics. Furthermore, the dissortative connectivity implies functional heterogeneity of the hubs and satellites. The hubs play the role of controllers and

the satellites sustain the feedback loops needed for attractor multiplicity. The large number of satellites guarantees a sufficient flexibility of the network dynamics and also buffer the perturbations transmitted to the hubs. This principle applies well to gene networks. The hubs in such networks can be the transcription factors, which are stabilized by numerous interactions with non-coding RNAs that represent the satellites (57). In addition to structural conditions, we will consider a special correlation between time scales and connectivity of the nodes: the hubs have slow response, whereas the satellites respond rapidly. This condition is natural for many real networks. The hubs have to cope with multiple tasks, therefore they must have more complex interaction than the satellites. Consequently, the hubs need more resources to be produced, decomposed, and react with other nodes, therefore their dynamics is slow. This property is obvious for gene networks, where transcription factors are complex proteins, much larger and more stable than the non-coding RNAs.

Our first result is valid without conditions on the structure and depends only on the condition on the timescales. We assume that there exist $n << N$ slow nodes, whereas all the remaining ones are fast. Then, the dynamics of the network can be reduced to n variables. We prove the existence of an inertial manifold of dimension n, which completely captures all network dynamics for large times. We recall that the fundamental concept of inertial manifold was introduced for infinite dimensional and multidimensional systems. The inertial manifolds are globally attracting invariant ones (92). The large time dynamics of a system possessing an inertial manifold, is defined by a smooth vector field F of relatively small dimension, so-called inertial form. All attractors lie on inertial manifold (92).

The second result holds under the structural assumption that the network is n/N_s-central. Under this condition, we show that the inertial forms F obtained from such networks are dense in the set of all smooth vector fields of dimension n. This implies that given a certain combination of attractors defined by vector fields Q_i we can construct a centralized network that exhibits a combination of attractors that is topologically equivalent to the one given dynamics. Furthermore, we show that n/N_s-central networks can exhibit "maximal switchability". By changing a control parameter ξ, which determines the response time of a single network hub ("controller" hub), we can sharply change the network attractor. For instance we can switch from a situation when the network has a single rest point for $\xi > \xi_0$ to a situation when the network has a complicated global attractor for $\xi < \xi_0$, including a number of local attractors, which may be periodic or chaotic. The network state tends to the corresponding local attractor depending on the initial state of the control hub. This result shows in an analytical and rigorous way how nonlinear networks can be switched by only one control node. The possibility of switching nonlinear networks by a small number of nodes is crucial in theories of genetic adaptation. Indeed, phenomenological theories predict and empirical data confirm that the main part of the adaptive evolution process consists in only a few mutations producing large fitness changes (67).

Our third result proves, in an analytical way, that the number of rest point local attractors (and therefore the network capacity) of n/N_s-central networks may be exponentially large in the number of nodes.

We also describe a constructive algorithm, which allows us to obtain a centralized network that performs a prescribed inertial dynamics and the desired switching properties of the network.

Chapter 2

Networks with complex behaviour and their applications

2.1 Centralized networks for elementary human motions

Let us assume that the networks consist of n centers with the states q_i, and a number of satellites with states X_j, Y_j, Z_j, where $j = 1, ..., N >> n$. In the simplest case, when we approximate a single relatively simple motion, the time evolutions of the center states are governed by harmonic oscillator equation:

$$\frac{\mathrm{d}^2 q_i}{\mathrm{d}t^2} + \omega_i^2 q_i = 0, \quad i = 1, \ldots, n, \tag{2.1.1}$$

where q_i is the coordinate of i-th oscillator, ω_i is the frequency of that oscillator, and n is the number of oscillators. Often even two oscillators $n = 2$ provide a good accuracy but for more complicated motions one can take $n \in \{3, 4, 5\}$. Let $q(t) = (q_1, \ldots, q_n)$ be the vector of the oscillator states, depending on time t, and $x_k(t)$ are output coordinates (here $x_1(t) = X(t), x_2(T) = Y(t), x_3(t) = Z(t)$).

The centers are connected with N output coordinates x_k by a network:

$$x_k = \sum_{j=1}^{N_m} W_{kj} \Phi_j(q, b) \tag{2.1.2}$$

where x_k is the k-th coordinate on the body, $k = 1, \ldots, N$, the functions Φ_j form a basis in the space $L_2([-X_0, X_0]$, where X_0 is characteristic maximal amplitude of motion for j-th point, and b is a parameter and N_m the number of basic functions. The matrix entry W_{kj} describes the action of the node j on x_k. Note that (2.6.1) defines a straight forward network that maps the center states q_i into the output coordinate x_k by a N_m hidden neurons (satellites), and therefore, there are no interactions between satellites.

There are possible different choices of Φ_j. For example, we can consider the following cases.

A Harmonic basis. Here we assume that

$$\Phi_j(q,b) = \cos(\omega j q), \tag{2.1.3}$$

where $\omega = b$ is a frequency.

B System of radial basic functions.

For the case where a motion consists of many segnemts and we observe sharp transitions between those segments, we can use radial basis functions

$$\Phi_j = \phi(b|q - \bar{q}^{(j)}|), \quad j = 1, \ldots, N_m, \tag{2.1.4}$$

where ϕ is a fixed function, b is a sharpness parameter, the $\bar{q}^{(j)}$ is the vector of centers of radial basis functions with components $\bar{q}^{(j)} = (\bar{q}_1^{(j)}, \ldots, \bar{q}_n^{(j)})$ and $|z|$ denotes the Euclidian norm of the vector z: $|z| = \sqrt{\sum_{i=1}^n z_i^2}$. We assume that the radial basic function $\phi(|z|)$ is well localized at $z = 0$ and smooth. For example, we can take a gaussian

$$\phi(|z|) = \exp(-|z|^2/2).$$

C Polynomial basis.

Here we take

$$\Phi(j,q,b) = q^{j-1}, \quad j = 1, \ldots, N_m. \tag{2.1.5}$$

The basis **B** has an important advantage: the radial basic functions provide local approximations that is important to approximate complicated motions with sharp transitions.

To make switching in the network, we will also use the sigmoidal functions σ.They are increasing and smooth (at least twice differentiable) functions such that

$$\sigma(-\infty) = 0, \quad \sigma(+\infty) = 1, \quad \sigma'(z) > 0. \tag{2.1.6}$$

Typical examples can be given by

$$\sigma(h) = \frac{1}{1+\exp(-h)}, \quad \sigma(h) = \frac{1}{2}\left(\frac{h}{\sqrt{1+h^2}} + 1\right). \tag{2.1.7}$$

The structure of interactions between centers and coordinates x_i can be described by Fig 2.10.

2.1.1 Centralized networks generating a large class of human body motions

To approximate different motions by a single network, we should have a possibility to change the frequencies and coefficients W_{kj}.

The main idea is as follows. Each motion can approximated by a network described in the previous subsection, with adjusted frequencies ω_i and appropriated

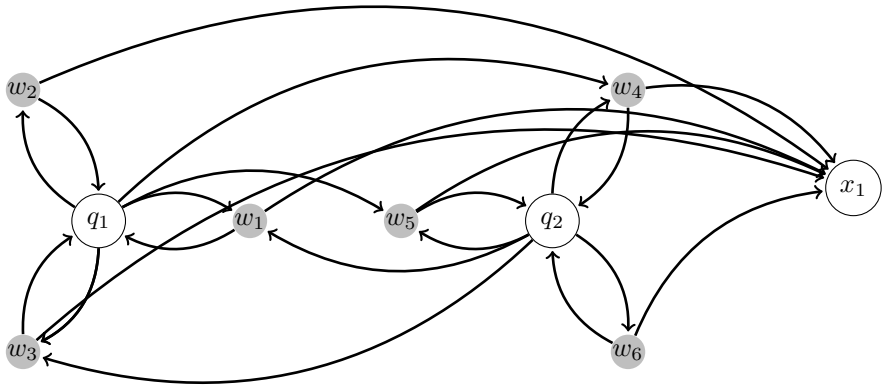

Figure 2.1: This image shows control of one of x-coordinates of human body motions by a networks consisting of two oscillators (v_1, v_2) and a RBF networks with $N = 6$ nodes. The graph consists of 8 nodes denoted by $v_1, v_2, w_1, w_2, w_3, w_4, w_5, w_6$. Each node w_i corresponds to a contribution of a radial basis function $\Phi(q - \bar{q}^{(j)})$. The nodes v_1, v_2 is the set of centers $\mathcal{C}$ and they affect w_i. In turn, the nodes w_i determine the output coordinate x_1.

coefficients W_{kj}. We can use nonlinear oscillators to obtain all possible frequencies. For example, one can use a model described below. Consider networks consisting of n centers, which evolve as nonlinear oscillators:

$$\frac{\mathrm{d}^2 q_i}{\mathrm{d}t^2} + z_c f(q_i) = 0, \quad i = 1, \dots, n, \tag{2.1.8}$$

where q_i is the coordinate of i-th oscillator, and $f(q)$ is a nonlinear function, z_c is a control paremeter. (One can take, for instance, $f = \sin(q)$ or $f = aq - bq^3$). We assume that

$$q_i(0) = 0, \quad p_i(0) = p_0, \quad p(t) = \frac{\mathrm{d}q}{\mathrm{d}t},$$

where p_0 is a fixed number. Solutions of (2.1.8) are periodic functions of time, with the period $T(z_c)$ and the frequency $\omega(z_c) = 2\pi/T$. It can be found by the motion integral of eq. (2.1.8)

$$E_i = \frac{1}{2}(\mathrm{d}q_i t)^2 + z_c F(q_i),$$

where F is the antiderivative of f: $f(q) = \mathrm{d}Fq$.

Consider a set of human motions characterized by a set of coordinates $x_1^{(j)}, \dots, x_N^{(j)}$, where the upper index j corresponds to a particular motion. Each motion can be described by the model (2.1.1) and (2.6.1) with corresponding frequencies $\omega_i^{(j)}$ and coefficients $W_{kl}^{(j)}$.

A switching between the different motions can be performed by a choice of control parameters $z_c = z_c^{(j)}$. We can take $z_c^{(j)} = z_0 + (j-1)\Delta z$, where $j = 1, \dots, M$.

By the switching module (which is described in the next subsection) we find a network subsystem, which has $z_c^{(j)}$ as local attractors. Then we can construct maps

$z_c \to \omega_1(z), \ldots, \omega_n(z)$ and $z_c \to W_{kl}(z_c)$ such that

$$\omega_l^{(j)} = \omega(z_c^{(j)}) \tag{2.1.9}$$

$$W_{li}^{(j)} = W_{ki}(z_c^{(j)}), \tag{2.1.10}$$

So, our global model for human locomotion consists of

1. a system of n nonlinear oscillators (2.1.8) with the control parameter z_c;
2. a RBF network defined by (2.6.1);
3. maps (2.1.9) and (2.1.10);
4. a switching module that is a network with $M+1$ nodes, where M the number of different motions.

In the next section we describe the switching module.

2.2 Switching module

Ideas behind construction. Before stating a formal statement, we present a brief outline, which describes main ideas of the proof and the architecture of the switchable network. The network consists of two modules. The first module is a generating one and it is a centralized neural network with n centers $q_1, \ldots, q_n$ and satellites $x_1, \ldots, x_N$. The second module consists of a center $v_{n+1} = z$ and m satellites $\tilde{w}_1, \ldots, \tilde{w}_m$. The satellites from this module interact only with the module center z, i.e., in this module the interactions can be described by a distar graph. Only the center of the second module interacts with the neurons of the first (generating) module. We refer to the second module as a switching one. This architecture is shown on Fig. 2.11.

For the switching module the corresponding differential equations have the following form. Let us consider a distar interaction motif, where a node z is connected in both directions with m nodes $\tilde{w}_1, \ldots, \tilde{w}_m$. By such notation the equations for the switching module can be written down in the form

$$\frac{\mathrm{d}\tilde{w}_i}{\mathrm{d}t} = \sigma\left(\tilde{b}_i z - \tilde{h}_i\right) - \kappa^{-1}\tilde{w}_i, \tag{2.2.1}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = \sigma\left(\kappa^{-1}\sum_{j=1}^{m}\tilde{a}_j\tilde{w}_j - h\right) - \xi\bar{\lambda}z, \tag{2.2.2}$$

where $i = 1, \ldots, m$ and $\tilde{b}_i, \tilde{a}_j, \bar{\lambda} > 0$.

To describe a mathematical idea how the switching module works, let us consider the system of the differential equations

$$\frac{\mathrm{d}v}{\mathrm{d}t} = Q(v, z), \quad v = (v_1, \ldots, v_{2n}) \tag{2.2.3}$$

where z is a real control parameter. Let $z_1, \dots, z_{m+1}$ be some values of this parameter. We find a vector field Q such that for $z = z_l$, where $l = 1, \dots, m$, the dynamics defined by (2.12.12) has the prescribed dynamics. For example, we can set $n = 2$ and

$$v_1 = q, \quad v_2 = p = \frac{\mathrm{d}q}{\mathrm{d}t},$$

and

$$Q = v_2, \quad \frac{\mathrm{d}v_2}{\mathrm{d}t} = z_c f(v_1)$$

that gives (2.1.8).

For the switching module we adjust the center-satellite interactions and the center response time parameter ξ in such a way that for a set of values ξ the switching module has the dynamics of system (2.12.8),(2.12.9) with m different rest points $z = z_1, z_2, \dots, z_{m+1}$ and for sufficiently large ξ system (2.12.8) and (2.12.9) has a single equilibrium close to $z_1 = 0$. Existence of such a choice will be shown in coming lemma 2.12.2.

Lemma 2.2.1 *Let m be a positive integer and $\beta \in (0, 1)$. For sufficiently small $\kappa > 0$ there exist $\bar{a}_j, b_i, \tilde{h}_i, h$ such that*

i *for an open interval of values ξ system (2.12.8),(2.12.9) has m stable hyperbolic rest points $z_j \in (j - 1 + \beta, j + \beta)$, where $j = 1, \dots, m$;*

ii *for $\xi > \xi_0 > 0$ system (2.12.8),(2.12.9) has a single stable hyperbolic rest point.*

Proof. Let $h = 0$. To find equilibria z, we set $\mathrm{d}\tilde{w}_i/\mathrm{d}t = 0$, and express $\tilde{w}_i$ via z. Then we obtain the following equation for the rest points z:

$$\xi z = \sigma\left(\sum_{j=1}^{m} \tilde{a}_j \sigma\left(\tilde{b}_j z - \tilde{h}_j\right)\right). \tag{2.2.4}$$

For especially adjusted parameters eq. (2.12.13) has at least m solutions, which give stable equilibria of system (2.12.8),(2.12.9). To show it, we assume that $0 < \kappa \ll 1$, $\tilde{b}_j = \tilde{b} = \kappa^{-1/2}$ and $\tilde{h}_j = \tilde{b}\mu_j$, where $\mu_j = j - 1 + \beta$. We obtain then

$$V(\xi z) = \sum_{j=1}^{m} \sigma(\tilde{b}(z - \mu_j)) + O(\kappa) = F_m(z, \beta, \kappa), \tag{2.2.5}$$

where $V(z)$ is a function inverse to $\sigma(z)$ defined on $(0, 1)$. Since $\tilde{b} \gg 1$ for small κ, the plot of the function F_m is close to a stairway (see Fig. 3). Let

$$\xi = 1, \quad \tilde{a}_1 = V(\mu_1) + \kappa, \quad \tilde{a}_j = V(\mu_j) - V(\mu_{j-1}), j = 2, \dots, m.$$

The intersections of the curve $V(z)$ with the almost horizontal pieces of the plot of F_m give us m stable equilibria of system (2.12.8),(2.12.9). These equilibria z_j lie in the corresponding intervals $(j-1+\beta, j+\beta)$. For sufficiently large ξ we have a single rest stable point z at 0. The lemma is proved.

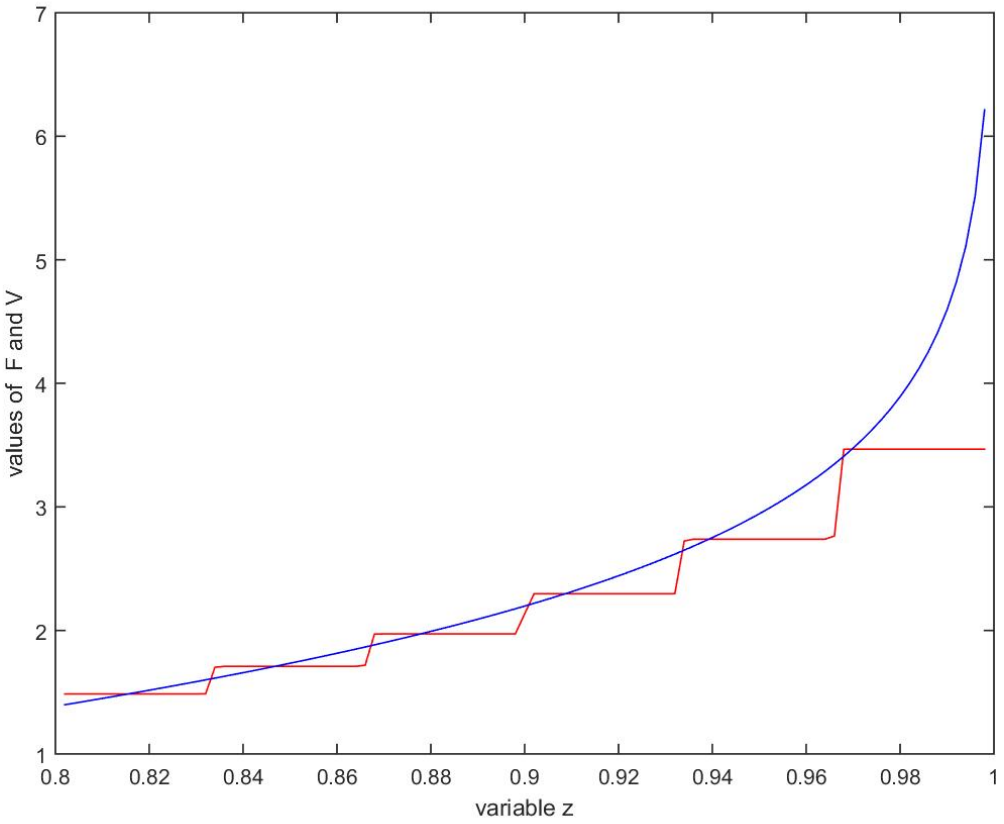

Figure 2.2: The intersections of the curve $F_m(z, \beta, \kappa)$ and the curve $V(z)$ give equilibria of system (2.12.8),(2.12.9) for $\xi = 1$. Stable equilibria correspond to the intersections of V with almost horizontal pieces of the graph of F_m.

2.3 Outline of RBF modules

Networks of radial basis functions (RBF). In mathematical modeling the RBF network is the artificial neural network using radial basis functions as activation functions. An output of network represents a combination of radial basis functions of inputs and neuron parameters. Application of such artificial neural network lies in the field of forecasting of time series and other similar tasks. For the first time, they were described in Brumkhed and Lowe's works in 1988.

The RBF networks, as a rule, has three layers: an input layer, the hidden layer with nonlinear function of activation RBF and a linear output layer. The input can be simulated by a vector of real numbers $x \in \mathbb{R}^n$. An output of network is a scalar function of an output vector $\varphi : \mathbb{R}^n \to \mathbb{R}$, and it is defined by the following relation:

$$\varphi(x) = \sum_{i=1}^{N} a_i \rho(||x - c_i||),$$

where N is the number of neurons in the hidden layer, c_i is a the central vector for a neuron i, a_i is neuron weight in i a linear output neuron. The vector c_i defines the location of the correspoding basis function. Functions, which depend only on the distance between the central vector and an entry vector x are called radial basic functions. In a RBF networks all inputs are connected to the hidden neuron.

Let us consider a concept of radial function more attentively. For definition of radial basic functions (RBF) we can use different relations, however, typically such a function must be well localized at a certain point (center). Throughout this thesis Gaussian functions will be used as RBF.

Remind that a normal distribution

$$\rho(x) = N(x, \mu, \sigma) \tag{2.3.1}$$

with a diagonal matrix of covariance and the mean $\mu = (\mu_1, \mu_2, ..., \mu_n)$ can be written down as follows:

$$\rho(x) = N \exp(-d(x, \mu)/2),$$

where $N = (2\pi)^{-n/2}(\sigma_1 ... \sigma_n)^{-1}$ — a normalizing multiplier and $d_j(x, x')$ — the weigthed Euclidean metrics in an n-dimensional Euclidian space of X:

$$d(x, x') = \sum_{d=1}^{n} \sigma_d^{-2} |x_d - x'_d|^2,$$

$$x = (x_1, ..., x_n), \quad x' = (x'_1, ..., x'_n).$$

As a distance $d(x, \mu)$ decreases, the value of density in a point x increases. Therefore the density $\rho(x)$ can be considered as a localized radial basis function.

RBF network were also applied to creation of the filter of Kallman [62]. The essence of work consists in use of the filter with some additions for increase in speed of preparation of information for processing of RBF by networks.

Also RBF networks are applied to recovery of images, which were defocused. The partial solution of this problem is available in work [61]. Kallman's filter, similar to work [62], however, with differences as receiving and data processing was here too used. In work [62] there was a training of the RBF neural network to use of the filter and assessment of parameter of a defocusing. Then, depending on value of this parameter, Kallman's filter with an illustration of his work and carrying out comparison with other methods was used. Networks of the radial basis functions (RBF) are applied not only to filtering or approximation of different functions of several variables, but also for forecasting problems. Training of RBF networks happens on algorithms, similar sigmoidal networks like a multilayer perseptron,

however the difference consists in data processing methods in the field of local displays. em eta poslednay fraza voobshe not clear

Thus, an acceleration of neural networks training and simplification of all structure of network in general is reached.

The important parameter of RBF networks is influence size. This parameter defines the steepness of function of activation of RBF of networks. The small size of influence will lead to a type of function with a small error of approximation and sharp changes of values that will cause problems in generalization and approximation of observation of a control set. Training of RBF networks has to happen in two stages, such as settings of the BF centers and training of neurons in the hidden layers that increases training speed.

Besides that RBF networks apply to approximation of the functions arising in various practical areas. For example, for forecasting of sales of air tickets the method of application of RBF networks shows satisfactory results [64].

In problems of filtration of signals and images the existing approaches use neural networks, in particular, networks of radial basis functions (RBF networks) for removal of noise. Practical applications of these methods find the application in problems of improvement of quality of processing of the speech [69], and other areas.

2.4 Algorithm of construction of RBF network to generate human body motions

2.4.1 Non-segmented motions

Simple motions can be handled as a whole, i.e., without any segmentation. Let us fix the index j, i.e., consider a particular motion. Let $t_1, \dots t_K$ be time moments where we have data on human body coordinates $X_j(t), Y_j(t), Z_j(t)$, where j is the index of an optical marker on the body and the number of the markers is N, $j = 1, ...N$. All X, Y and Z are thus vectors with N components. Let $\varepsilon(k, \omega)$ be the L_2-approximation accuracy for x-component and k-th marker defined by

$$\varepsilon_X^2(k, \omega) = \sum_{m=1}^{K} (X_k(t_m) - x_k(q(t_m), \omega))^2, \tag{2.4.1}$$

where $x_k(q)$ are defined by (2.6.1). Similarly,

$$\varepsilon_Y^2(k, \omega) = \sum_{m=1}^{K} (Y_k(t_m) - y_k(q(t_m), \omega))^2, \tag{2.4.2}$$

$$\varepsilon_Z^2(k, \omega) = \sum_{m=1}^{K} (Z_k(t_m) - z_k(q(t_m), \omega))^2, \tag{2.4.3}$$

The relative accuracies for X, Y, Z components are given by

$$\varepsilon^2_{r,X,\omega}(k) = \varepsilon^2_X(k,\omega) / \sum_{m=1}^{K} X_k(t_m)^2, \tag{2.4.4}$$

$$\varepsilon^2_{r,Y,\omega}(k) = \varepsilon^2_Y(k,\omega) / \sum_{m=1}^{K} Y_k(t_m)^2, \tag{2.4.5}$$

$$\varepsilon^2_{r,Z,\omega}(k) = \varepsilon^2_Z(k,\omega) / \sum_{m=1}^{K} Z_k(t_m)^2, \tag{2.4.6}$$

respectively. Let us fix a $k \in \{1, 2, ..., N\}$, i.e. a marker on the human body. For a set of frequency vector ω we compute the integral relative accuracy

$$\varepsilon_{r,k}(\omega) = \sqrt{(\varepsilon^2_{r,X,\omega}(k) + \varepsilon^2_{r,Y,\omega}(k) + \varepsilon^2_{r,Z,\omega}(k))/3}.$$

Then we find a ω^* such that $\varepsilon(\omega_*)$ be minimal:

$$\omega_* = \operatorname{argmin} \varepsilon_{r,k}(\omega).$$

The corresponding coefficients W_{kl} can be found by the standard Matlab programs, which approximate a target function by RBF networks. Here we use standard radial basis functions of Gaussian type, where the sharpness parameter b can be adjusted by trials and errors to minimize ε.

Numerical results show that the frequencies found for a particular motion by a value of k (a specific marker choice) and giving a small $\varepsilon_{r,k}$ can be applied to find good approximations for all rest values of k, i.e., for all other markers.

2.4.2 Segmented motions

For complex motions it is hard to approximate uniformly a whole motion by a few of neurons, sometimes such approximation is well anywhere except for a certain interval. In fact, it is difficult to expect that all parts of complicated motions consisting of quite different elementary submotions can be handled with the same frequencies. However, we can use the segmentation. We then decompose the motion in segments $[T_i, T_{i+1}]$, where $i = 1, ..., N_{seg}$. For each segment we can determine optimal frequencies as described above and compute the accuracies. The frequencies optimization can be done by two ways. If the number of oscillators is small (say, $n = 1, 2$), we can perform an exhaustive search over a uniform grid. For larger n one can use a random search.

2.4.3 Using RBF networks together with DMD

In this section we consider application of Dynamical Mode Decomposition (DMD) method of human body motion. We use the DMD method to compute frequensy oscillators.

Let us outline the DMD method briefly. The application of DMD to a model system depends on our choice of variables (observables). The best situation arises if te model system to be governed by a linear evolution law. Then the dynamics would be completely determined by the spectrum of the evolution operator. In this case one can use simple and effective algorithms designed for linear systems, for example controller design (1; 11) or stability analysis (84; 78). Mathematically, the evolution of observables of the system state is governed by the Koopman operator (64; 40), which is a linear but unfortunately infinite dimensional operator defined for an autonomous dynamical system. To overcome the difficulty conected with a large dimension of the Koopman operator, we make a dimension truancation. We introduce the slow subspace of the Koopman operator, which is the span of the eigenfunctions associated with eigenvalues near the unit circle in discrete time (or near the imaginary axis in continuous time) (44). The truncated slow dynamics captures the long term dynamics of observables and could serve as a low dimensional approximation of the otherwise infinite dimensional Koopman operator. It is possible if there exists a spectral gap, which separates the fast and slow modes. In addition to the eigenvalues and eigenfunctions, the important element of DMD analysis is the set of Koopman modes for the full state observable (15; 40). It allows us to reconstruct the state of the system as a linear combination of the Koopman eigenfunctions.

DMD gives us : (a) transform state space so that it the dynamics appear to be linear, (b) allows to determine the temporal dynamics of the linear system, and (c) allows to reconstruct the state of the original system from our model linear representation.

Note that beside DMD, there are several algorithms in the literature that can compute a), b) and c). For example, there are Generalized Laplace analysis (GLA) (GLA) (15; 54; 24), and the Ulam Galerkin Method (43; 79). These algotirhms do not require explicit governing equations, and thus they can be applied directly to data.

In this thesis, we prefer DMD because it is a powerful technique to isolate spatially coherent modes that oscillate at fixed frequencies (16; 68), thus it is relevant to describe data obtained by different markers on human body within human motions. The DMD is a data-driven and equation-free method. When the data is generated by a nonlinear dynamical system, then the DMD modes are closely related to eigenvectors of the infinite-dimensional Koopman operator (10; 16; 38). DMD has connections to the eigensystem realization algorithm (ERA) (9; 42; 103). Note that DMD was applied to describe fluid flows (69; 70), shock turbulent boundary layers (61), and foreground/ background separation in videos (39).

Our Implementation of DMD

We use the algoritihm described in the book (44). The corresponding Matlab code is as follows:

This algorithm gives us the approximative eigenvalues λ of the Koopman operator. The imaginary parts of those eigenvalues can be used as frequencies ω_i in our method. The simulations show that for simple motions it is sufficient to use 1, 2,

Algorithm 1 Our Implementation of DMD

Define $function[\Phi, \omega, \lambda, b, Xdmd, U, S, V, \tilde{A}] = DMDMine(X1, X2, r, dt)$;

Define $[U, S, V] = svd(X1,' econ')$;

Define $r = min(r, size(U, 2))$;

Define $U_r = U(:, 1 : r)$;

Define $S_r = S(1 : r, 1 : r)$;

Define $V_r = V(:, 1 : r)$;

$\tilde{A} = U_r' * X2 * V_r/S_r$;

$[W_r, D] = eig(\tilde{A})$;

$\Phi = X2 * V_r/S_r * W_r$;

$\lambda = diag(D)$;

$\omega = log(\lambda)/dt$;

$Compute\ DMD\ mode\ amplitudes$

$x1 = X1(:, 1)$;

$b = \Phi\ x1$;

$CalculateDMD\ reconstruction$

$mm1 = size(X1, 2)$;

$time_{dynamics} = zeros(r, mm1)$;

$t = (0 : mm1 - 1) * dt$;

for $1 \leq nm \leq iter$ **do**

    $time_{dynamics}(:, iter) = (b. * exp(\omega * t(iter)))$;

$Xdmd = \Phi * time_{dynamics}$;

$\tilde{A} = ctranspose(U) * X2 * V * inv(S)$;

or 3 frequencies. To obtain these frequencies by DMD, we should take a sufficiently large r, the number, which defines the size of matrix $\tilde{A}$ and the number of main modes in DMD approach.

To use this algorithm we must form the array X. To form that array, we choose markers numerated by $i_1, i_2, ..., i_k$, where the integer $k \geq r$, the integer number M_1, which defines the number of snapshots in time, T_0.The integer T_0 determines location of first snapshot. The code is as follows:

Algorithm 2

1: **for** $1 \leq m \leq 45 - 5$ **do**

2: $load([['X', num2str(m)],'.mat']);$

3: $load([['Y', num2str(m)],'.mat']);$

4: $load([['Z', num2str(m)],'.mat']);$

5: $Define function[X, Y, Xshift] = VakMorozDMD(Nmark, T0, M1, X1, X2, X3);$

6: **for** $1 \leq m \leq 3$ **do**

7: $q = load([['X', num2str(m)],'.mat']);$

8: $Y = eval(['X' num2str(m)]);$

9: **for** $1 \leq 1 \leq M1$ **do**

10: $X(j, 1) = X1(j + T0);$

11: $X(j, 2) = X2(j + T0);$

12: $X(j, 3) = X3(j + T0);$

13: $Xshift(j, 1) = X1(j + 1 + T0);$

14: $Xshift(j, 2) = X2(j + 1 + T0);$

15: $Xshift(j, 3) = X3(j + 1 + T0);$

16: $X = transpose(X);$

17: $Xshift = transpose(Xshift);$

In the previous part of the work, we used a combination of a linear model of autoregression and a network of radial basis functions (RBF). The obtained values after approximation are satisfactory in accuracy. In the calculations, an automatic and manual method of selecting the frequency for the oscillators was used. To test the theory of approximation by the DMD model, we performed several numerical experiments and the obtained values allow us to conclude that the proposed technique is effective.

To use centralized networks and RBF networks, we use first DMD to compute

Table 2.1: Usind DMD model and two oscillators

	epsX	epsY	epsZ	om1 (1)	om1(2)
X1...X8	0.0710	0.0692	0.0161	0.0026	0.0053
X8...X16	0.0630	0.0343	0.0174	0.0037	0.0011
X16...X24	0.0550	0.0171	0.0129	0.0046	0.0012
X24...X32	0.0237	0.0082	0.0179	0.0083	0.0016
X32...X40	0.0332	0.0205	0.0190	0.0050	0.0012
Y1...Y8	0.1384	0.2343	0.0268	0.0069	0.0036
Y8...Y16	0.0631	0.0392	0.0116	0.0060	0.0021
Y16...Y24	0.0261	0.1751	0.0090	0.0059	0.0022
Y24...Y32	0.2441	0.3479	0.0447	0.0100	0.0033
Y32...Y40	0.0292	0.1725	0.0087	0.0062	0.0024
Z1...Z8	0.1931	0.0949	0.0485	0.0044	0.0053
Z8...Z16	0.1093	0.1449	0.0304	0.0053	1.2312e-04
Z16...Z24	0.0806	0.1567	0.0126	0.0030	0.0053
Z24...Z32	0.1076	0.3083	0.0671	0.0154	0.0027
Z32...Z40	0.0445	0.1291	0.0309	0.0092	0.0028

frequencies. We used a set of programs that allows us to process values after using DMD, the RBF networks.

The first set of simulations we used the frequencies of the oscillators obtained on the basis of automatic selection, the parameter kdeg determines the number of oscillators, for the first case, kdeg = 2.

The table shows the values after using the DMD model, where:

om (n) the frequencies of oscillators, n = 1,...,kdeg;

eps - error;

kdeg - number of oscillators.

Each group of points according to the coordinates X, Y, Z was divided into parts consisting of 8 points. The value of the parameter "eps" after calculations show that the error is within normal limits.

The second set of simulations also used the frequencies the oscillators obtained on the basis of automatic selection. But in the second case, the number of oscillators was increased to three. This allowed us to analyze the effect of the number of oscillators on the value of "eps" - error.

In the table illustrating results of the second numerical experiment, the values of the "eps" parameter also are within the normal range; however, for some values of X, Y, Z, the error value is very small.

Table 2.2: Usind DMD model and three oscillators

	epsX	epsY	epsZ	om1 (1)	om1(2)	om(3)
X1...X8	0.0060	0.0044	0.0047	0.0150	0.0058	0.0020
X8...X16	0.0537	0.0103	0.0092	0.0066	0.0035	0.0013
X16...X24	0.0043	0.0046	0.0055	0.0162	0.0041	0.0014
X24...X32	0.0201	0.0318	0.0069	0.0135	0.0103	8.5591e-04
X32...X40	0.0282	0.0251	0.0061	0.0180	0.0026	0.0060
Y1...Y8	0.0630	0.0073	0.0089	0.0086	0.0052	3.7883e-04
Y8...Y16	0.0117	0.0041	0.0065	0.0135	0.0043	0.0018
Y16...Y24	0.0258	0.0088	0.0084	0.0078	0.0037	0.0025
Y24...Y32	0.0091	0.0079	0.0074	0.0194	0.0058	0.0013
Y32...Y40	0.0088	0.0125	0.0065	0.0121	0.0077	0.0029
Z1...Z8	0.0117	0.0041	0.0065	0.0135	0.0103	8.5591e-04
Z8...Z16	0.0258	0.0088	0.0084	0.0180	0.0026	0.0060
Z16...Z24	0.0080	0.0069	0.0057	0.0140	0.0048	3.9582e-04
Z24...Z32	0.1896	0.4844	0.0496	0.0210	0.0135	1.9248e-04
Z32...Z40	0.1213	0.1905	0.0116	0.0255	0.0091	8.1062e-04

2.5 Numerical Results

For empiric tests we use the CMU Motion Capture Database (20). We use the motions from family number 86, as these consist of sequences of several different motions performed by one actor subsequently, and hence have also been used as a test suite for different motion segmentation algorithms (see e.g.g (53) and references therein).

We use markers on left and right heels and left and right wrists, as in general from the position of these 4 markers even the full body motion can be reconstructed quite well (91).

We have considered 3 motions. The first motion consists of jumps, kicks and punches, the second one is walk, squats, run, stretch, jumps, punches, and drinking, and the third one is composed by walking, running, kicking, punching, knee kicking, and stretching.

The first motion is split in 4 segments [1, 1300], [1300, 2000], [2000, 3000], [3000, 4500], which were chosen visually by hand. Similarly, 2 and 3-th motions were decomposed into segments $[1, 1800], [1800, 2500], [2500, 4500]$. The second motion is particularly complicated and we used 3 oscillators and 100 neuron-satellites for each segments, for 1 and 3 -th is sufficient to use 2 oscillator and 100 neurons.

All networks could be computed in less than 500 sec of CPU time (on a Laptop Computer with 6 GB RAM and I5 Processor).

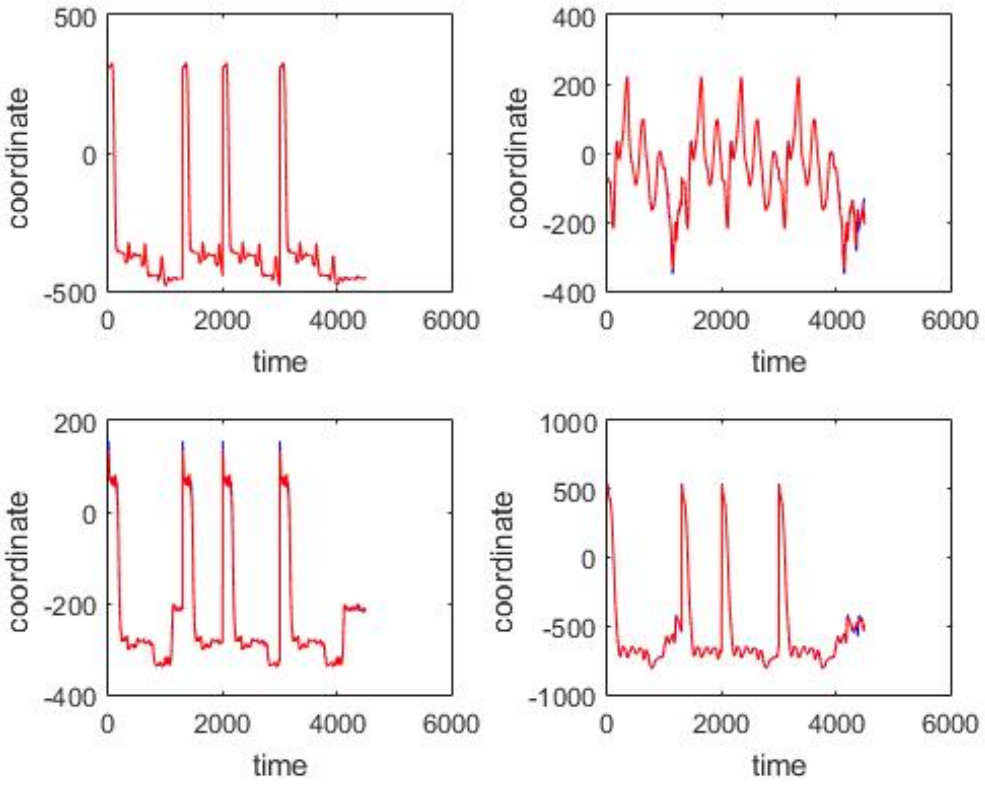

Figure 2.3: An approximation of vertical coordinates z by 100 neurons and two oscillators of motion CMU 86 trial consisting of jumps, kicks, and punches. The motion was segmented in 4 segments $[1, 1300, 2000, 3000, 4500]$. The relative integral accuracy over all x, y, z components is $0.0510, 0.0080, 0.0104, 0.0247$ for the corresponding segments. **a** Left, top. x-coordinate for the right heel. **b** Right, top. The left heel. **c** Left, bottom. x-coordinate for the right wrist, distal. **d** Right, bottom. The left wrist, distal. (the red curve shows the experimentally observed coordinates and the green curve gives their neural approximations).

2.6 Networks with complex large time behaviour

2.6.1 Problem statement and main assumptions

We consider the Hopfield-like networks defined by the following ordinary differential equations

$$\frac{du_i}{dt} = \sigma(\sum_{j=1}^{N} W_{ij}u_j - h_i) - \lambda_i u_i, \tag{2.6.1}$$

where u_i, h_i and $\lambda_i > 0$, $i = 1, ..., N$ are node activities, activation thresholds and degradation coefficients, respectively. The matrix entry W_{ij} describes the action

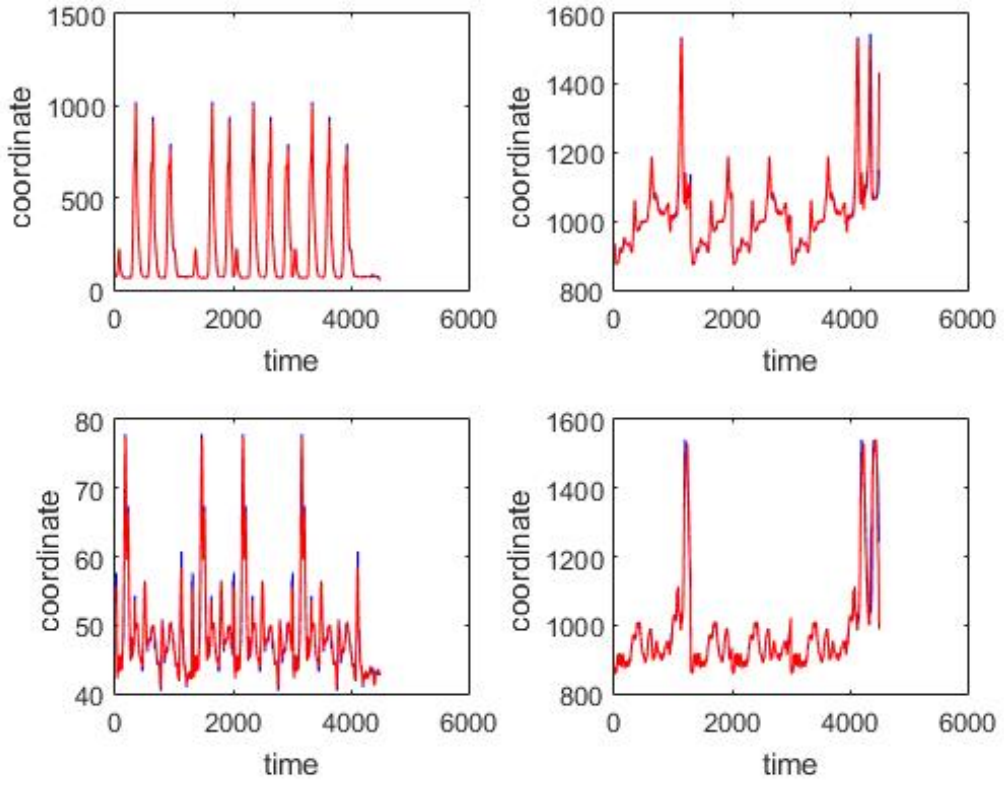

Figure 2.4: An approximation of vertical coordinates z by 100 neurons and two oscillators of motion CMU 86 trial consisting of jumps, kicks, and punches. **a** Left, top. x-coordinate for the right heel. **b** Right, top. The left heel. **c** Left, bottom. x-coordinate for the right wrist, distal. **d** Right, bottom. The left wrist, distal. (the red curve shows the experimentally observed coordinates and the green curve gives their neural approximations).

of the node j on the node i, which is an activation if $W_{ij} > 0$ or a repression if $W_{ij} < 0$. Contrary to the original Hopfield model, the interaction matrix W is not necessarily symmetric. The function σ is an increasing and smooth (at least twice differentiable) "sigmoidal" function such that

$$\sigma(-\infty) = 0, \quad \sigma(+\infty) = 1, \quad \sigma^{'}(z) > 0. \tag{2.6.2}$$

Typical examples can be given by

$$\sigma(h) = \frac{1}{1+\exp(-h)}, \quad \sigma(h) = \frac{1}{2}\left(\frac{h}{\sqrt{1+h^2}} + 1\right). \tag{2.6.3}$$

The structure of interactions in the model is defined by a weighted digraph (V, E, W) with the set V of nodes, the edge set E and weights W_{ij}. The nodes v_j, $j = 1..., N$ can be neurons or genes, depending on applications.

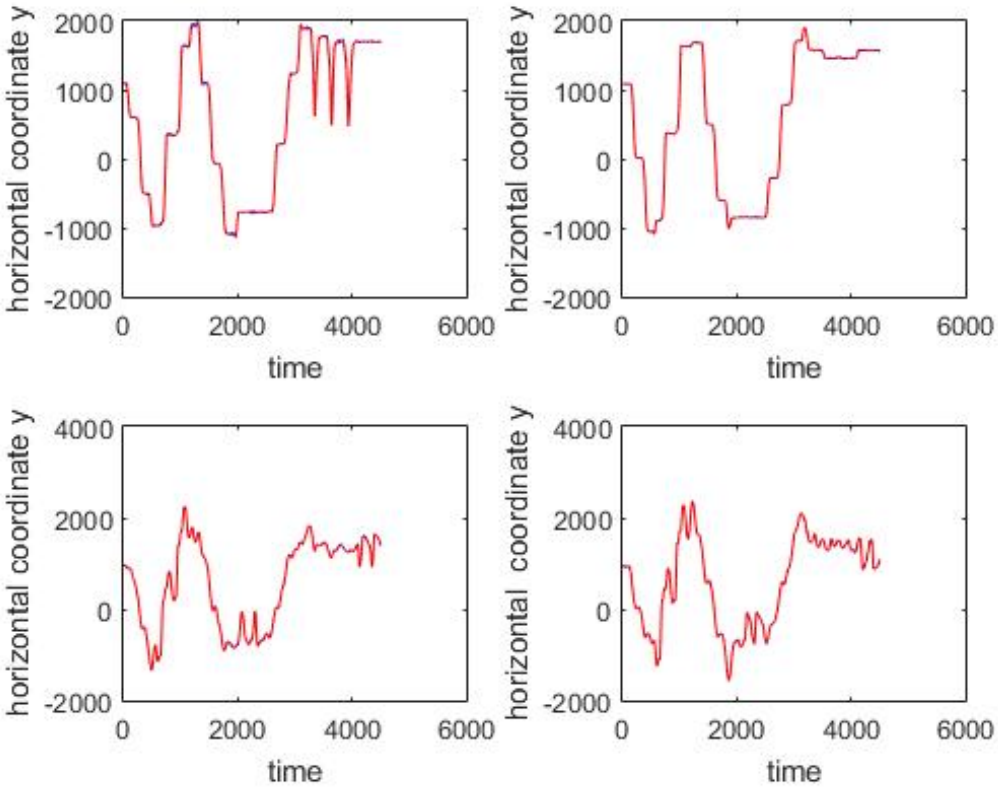

Figure 2.5: An approximation of y-horizontal coordinates for a simple non-segmented motion by 200 neurons and two oscillators. The oscillator frequencies are 0.0025 and 0.0006. **a** Left, top. y-coordinate for the right heel. **b** Right, top. The left heel. **c** Left, bottom. y-coordinate for the right wrist, distal. **d** Right, bottom. The left wrist, distal. (the red curves show the experimentally observed coordinates and the blue ones are their neural approximations).

Assumption 1.

Assume that if $W_{ji} \neq 0$, *then* (i, j) *is an edge of the graph,* $(i, j) \in E$. *This means that the* i*-th node can act on the* j*-th node only if it is prescribed by an edge of the digraph* (V, E, W). *We also suppose that* $(i, i) \notin E$, *i.e., the nodes do not act on themselves.*

Assume that the digraph (V, E, W) satisfies a condition, which is a variant of the centrality property. This condition is a purely topological one and thus it is independent on the weights W_{ij}. To formulate this condition, we introduce a special notation.

Let us consider a node v_j. Let us denote by $S^*(j)$ the set of all nodes, which act on the neuron j:

$$S^*(j) = \{v_i \in V : \quad edge\ (i, j) \in E\}. \tag{2.6.4}$$

For each set of nodes $\mathcal{C} \subset V$ we introduce the set $\mathcal{S}(\mathcal{C})$ of the nodes, which are under

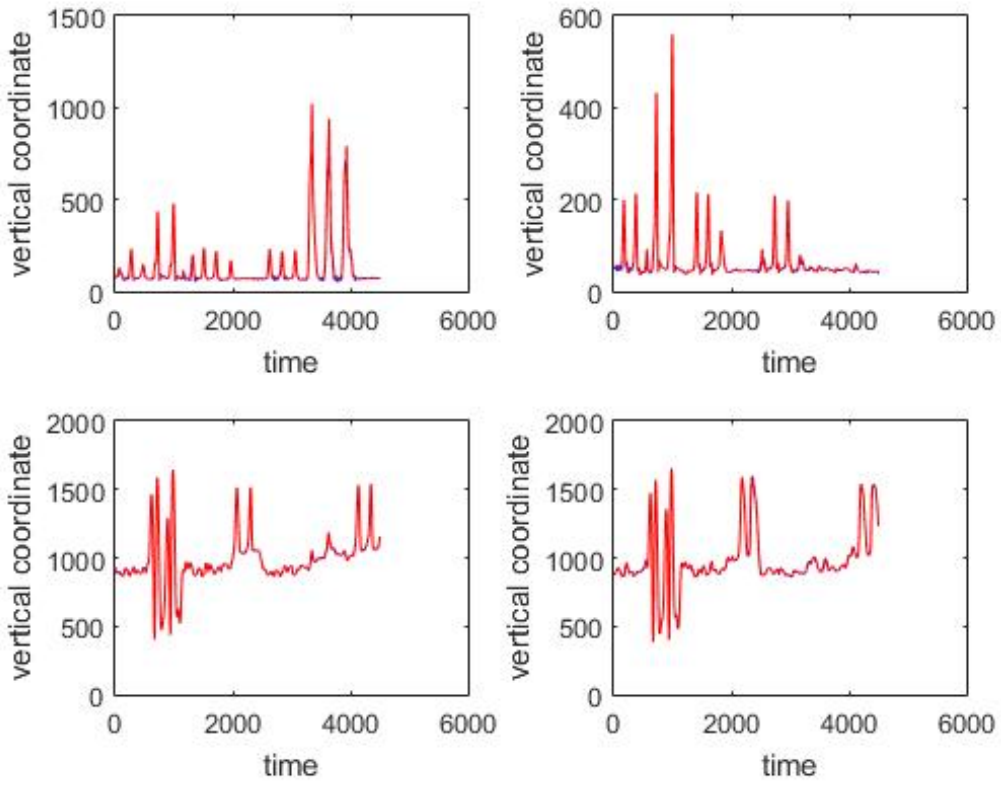

Figure 2.6: An approximation of z (vertical) coordinates by 200 neurons and two oscillators for a simple non-segmented motion. The oscillator frequencies are 0.0025 and 0.0006. **a** Left, top. z-coordinate for the right heel. **b** Right, top. The left heel. **c** Left, bottom. z-coordinate for the right wrist, distal. **d** Right, bottom. The left wrist, distal. (the red curves show the experimentally observed coordinates and the blue curves give their neural approximations).

action of all nodes from $\mathcal{C}$ and which are not belonging to $\mathcal{C}$:

$$\mathcal{S}(\mathcal{C}) = \{v_i \in V : \ for\ each\ j \in \mathcal{C}\ edge\ (j,i) \in E\ and\ v_i \notin \mathcal{C}\}. \tag{2.6.5}$$

n/N_s-Centrality assumption. *The graph (V, E, W) is connected and there exists a set of nodes $\mathcal{C}$ such that*

i *$\mathcal{C}$ consists of n nodes;*

ii *for each $j \in \mathcal{C}$ the intersection $\mathcal{S}^*(j) \cap \mathcal{S}(\mathcal{C})$ contains at least N_s nodes, where $N_s > c_0 N^\theta$ with constants $c_0 > 0, \theta \in (0,1)$, which are independent of j and N.*

The nodes from $\mathcal{C}$ can be interpreted as hubs (centers) and the nodes from $\mathcal{S}(\mathcal{C})$ are the satellites. The condition **ii** implies that each center is under action of sufficiently many satellites. In turn, if we consider the union of these satellites, all the centers act on them (see Fig.2.10). Such an intensive interaction leads, as we

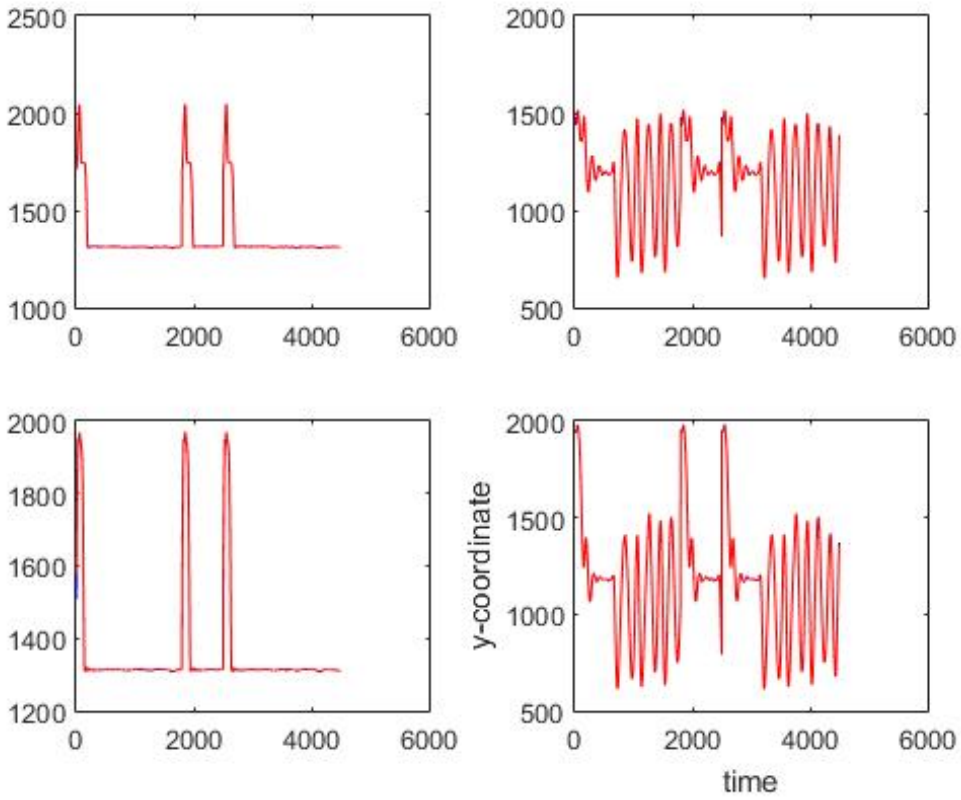

Figure 2.7: An approximation of y-horizontal coordinates of a complicated motion CMU 86 trial consisting of walk, squats, run, stretch, jumps, punches, and drinking. Before the approximation by a RBF-network with 3 centers and 100 satellites the motion was segmented into 3 intervals $[1, 1800]$, $[1800, 2500]$, and $[2500, 4000]$. The oscillator frequencies are: for 1-th interval $\omega = (0.0106, 0.0050, 0.0060)$, for the second one $\omega = (0.0188, 0.0287, 0.0058)$ and for 3-th one $\omega = (0.01160.00710.0046)$. The approximation accuracies for the segments are $0.0052, 0.0024, 0.0042$, respectively (for x, y and z coordinates together). The accuracy is so high that the blue curve (the target motion) and the red curve (the approximation) are almost non-distinguishable. **a** Left, top. x-coordinate for the right heel. **b** Right, top. The left heel. **c** Left, bottom. x-coordinate for the right wrist, distal. **d** Right, bottom. The left wrist, distal. (the red curve shows the experimentally observed coordinates and the green curve gives their neural approximations).

will see below, to a very complicated large time behaviour.

2.7 Outline of main results

Our results can be outlined as follows. The result on the inertial dynamics existence describes a situation, when the interaction topology is quite arbitrary. We assume

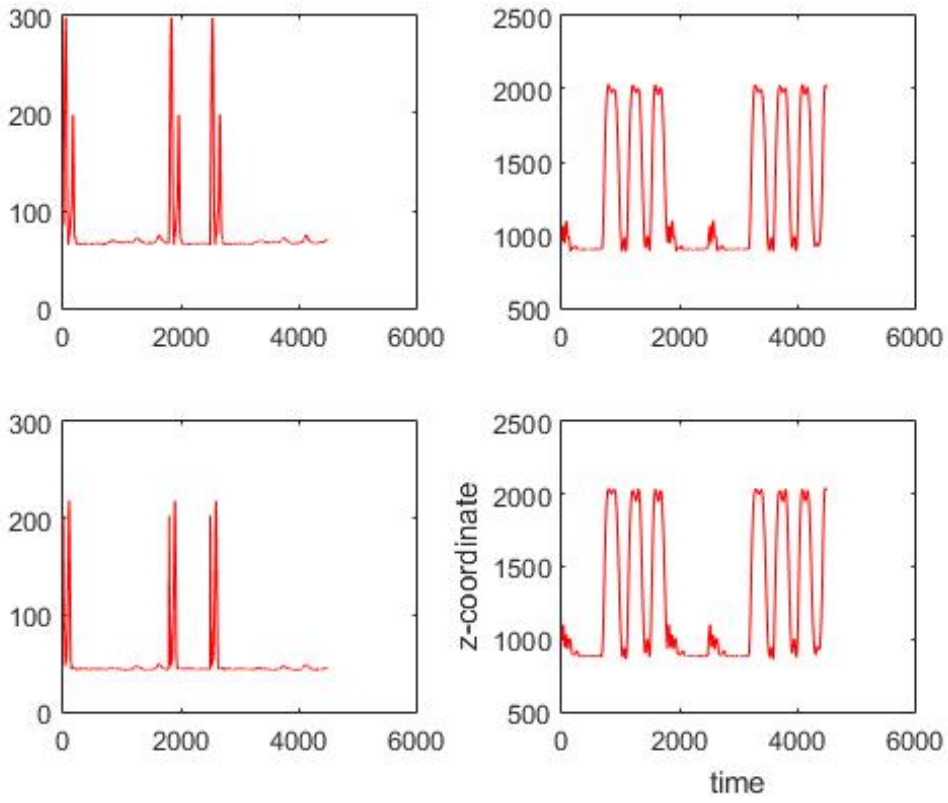

Figure 2.8: An approximation of vertical z coordinates of a complicated motion CMU 86 trial consisting of walk, squats, run, stretch, jumps, punches, and drinking. Before the approximation by a RBF-network with 3 centers and 100 satellites the motion was segmented into 3 intervals $[1, 1800]$, $[1800, 2500]$, and $[2500, 4000]$. The oscillator frequencies are: for 1-th interval $\omega = (0.0106, 0.0050, 0.0060)$, for the second one $\omega = (0.0188, 0.0287, 0.0058)$ and for 3-th one $\omega = (0.01160.00710.0046)$. The approximation accuracies for the segments are $0.0052, 0.0024, 0.0042$, respectively (for x, y and z coordinates together). The accuracy is so high that the blue curve (the target motion) and the red curve (the approximation) are almost non-distinguishable. **a** Left, top. x-coordinate for the right heel. **b** Right, top. The left heel. **c** Left, bottom. x-coordinate for the right wrist, distal. **d** Right, bottom. The left wrist, distal. (the red curve shows the experimentally observed coordinates and the green curve gives their neural approximations).

that there exist n slow nodes, say, $u_1, u_2, ..., u_n$ with $\lambda_i = O(1)$ whereas all the rest ones $u_{n+1}, ...u_N$ are fast, i.e., the corresponding λ_i have order $\mathcal{O}(\kappa^{-1})$, where κ is a small parameter. Then we show that there exists an inertial manifold of dimension n. We obtain, under general conditions, that for times $t >> \kappa \log \kappa$ the dynamics

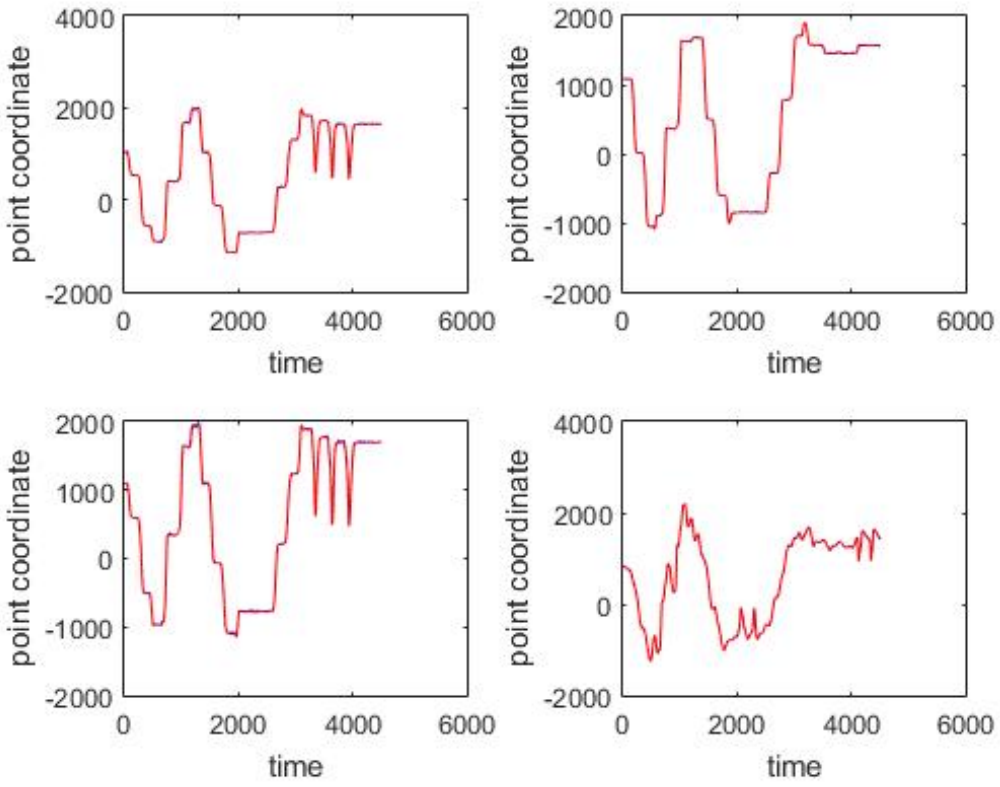

Figure 2.9: An approximation of y-horizontal coordinates by 200 neurons and two oscillators for non-segmented motion. The oscillator frequencies are 0.0025 and 0.0006. **a** Left, top. y-coordinate for the right ankle. **b** Right, top. The left ankle. **c** Left, bottom. x-coordinate for the right heel. **d** Right, bottom. The left wrist, distal. (the red curves show the experimentally observed coordinates and the blue curves represent their neural approximations).

of (2.6.1) is defined by the reduced equations

$$\frac{du_j}{dt} = F_j(u_1, ..., u_n, W, h, \lambda), \tag{2.7.1}$$

$$u_k = U_k(u_1, ..., u_n, W, h, \lambda), \quad k = n+1, ..., N, \tag{2.7.2}$$

where F_j and U_k are some smooth functions of $u_1, ..., u_n$, and h, λ denote the vector parameters $(h_1, ..., h_N)$ and $(\lambda_1, ..., \lambda_N)$, respectively. So, F gives us the inertial form on an inertial manifold. The inertial form completely defines the dynamics for large times (92).

More interestingly, we can show that the vector field F is, in a sense, maximally flexible. Roughly speaking, by the number of nodes N , the matrix W and h we can obtain all possible fields F (up to a small accuracy ϵ, which can be done arbitrarily small as N goes to ∞), see section 2.9 for a formal statement of this flexibility

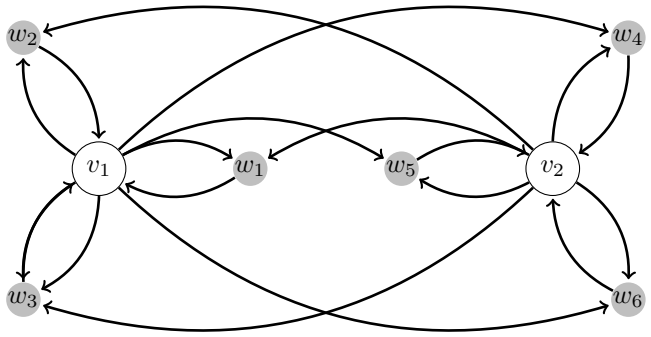

Figure 2.10: This image shows an n/N_s-central network with $n = 2$ and $N_s = 3$. The graph consists of 8 nodes denoted by $v_1, v_2, w_1, w_2, w_3, w_4, w_5, w_6$. The set $\{v_1, v_2\}$ is the set of centers $\mathcal{C}$. The sets $\mathcal{S}(\mathcal{C}), \mathcal{S}^*(v_1)$ and $\mathcal{S}^*(v_2)$ are as follows: $\mathcal{S}(\mathcal{C}) = \{w_1, w_2, w_3, w_4, w_5, w_6\}$, $\mathcal{S}^*(v_1) = \{w_1, w_2, w_3\}$ and $\mathcal{S}^*(v_2) = \{w_4, w_5, w_6\}$. The sets $\mathcal{S}^*(v_1) \cap \mathcal{S}(\mathcal{C}) = \{w_1, w_2, w_3\}$ and $\mathcal{S}^*(v_2) \cap \mathcal{S}(\mathcal{C}) = \{w_4, w_5, w_6\}$ contain three nodes each.

property. For the networks this flexibility property holds under n/N_s**-Centrality assumption**.

Let us introduce a special control parameter ξ, which modulates the degradation coefficient λ_i for a hub: $\lambda_i = \xi\bar{\lambda}_i$ for some $i \in \mathcal{C}$. This hub is a "controller". When we vary the coefficient ξ, the interaction topology and the entries of the interaction matrix do not change, but the response time of the controller hub changes.

One can choose the network parameters N, W, λ in such a way that for $\xi > \xi_0$ the global attractor is trivial, it is a rest point, but for an open set of other values ξ the global attractor of (2.6.1) contains a number of local attractors.

This result can be interpreted as "maximal switchability". A similar effect was found in (25) by numerical simulations for some models of neural networks. This effect describes a transition from neural resting states (NRS) to complicated global attractors, which occur as a reaction on learning tasks. Note that in (25) attractors consist of a number of steady states. In our case the global attractors can include many local attractors of all possible kinds including chaotic and periodic ones.

We end this section with a remark. Our method approximates vector fields by neural networks, but what can be said about the relationship between the trajectories of the simulated system and the ones corresponding to the neural network?

For chaotic and even for periodic attractors, direct comparison of trajectories is not a suitable test for the accuracy of the approximation. General mathematical arguments allow us say only that these trajectories will be close for bounded times. For large times we can say nothing especially for general chaotic attractors. Consider the case when the attractor $\mathcal{A}$ of the simulated system is transitive. This means the dynamics is ergodic and for smooth function ϕ the time averages

$$S_{F,\phi} = \lim_{T \to +\infty} T^{-1} \int_0^T \phi(v(t))dt \tag{2.7.3}$$

coincide with the averages $\int_{\mathcal{A}} \phi(v)d\mu(v)$ over the attractor, where μ is an invariant measure on $\mathcal{A}$.

Then, a suitable criterion of approximation is that the averages $S_{F,\phi}$ and the corresponding ones generated by the approximating centralized neural network, are close for smooth ϕ:

$$|S_{F,\phi} - S_{G_{anN}}, \phi| = Err_{approx} < \delta(\epsilon, \phi) \tag{2.7.4}$$

where G_{anN} is the neural network approximation of F and $\delta \to 0$ as $\epsilon \to 0$. This "stochastic stability" property holds for hyperbolic (structurally stable) attractors (52; 101; 97).

2.8 Conditions on network parameters and attractor existence

Our first results do not use any assumptions on the network topology. However, we suppose that there are two types of network components that are distinguished by their time scales into slow nodes and fast nodes. To take into account the two types of the nodes, we use distinct variables v_j for slow variables, $j = 1, \dots, n$ and w_i for the fast ones, $i = 1, \dots, N-n = N_1$. The real matrix entry A_{ji} defines the intensity of the action of the fast node i on the slow node j. Similarly, the $n \times N_1$ matrix $\mathbf{B}$, $N_1 \times N_1$ matrix $\mathbf{C}$ and $n \times n$ matrix $\mathbf{D}$ define the action of the slow nodes on the fast ones, the interactions between the fast nodes and the interactions between the slow nodes, respectively. We denote by h_i and λ_i the threshold and degradation parameters of the fast nodes and by $\tilde{h}_i$ and $\tilde{\lambda}_i$ the same parameters for the slow nodes, respectively. To simplify formulas, we use the notation

$$\sum_{j=1}^{n} D_{ij} v_j = \mathbf{D}_i v, \quad \sum_{k=1}^{N} C_{jk} w_k = \mathbf{C}_j w.$$

Then, equations (2.6.1) can be rewritten as follows:

$$\frac{dw_i}{dt} = \sigma\left(\mathbf{B}_i v + \mathbf{C}_i w - \tilde{h}_i\right) - \kappa^{-1}\tilde{\lambda}_i w_i, \tag{2.8.1}$$

$$\frac{dv_j}{dt} = \sigma\left(\mathbf{A}_j w + \mathbf{D}_j v - h_j\right) - \lambda_j v_j, \tag{2.8.2}$$

where $i = 1, ..., N_1,\ j = 1, ..., n$. Here unknown functions $w_i(t), v_j(t)$ are defined for times $t \geq 0$. We assume that κ is a positive parameter, therefore, the variables w_i are fast.

We set the initial conditions

$$w_i(0) = \tilde{\phi}_i \geq 0, \quad v_j(0) = \phi_j \geq 0. \tag{2.8.3}$$

It is natural to assume that all concentrations are non-negative at the initial moment. It is clear that they stay non-negative for all times.

2.8.1 Global attractor exists

Let us prove that the network dynamics is correctly defined for all t and solutions are non-negative and bounded. For positive vectors $r = (r_1, ..., r_n)$ and $R = (R_1, ..., R_{N_1})$, let us introduce the sets $\mathcal{B}$ defined by

$$\mathcal{B}(r, R) = \{(w, v) : 0 \le v_j \le r_j,\ 0 \le w_i \le R_j,\ j = 1, ..., n,\ i = 1, ..., N_1\}.$$

Note that

$$\frac{dw_i}{dt} < 1 - \kappa^{-1}\tilde{\lambda}_i w_i.$$

Thus, $w_i(t) < X(t)$ for positive times t, where

$$\frac{\mathrm{d}X}{\mathrm{d}t} = 1 - \kappa^{-1}\tilde{\lambda}_i X, \quad X(0) = w_i(0).$$

Therefore, resolving the last equation, and repeating the same estimates for $v_i(t)$, one finds

$$\begin{aligned} 0 \le w_i(x,t) &\le \tilde{\phi}_i \exp(-\tilde{\kappa}^{-1}\lambda_i t) + \kappa\tilde{\lambda}_i^{-1}(1 - \exp(-\kappa^{-1}\tilde{\lambda}_i t)), \\ 0 \le v_j(x,t) &\le \phi_j \exp(-\lambda_j t) + \lambda_j^{-1}(1 - \exp(-\lambda_j t)), \end{aligned} \tag{2.8.4}$$

Let us take arbitrary $a > 1$ and let $r_j(a) = a\lambda_j^{-1}$ and $R_i(a) = a\kappa\tilde{\lambda}_i^{-1}$. Estimates (2.8.4) show that solutions of (2.8.1), (2.8.2) exist for all times t and they enter the set $\mathcal{B}(r(a), R(a))$ at a time moment t_0. The solutions stay in this set for all $t > t_0$, thus, this set is absorbing. This shows that system (2.8.1),(2.8.2) defines a global dissipative semiflow S_H^t. Moreover, this semiflow has a global attractor contained in each $\mathcal{B}(r(a), R(a))$, where $a > 1$.

2.8.2 Assumptions for slow/fast networks.

A simpler asymptotic description of system dynamics is possible under assumptions on network components timescales. We suppose here that the u-variables are fast and the v-ones are slow. We show then that the fast w variables are slaved, for large times, by the slow v modes. More precisely, one has $w = \kappa U(v) + \tilde{w}$, where $\kappa U(v)$ is a correction and $\kappa > 0$ is a small parameter. This means that, for large times, the fast nodes dynamics is completely controlled by the slow nodes.

To realize this approach, let us assume that the system parameters

$$\mathbf{P} = \{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, h, \tilde{h}, \tilde{\lambda}, \lambda\}$$

satisfy the following conditions:

$$\mathbf{A} = \kappa^{-1}\bar{\mathbf{A}}, \tag{2.8.5}$$

$$|\bar{\mathbf{A}}|, |\mathbf{B}|, |\mathbf{C}|, |\mathbf{D}| < c_0, \tag{2.8.6}$$

$$0 < c_1 < \bar{\lambda}_i < c_2, \quad 0 < \tilde{\lambda}_i < c_3. \tag{2.8.7}$$

Here all positive constants c_k are independent of κ for small κ.

The scaling assumption on $\mathbf{A}$ is needed because, as we will prove later, $w = \mathcal{O}(\kappa)$ for small κ. For the same reasons, $\mathbf{C}_i w$ can be neglected with respect to $\mathbf{B}_i v$ for small κ, meaning that the action of centers on satellites is dominant with respect to satellites mutual interactions. In other words, these conditions describe a divide and rule control principle .

2.9 Realization of prescribed dynamics and maximally flexible systems

Our goal is to show that the network dynamics can realize, in a sense, arbitrary structurally stable dynamics of the centers. To precise this assertion, let us describe the method of realization of the vector fields for dissipative systems (proposed in (77)). More precisely, we are interested in systems enjoying the following properties:

A *These systems generate global semiflows $S^t_{\mathcal{P}}$ in an ambient Hilbert or Banach phase space H. These semiflows depend on some parameters $\mathcal{P}$ (which could be elements of another Banach space $\mathcal{B}$). They have global attractors and finite dimensional local attracting invariant C^1 - manifolds $\mathcal{M}$, at least for some $\mathcal{P}$.*

B *Dynamics of $S^t_{\mathcal{P}}$ reduced on these invariant manifolds can be, in a sense, almost completely tuned by variations of the parameter $\mathcal{P}$.*

It can be described as follows. Assume the differential equations

$$\frac{dq}{dt} = Q(q), \quad Q \in C^1(B^n) \tag{2.9.1}$$

define a global semiflow in a unit ball $B^n \subset \mathbb{R}^n$.

For any prescribed dynamics (2.9.1) and any $\epsilon > 0$, we can choose suitable parameters $\mathcal{P} = \mathcal{P}(n, F, \epsilon)$ such that

B1 *The semiflow $S^t_{\mathcal{P}}$ has a C^1- smooth locally attracting invariant manifold $\mathcal{M}_{\mathcal{P}}$ diffeomorphic to B^n;*

B2 *The reduced dynamics $S^t_{\mathcal{P}}|_{\mathcal{M}_{\mathcal{P}}}$ is defined by equations*

$$\frac{dq}{dt} = \tilde{Q}(q, \mathcal{P}), \quad \tilde{Q} \in C^1(B^n) \tag{2.9.2}$$

where the estimate

$$|Q - \tilde{Q}|_{C^1(B^n)} < \epsilon \tag{2.9.3}$$

holds. In other words, one can say that, by $\mathcal{P}$, the reduced dynamics on the invariant manifold can be specified to within an arbitrarily small error.

Therefore, roughly speaking all robust dynamics (stable under small perturbations) can be generated by the systems, which satisfy above formulated properties. Such systems can be named *maximally flexible*. In order to show that maximal flexibility covers also the case of chaotic dynamics, let us recall some facts about chaos and hyperbolic sets.

Let us consider dynamical systems (global semiflows) $S_1^t, ..., S_k^t$, $t > 0$, defined on the n-dimensional closed ball $B^n \subset \mathbb{R}^n$ defined by finite dimensional vector fields $F^{(k)} \in C^1(B^n)$ and having structurally stable attractors $\mathcal{A}_l$, $l = 1, ..., k$. These attractors can have a complex form, since it is well known that structurally stable dynamics may be "chaotic". There is a rather wide variation in different definitions of "chaos". In principle, one can use here any concept of chaos, provided that this is stable under small C^1 -perturbations. To fix ideas, we shall use here, following (81), such a definition. We say that a finite dimensional dynamics is chaotic if it generates a compact invariant hyperbolic set Γ, which is not a periodic cycle or a rest point (for a definition of hyperbolic sets see, for example, (81)). The hyperbolic sets give remarkable analytically tractable examples, where chaotic dynamics can be studied. For example, the Smale horseshoe is a hyperbolic set. If this set Γ is attracting we say that Γ is a chaotic (strange) attractor. In this thesis, we use only the following basic property of hyperbolic sets, so-called Persistence (81). This means that the hyperbolic sets are, in a sense, stable(robust). This property can be described as follows. Let a system of differential equations be defined by a C^1-smooth vector field Q on an open domain in $\mathbb{R}^n$ with a smooth boundary or on a smooth compact finite dimensional manifold. Assume this system defines a dynamics having a compact invariant hyperbolic set Γ. Let us consider ϵ-perturbed the vector field $Q + \epsilon\tilde{Q}$, where $\tilde{Q}$ is bounded in C^1-norm. Then, if $\epsilon > 0$ is sufficiently small, the perturbed field also generates dynamics with another compact invariant hyperbolic set $\tilde{\Gamma}$. The corresponding dynamics restricted to Γ and $\tilde{\Gamma}$ respectively, are topologically orbitally equivalent (topological equivalency of two semiflows means that there exists a homeomorphism, which maps the trajectories of the first semiflows on the trajectories of the second one, see (81) for details).

We recall that chaotic structurally stable (persistent) attractors and invariant sets exist: this fact is well known from the theory of hyperbolic dynamics (81).

Thus, any kind of the chaotic hyperbolic sets can occur in the dynamics of the systems, for example, the Smale horseshoes, Anosov flows, and the Ruelle-Takens-Newhouse chaos, see (81). Examples of systems satisfying these properties can be presented by some reaction-diffusion equations and systems (77; 93; 94), and neural network models (94).

2.10 Main results

For vectors $a = (a_1, ..., a_n)$ and $b = (b_1, ..., b_n)$ such that $a_i < b_i$ for each i let us denote by

$$\Pi(a, b) = \{v \in \mathbb{R}^n : a_i \leq v_i \leq b_i\} \tag{2.10.1}$$

a n-dimensional box in v-space. Moreover, let us define Π_λ by $\Pi_\lambda = \Pi(0, \lambda^{-1})$, where the vector λ^{-1} has components $(\lambda_1^{-1}, ..., \lambda_n^{-1})$.

Theorem 2.10.1 *Under assumptions (2.6.2), (2.8.5), (2.8.6) and (2.8.7) for sufficiently*

small κ there exists a n-dimensional inertial manifold $\mathcal{M}_n$ defined by

$$w_i = \kappa\tilde{\lambda}_i^{-1}U_i(v,\kappa,\mathbf{P}), \quad v \in \Pi_\lambda \tag{2.10.2}$$

where $U_i \in C^{1+r}(\Pi_\lambda)$, and $r \in (0,1)$. The functions U_i admit the estimate

$$|U_i(v,\kappa,\mathbf{P}) - \sigma\left(\mathbf{B}_i v - \tilde{h}_i\right)|_{C^1(\Pi_\lambda)} < c_4\kappa, \quad v \in \Pi_\lambda. \tag{2.10.3}$$

The v dynamics for large times takes the form

$$\frac{dv_j}{dt} = F_j(v,\mathbf{P}) + \tilde{F}_j(v,\kappa,\mathbf{P}), \tag{2.10.4}$$

where $\tilde{F}_j$ satisfy

$$|\tilde{F}_j|_{C^1(\Pi_\lambda)} < c_6\kappa \tag{2.10.5}$$

with

$$F_j(v,\mathbf{P}) = \sigma\left(\sum_{i=1}^{N-n}\bar{A}_{ji}\tilde{\lambda}_i^{-1}\sigma\left(\mathbf{B}_i v - \tilde{h}_i\right) + \mathbf{D}_j v - h_j\right) - \lambda_j v_j. \tag{2.10.6}$$

Note that the matrix $\mathbf{C}$ is not involved in relation (2.10.6), which defines the family of the vector fields F (inertial forms). This property holds due to the property that inter-satellite interactions are dominated by the satellite-center ones. The next assertion means that this principle allows us to create a network dynamics with prescribed dynamics (if the network satisfies n/N_s-centrality assumption and N is large enough). It is valid under the additional condition that the interaction graph (V,E) verifies the centrality condition.

Theorem 2.10.2 *Assume n/N_s-centrality assumption is satisfied. Then the family of the vector fields F defined by (2.10.6) is dense in the set of all C^1 vector fields Q defined on the unit ball $B^n \subset \mathbb{R}^n$. In the other words, centralized Hopfield neural networks are maximally flexible.*

Let us choose some i_C such that i_C belongs to $\mathcal{C}$. The corresponding node will be called a controller hub. We introduce the control parameter ξ by

$$\lambda_{i_C} = \xi\bar{\lambda}_{i_C}, \tag{2.10.7}$$

where we fix a positive $\bar{\lambda}_{i_C}$.

Theorem 2.10.2 can be used to show the following

Theorem 2.10.3 (Maximal switchability theorem)

Let us consider dynamical systems (global semiflows) $S_1^t, ..., S_k^t$, $t > 0$, defined on the n-dimensional closed ball $B^n \subset \mathbb{R}^n$ defined by finite dimensional vector fields $F^{(k)} \in C^1(B^n)$ and having structurally stable attractors $\mathcal{A}_l$, $l = 1, ..., k$. For sufficiently large N and any graph (V,E) satisfying the n/N_s- centrality condition

there exists a choice of interactions W_{ij} and thresholds h_i such that Assumption 1 holds and

*(**i**) there exist a ξ_0 such that for all $\xi > \xi_0$ the dynamics of network (2.6.1) has a rest point, which is a global attractor;*

*(**ii**) for an open interval of values ξ the global semiflow S_H^t defined by (2.6.1) have local attractors $\mathcal{B}_l$ such that the restrictions of the semiflow S_H^t to $\mathcal{B}_l$ are orbitally topological equivalent to the semiflows S_l^t restricted to $\mathcal{A}_l$.*

Finally, let us give an estimate on the maximal number of equilibria N_{eq} of centralized networks. This number is a characteristics of the network capacity, flexibility and adaptivity. To proceed to these estimates, let us define a procedure, which can be named decomposition into "distar" motifs. In the network interaction graph (E, V) we choose some nodes $v_1, ..., v_n$, which we conditionally consider as hubs. By "distar" motif we understand a part of interaction graph consisting of the hub v_j and the subset S_j of the set S_j^* (defined by (2.6.5)) consisting of the nodes connected in both directions to v_j: $S_j = \{v_i \in V : (i, j) \text{ and } (j, i) \in E\}$. This distar motif becomes an usual star if directions of the edges are ignored. Consider the union U_n of all S_j. Some nodes $w \in U_n$ may belong to two different sets S_j and S_k, where $k \neq j$. We remove from the vertex set V all such nodes. After such removing we obtain a part of graph $G_n = (V', E')$ of the initial graph (E, V), which is a union of n disjoint distars $S_1, ..., S_n$, where each S_k contains a single center $\{v_k\}$ and $\mu(S_k)$ satellites connected with the center in both directions. Recall that the graph (V', E') is a part of graph (V, E) if $V' \subset V$ and $E' \subset E$. These numbers $\mu(S_k)$ depend on the choice of hub nodes $\{v_1, ..., v_n\}$.

We will prove the following theorem:

Theorem 2.10.4 *The maximal possible number $N_{eq}(E, N)$ of equilibria of a network with a given interaction graph (E, V), where V consists of N nodes, satisfies*

$$N_{eq} \geq \sup \mu(S_1)\mu(S_2)...\mu(S_n), \tag{2.10.8}$$

where the supremum is taken over all integers $n > 0$ and all graphs G_n, which are parts of interaction graph (V, E) and consist of n disjoint distars. Here $\mu(S_l)$ is the number of the nodes in the distar S_l.

Consider now graphs, which are unions of identical distars. The degree of the center of each distar is $\lfloor (N - n)/n \rfloor$. Then, the maximal possible number N_{eq} of equilibria in such a centralized network (2.6.1) with N nodes and n centers satisfies $N_{eq} \geq \lfloor (N - n)/n \rfloor^n$, where $\lfloor x \rfloor$ denotes the floor of a real number x. Note that for a fixed N the maximum of $(N/n)^n$ over $n = 1, 2, ...$ is attained at $n = \lfloor N/5 \rfloor$, when the distars contain 5 satellites each. Therefore we obtain the estimate $N_{eq} \geq 4^{\lfloor N/5 \rfloor}$.

2.11 Proof of Theorem 2.10.1

Let us start by proving a lemma

Lemma 2.11.1 *Under assumptions (2.8.5), (2.8.6) and (2.8.7) for sufficiently small positive $\kappa < \kappa_0$ solutions (u, v) of (2.8.1), (2.8.2) and (2.8.3) satisfy*

$$w_i(t) = \kappa U_i(v(t), \mathbf{B}, \tilde{h}) + \tilde{w}_i(t), \tag{2.11.1}$$

where $U = (U_1, ..., U_n)$ is defined by

$$U_i(v, \mathbf{B}, \tilde{h}) = \tilde{\lambda}_i^{-1} \sigma \left(\mathbf{B}_i v(t) - \tilde{h}_i \right). \tag{2.11.2}$$

Then, for some T_0 function $\tilde{w}$ satisfies the estimates

$$|\tilde{w}(t)| < c_1 \kappa^2, \quad t > T_0 \tag{2.11.3}$$

where c_1 does not depend on t and κ. The time moment T_0 depends on initial data and the network parameters.

Proof. Let us introduce a new variables $\tilde{w}_i$ by (2.11.1). They satisfy the equations

$$\frac{d\tilde{w}_i}{dt} = H_i(v, \tilde{w}) - \kappa^{-1} \tilde{\lambda}_i \tilde{w}_i, \tag{2.11.4}$$

where

$$H_i(v, \tilde{w}) = \kappa Z_i(v) + W_i(v, \tilde{w}),$$

$$Z_i(v) = \sum_{j=1}^{n} \frac{\partial U_i(v)}{\partial v_j} (\sigma \left(\bar{\mathbf{A}}_j U + \mathbf{D}_j v - h_j \right) - \xi \bar{\lambda}_j v_j),$$

and

$$W_i(v, \tilde{w}) = \sigma \left(\mathbf{B}_i v + \mathbf{C}_i w - \tilde{h}_i \right) - \sigma \left(\mathbf{B}_i v - \tilde{h}_i \right).$$

Let us estimate $H_i(v, \tilde{w})$ for sufficiently large t. According to (2.8.4), for such times we can use that $(w, v) \in \mathcal{B}(r(a), R(a))$, where $a > 1$. In this domain $\mathcal{B}(r(a), R(a))$ one has $\sup |Z_i| < c_2$ and $\sup |W_i| < c_3 \kappa$, where c_2, c_3 are independent of κ. Therefore,

$$H_i(v(t), \tilde{w}(t)) < c_0 \kappa, \quad t > T_0(\kappa, \mathbf{P}).$$

Now, as above in subsection 2.8.1, equation (2.11.4) entails estimate (2.11.3). The assertion is proved.

Proof of Theorem 6.1. The rest part of the proof of Theorem 2.10.1 uses the well known technique of invariant manifold theory, see, for example, (81; 92; 33). Let us consider the domain $D_\kappa = \{w : |w| < c_1 \kappa^2\}$. Theorem 6.1.7 (33) shows that for $d \in (0, 1)$ there is a locally attractive C^{1+d}- smooth invariant manifold $\mathcal{M}_n$. Relation (2.10.3) follows from (2.11.3). The global attractivity of this manifold also follows from (2.11.3). The theorem is proved.

2.12 Demonstrations of Theorems

2.12.1 Proof of Theorem 2.10.2

The main idea of the subsequent statement is to study the dependence of the fields F_j defined by Eq.(2.10.6) on the parameters $\mathbf{P}$. To this end, we apply a special method stated in the next subsection.

Let us formulate a lemma, that gives us a key tool and which implies Theorem 2.10.2.

Lemma 2.12.1 *Assume*

$$a_i > \delta/\lambda_i, \quad b_i < (1-\delta)/\lambda_i \quad i = 1, ..., n. \tag{2.12.1}$$

Let $Q = (Q_1(v), ..., Q_n(v))$ *be a* C^1 *smooth vector field on* $\Pi(a, b)$ *and* $\delta > 0$ *verify*

$$-\delta < Q_i(v) < \delta, \quad v \in \Pi(a, b), \quad i = 1, ..., n. \tag{2.12.2}$$

Then there are parameters $\mathbf{P}$ *of the neural network such that the field* F *defined by (2.10.6) satisfies the estimates*

$$\sup_{v\in\Pi(a,b)} |F(v, \mathbf{P}) - Q(v)| < \epsilon, \tag{2.12.3}$$

$$\sup_{v\in\Pi(a,b)} |\nabla F(v, \mathbf{P}) - \nabla Q(v)| < \epsilon. \tag{2.12.4}$$

In other words, the fields F *are dense in the vector space of all* C^1 *smooth vector fields satisfying to (2.12.2).*

Proof. The proof uses the standard results of the multilayered network theory.

Step 1. The first preliminary step is as follows. Let us solve the system of equations

$$\sigma(R_j) = Q_j(v) + \lambda_j v_j, \quad v \in \Pi(a, b) \tag{2.12.5}$$

with unknown R_j. Here R_j are the regulatory inputs of the sigmoidal functions. These equations have a unique solution due to conditions (2.6.2), (2.12.1) and (2.12.2): the right hand sides $V_j + \lambda_j v_j$ range in $(0, 1)$. The solutions $R_i(v)$ are C^1-smooth vector fields.

Step 2. Consider relation (2.10.6). We choose entries A_{ji} and B_{il} in a special way. First, let us set $A_{ji} = 0$ if $i \notin \mathcal{S}^*(j)$, where the set $\mathcal{S}^*(j)$ is defined in the n/N_s-centrality assumption, see condition **ii**. Recall that $\mathcal{S}^*(j)$ is the set of the satellites acting on the center j. Note that then sum (2.10.6) can be rewritten as

$$F_j(v, \mathbf{P}) = \sigma\left(\sum_{i\in\mathcal{S}^*(j)} \bar{A}_{ji}\tilde{\lambda}_i^{-1}\sigma\left(\mathbf{B}_i v - \tilde{h}_i\right) + \mathbf{D}_j v - h_j\right) - \lambda_j v_j. \tag{2.12.6}$$

Using the result of step 1 and this relation, we see that our problem is reduced to the following: to approximate $R_j(v)$ in C^1 norm with a small accuracy $O(\epsilon)$ by

$$H_j(v, \mathbf{P}) = \sum_{i \in \mathcal{S}^*(j)} \bar{A}_{ji} \tilde{\lambda}_i^{-1} \sigma \left(\mathbf{B}_i v - \tilde{h}_i \right) + \mathbf{D}_j v - h_j. \tag{2.12.7}$$

Note that, according to the centrality assumption, the set $\mathcal{S}^*(j)$ contains $N_s > CN^\theta$ elements. Moreover, due to this assumption, the sum $\mathbf{B}_i = \sum_k B_{ik} v_k$ involves all k, $k = 1, ..., n$. Therefore, since n is fixed and N can be taken arbitrarily large, the theorem on the universal approximation by multilayered perceptrons (see, for example, (13)) implies that the fields $H = (H_1, ..., H_n)$ are dense in the Banach space of all the vector fields on $\Pi(a, b)$ (with C^1- norm). Therefore, H_j approximate R_j with $O(\epsilon)$-accuracy in C^1- norm. This finishes the proof.

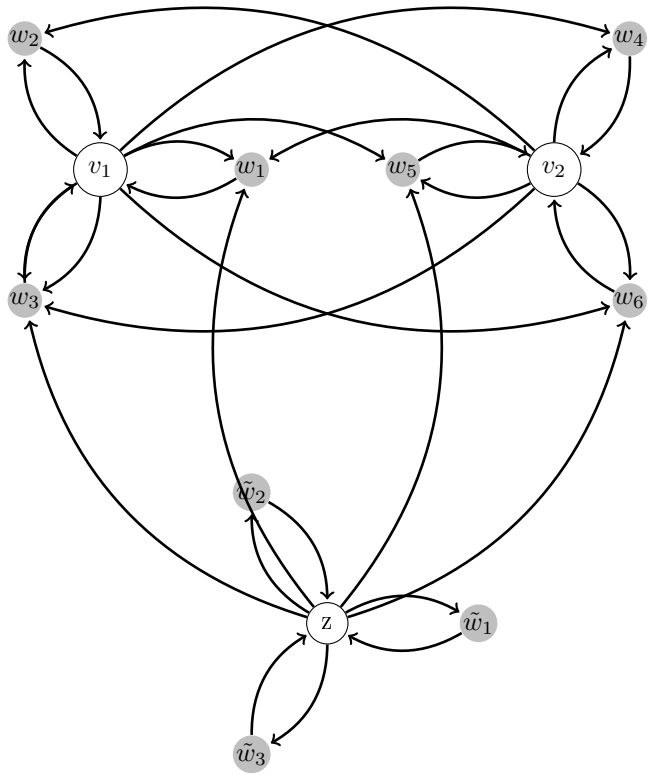

Figure 2.11: Modular architecture. The switching module consists of the center z and the satellites $\tilde{w}_1, \tilde{w}_2, \tilde{w}_3$. The generating module consists of the centers v_1, v_2 and the satellites $w_1, ..., w_6$.

2.12.2 Proof of Theorem 2.10.3

Ideas behind proof. Before stating a formal proof, we present a brief outline, which describes main ideas of the proof and the architecture of the switchable network. The network consists of two modules. The first module is a generating one and

it is a centralized neural network with n centers $v_1, ..., v_n$ and satellites $w_1, ..., w_N$. The second module consists of a center $v_{n+1} = z$ and m satellites $\tilde{w}_1, ..., \tilde{w}_m$. The satellites from this module interact only with the module center z, i.e., in this module the interactions can be described by a distar graph. Only the center of the second module interacts with the neurons of the first (generating) module. We refer to the second module as a switching one. This architecture is shown on Fig. 2.11.

For the switching module the correspoding equations have the following form. Let us consider a distar interaction motif, where a node z is connected in both directions with m nodes $\tilde{w}_1, ..., \tilde{w}_m$. We set $n = 1$ and $N_1 = m$, $\tilde{\lambda}_i = 1$, $\mathbf{D} = \mathbf{0}$, $\mathbf{C} = \mathbf{0}$, $\lambda_1 = 1$, and $A_{1j} = \kappa^{-1}\tilde{a}_j$ in eqs. (2.8.1) and (2.8.2). By such notation the equations for the switching module can be rewritten in the form

$$\frac{d\tilde{w}_i}{dt} = \sigma\left(\tilde{b}_i z - \tilde{h}_i\right) - \kappa^{-1}\tilde{w}_i, \tag{2.12.8}$$

$$\frac{dz}{dt} = \sigma\left(\kappa^{-1}\sum_{j=1}^{m}\tilde{a}_j\tilde{w}_j - h\right) - \xi\bar{\lambda}z, \tag{2.12.9}$$

where $i = 1, ..., m$ and $\tilde{b}_i, \tilde{a}_j, \bar{\lambda} > 0$.

Under above assumptions on the network interactions, equations for generating module can be represented as follows:

$$\frac{dw_i}{dt} = \sigma\left(\mathbf{B}_i v + \mathbf{C}_i w - d_i z - \bar{h}_i\right) - \kappa^{-1}\tilde{\lambda}_i w_i, \tag{2.12.10}$$

$$\frac{dv_j}{dt} = \sigma\left(\mathbf{A}_j w + \mathbf{D}_j v - \tilde{d}_j z - h_j\right) - \lambda_j v_j, \tag{2.12.11}$$

where $i = 1, ..., N$, $j = 1, ..., m$ and $d_i, \tilde{d}_j$ are coefficients.

These equations involve z as a parameter. This fact can be used in such a way. Consider the system of the differential equations

$$dv/dt = Q(v, z), \quad v = (v_1, \ldots, v_n) \tag{2.12.12}$$

where z is a real control parameter. Let $z_1, ..., z_{m+1}$ be some values of this parameter. We find a vector field Q such that for $z = z_l$, where $l = 1, ..., m$, the dynamics defined by (2.12.12) has the prescribed structurally stable invariant sets $\mathit{\Gamma}_l$.

Furthermore, according to theorem 2.10.2, for each positive ϵ we can choose the parameters $N, \mathbf{B}_i, \mathbf{C}_i, \tilde{b}_i, \tilde{a}_i, \bar{h}_i, \mathbf{A}_j, \mathbf{D}_j, d_i, \tilde{d}_j, h_j, \lambda_j, \tilde{\lambda}_i$ of the system (2.12.10) and (2.12.11) such that the dynamics of this system will have structurally stable invariant sets $\tilde{\mathit{\Gamma}}_l$ topologically equivalent to $\mathit{\Gamma}_l$.

For the switching module we adjust the center-satellite interactions and the center response time parameter ξ in such a way that for a set of values ξ the switching module has the dynamics of system (2.12.8),(2.12.9) with m different stable hyperbolic equilibria $z = z_1, z_2, ..., z_{m+1}$ and for sufficiently large ξ system (2.12.8) and (2.12.9) has a single equilibrium close to $z_1 = 0$. Existence of such

a choice will be shown in coming lemma 2.12.2. Then the both modules form a network having need dynamical properties formulated in the assertion of Theorem 2.10.3.

Proof. Let us formulate some auxiliary assertions. First we consider the switching module.

Lemma 2.12.2 *Let m be a positive integer and $\beta \in (0,1)$. For sufficiently small $\kappa > 0$ there exist $\bar{a}_j, b_i, \tilde{h}_i, h$ such that*

i *for an open interval of values ξ system (2.12.8),(2.12.9) has m stable hyperbolic rest points $z_j \in (j-1+\beta, j+\beta)$, where $j = 1, ..., m$;*

ii *for $\xi > \xi_0 > 0$ system (2.12.8),(2.12.9) has a single stable hyperbolic rest point.*

Proof. Let $h = 0$. To find equilibria z, we set $d\tilde{w}_i/dt = 0$, and express $\tilde{w}_i$ via z. Then we obtain the following equation for the rest points z:

$$\xi z = \sigma\left(\sum_{j=1}^{m} \tilde{a}_j \sigma\left(\tilde{b}_j z - \tilde{h}_j\right)\right). \tag{2.12.13}$$

For especially adjusted parameters eq. (2.12.13) has at least m solutions, which give stable equilibria of system (2.12.8),(2.12.9). To show it, we assume that $0 < \kappa << 1$, $\tilde{b}_j = \tilde{b} = \kappa^{-1/2}$ and $\tilde{h}_j = \tilde{b}\mu_j$, where $\mu_j = j - 1 + \beta$. We obtain then

$$V(\xi z) = \sum_{j=1}^{m} \sigma(\tilde{b}(z - \mu_j)) + O(\kappa) = F_m(z, \beta, \kappa), \tag{2.12.14}$$

where $V(z)$ is a function inverse to $\sigma(z)$ defined on $(0,1)$. Since $\tilde{b} >> 1$ for small κ, the plot of the function F_m is close to a stairway (see Fig. 3). Let

$$\xi = 1, \quad \tilde{a}_1 = V(\mu_1) + \kappa, \quad \tilde{a}_j = V(\mu_j) - V(\mu_{j-1}), j = 2, ..., m.$$

The intersections of the curve $V(z)$ with the almost horizontal pieces of the plot of F_m give us m stable equilibria of system (2.12.8),(2.12.9). These equilibria z_j lie in the corresponding intervals $(j-1+\beta, j+\beta)$. For sufficiently large ξ we have a single rest stable point z at 0. The lemma is proved.

Consider compact invariant hyperbolic sets $\Gamma_1, ..., \Gamma_m$ of semiflows defined by arbitrarily chosen C^1 smooth vector fields $Q^{(l)}$ on the unit ball $B^n \subset \mathbb{R}^n$, where $l = 1, ..., m$.

Lemma 2.12.3 *Let $\Pi(a,b)$ be a box in $\mathbb{R}^n$ and $m > 1$ be a positive integer. There is a C^1-smooth vector field Q on $\Pi(a,b) \times [0, m+1]$ such that equation (2.9.1) defines a semiflow having hyperbolic sets $\Gamma_1, ..., \Gamma_m$ and the restriction of this field on $\Pi(a,b) \times [0,1]$ has an attractor consisting of a single hyperbolic rest point.*

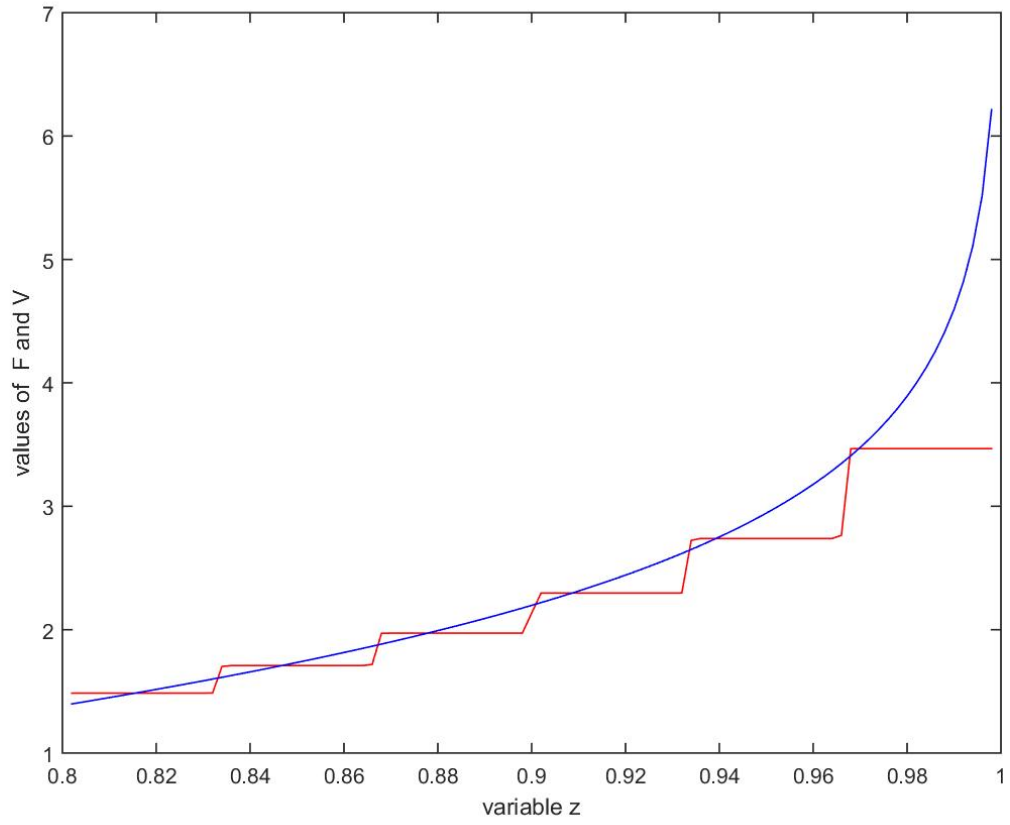

Figure 2.12: The intersections of the curve $F_m(z,\beta,\kappa)$ and the curve $V(z)$ give equilibria of system (2.12.8),(2.12.9) for $\xi = 1$. Stable equilibria correspond to the intersections of V with almost horizontal pieces of the graph of F_m.

Proof. The proof uses the following idea. For $k \in \{2,...,m+1\}$ let $Q^{(k)}(v)$ be a vector field on $\Pi(a,b)$ having Γ_{k-1} as an invariant compact hyperbolic set. Moreover, suppose that $Q^{(1)}$ has a single globally attracting rest point in $\Pi(a,b)$, $z_j \in (j-1+\beta, j+\beta)$, where $j = 1,...,m$ and $\beta \in (0,1)$. Let $\chi_k(z)$ be smooth functions of $z \in \mathbb{R}$ such that

$$\chi_k(z_l) = \delta_{lk},\ l \in \{1,...,m\}, \quad k = 1,...,m$$

where δ_{lk} stands for the Kronecker delta. Let $Q(v,z)$ be the vector field on $\Pi(a,b) \times [0, m+\beta]$ defined by

$$Q_i(v,z) = \sum_{k=1}^{m} Q_i^{(k)} \chi_k(z), \quad i \in \{1,...,n\}, \tag{2.12.15}$$

for first n components and $n+1$-th component of this field (denoted by z) is defined

by

$$Q_{n+1}(v,z) = F_m(z,\beta,\kappa), \tag{2.12.16}$$

where F_m is defined by (2.12.14). For $\beta \in (0,1)$ the function F_m has stable roots at the points $z = 1, 2, ..., m$. We observe that the equation for z-component $dz/dt = F_m(z,\beta,\kappa)$ does not involve v. By applying Lemma 2.12.2 we note that solutions $z(t, z(0))$ of the Cauchy problem for this differential equation verify $|z(t) - z_j| < \exp(-c_1 t)$, if $z(0)$ lies in an open neighbourhood of z_j. To conclude the proof, we consider the system

$$\mathrm{d}v_i/\mathrm{d}t = Q_i(v,z),\ \ i = 1, ..., n,$$

$$\mathrm{d}z/\mathrm{d}t = F_m(z,\beta,\kappa) - \xi\bar{\lambda}z = Q_{n+1}(z).$$

The right hand sides of this system define the field Q of dimension $n+1$ from the assertion of Lemma 2.12.3. To check this fact, we apply Lemmas 2.12.1 and 2.12.2 that completes the proof.

Next, to finish the proof of Theorem 2.10.3, let us take a box $\Pi(a,b)$, where $0 < a_i < b_i$. The semiflows defined by differential equations $dv/dt = \delta Q(v)$ are orbitally topologically equivalent for all $\delta > 0$. We approximate the first n components of the field Q by our neural network using Lemma 2.12.3 . We multiply here Q on an appropriate positive δ to have a field with components bounded by sufficiently small number in order to apply Lemma 2.12.1. Namely, we take δ such that $a_i > \delta/(\xi_0\bar{\lambda}_i)$ and $b_i < (1-\delta)/(\xi_1\bar{\lambda}_i)$ and apply Lemma 2.12.1. Note that this appoximation does not involve the control parameter ξ. Indeed, this parameter is involved only in the approximation of Q_{n+1}, which can be done independently, see the distar graph lemma 2.12.2. This concludes the proof of Theorem 2.10.3.

Remark. In Theorem 2.10.3, we assume that the vector field $Q(v)$ is given. However, by centralized networks we can solve the problem of identification of dynamical systems supposing that the trajectories $v(t)$ are given on a sufficiently large time interval whereas Q is unknown or we know this field only up to unknown parameters. An example, where we consider an identification construction for a modified noisy Lorenz system, can be found in section 2.13.

2.12.3 Proof of Theorem 2.10.4

Let us refer to the distar centers as hubs and to periphery nodes as satellites. We suppose that satellites do not interact each with others and a satellite interacts only with the corresponding hub. Therefore the interaction graph resulting from the "hub disconnecting" construction consists of n disconnected distar motifs.

Step 1. Let $n = 1$. We apply lemma 2.12.2 to the distar graphs, see the proof of the previous theorem. Then we have m_1 stable equilibria, where m_1 is the number of satellites in the distar motif.

Step 2. In the case $n > 1$ we consider the disconnected interaction graph consisting of n distar motifs, where the j-th distar motif contains m_j nodes. One has $m_1 + m_2 + ... + m_n = N - n$ and totally the graph consists of N nodes. For each

distar we adjust the parameters as above (see step 1). We obtain thus $m_1 m_2 ... m_n$ of equilibria and the theorem is proven.

2.13 Synthesis of switchable network with prescribed dynamics

The proof of Theorem 2.10.3 can be used to construct practically feasible algorithms, which solve the problem of construction of a switchable network with prescribed dynamical properties. As a matter of fact, we can address two different, but related problems. The first problem is the *synthesis* of a neural network with prescribed attractors and switchability properties. The second problem is the *identification* of a neural network from time series. First we state the solution of the first problem and after we describe how to resolve the second one by analogous methods.

The prescribed network properties for the *synthesis problem* are stated in Theorem 2.10.3. We describe here a step by step algorithm, allowing to construct a network with these properties.

Consider structurally stable dynamical systems defined by the equations

$$\mathrm{d}v/\mathrm{d}t = Q^{(l)}(v) \quad v = (v_1, ..., v_n) \in \mathit{\Pi}(a, b) \subset \mathbb{R}^n, \tag{2.13.1}$$

where $l = 1, ..., m$ and $\mathit{\Pi}(a, b)$ is a defined by (2.10.1). We suppose that the fields $Q^{(l)}(v)$ are sufficiently smooth, for example, $Q^{(l)} \in C^\infty(\mathit{\Pi}(a, b))$. Without any loss of generality we can assume that

$$1 < a_i < b_i, \tag{2.13.2}$$

(otherwise we can shift variables v_i setting $v_i = \tilde{v}_i - c_i$).

Step 1. Find a sufficiently small ϵ such that perturbations of vector fields $Q(v)^{(l)}$, which are ϵ small in C^1 norm, do not change topologies of semiflows defined by 2.13.1. Actually, it is hard to compute such a value of ϵ, so, in practice we simply choose a small ϵ by the trial and error method.

Step 2. We find a vector field $Q(v, z)$ with $n + 1$ components, where $z = v_{n+1} \in [a_{n+1}, b_{n+1}] \subset \mathbb{R}$ such that the first n components of $Q(v, z)$ are defined by relations (2.12.15) and the $n + 1$ component is defined by (2.12.16). Let $D = \mathit{\Pi}(a, b) \times [a_{n+1}, b_{n+1}]$.

To describe the next steps, first let us introduce the functions

$$G_j(\bar{v}, \mathbf{P}) = \sum_{i=1}^{N} \bar{A}_{ji} \sigma(\mathbf{B}_i \bar{v} - h_i), \tag{2.13.3}$$

where the parameter $\mathbf{P} = \{N, \bar{A}_{ji}, B_{ik}, h_j, j = 1, ..., n + 1, i, k = 1, ..., N\}$ and $\bar{v} = (v_1, ..., v_n, z)$.

Let us observe that dynamical systems $dq/dt = Q(q)$ and $dq/dt = \gamma Q(q)$ with $\gamma > 0$ have the same trajectories, invariant sets and attractors, therefore, instead of Q we can use γQ. We choose a $\gamma > 0$ and a small positive $\delta < 1$ such that

$$-\delta < \gamma Q_i(\bar{v}) < \delta, \quad \bar{v} \in D, \quad i = 1, ..., n + 1 \tag{2.13.4}$$

and

$$a_i > \delta/\lambda_i, \quad b_i < (1-\delta)/\lambda_i \quad i = 1, ..., n+1 \tag{2.13.5}$$

for $\lambda_i > 1$.

Then (2.13.4) and (2.13.5) imply that

$$0 < \gamma Q_j(\bar{v}) + \lambda_j \bar{v}_j < 1, \quad \bar{v} \in D, \ j = 1, ..., n+1. \tag{2.13.6}$$

Let σ^{-1} be the function inverse to σ. Due to (2.13.6) the functions

$$R_j(\bar{v}) = \sigma^{-1}(\gamma Q_j(\bar{v}) + \lambda_j \bar{v}_j) \tag{2.13.7}$$

are correctly defined and smooth on D.

Now we solve the following approximation problem.

To find the number N, the matrices $\bar{\mathbf{A}}, \mathbf{B}$ and vector h such that

$$|R_j(\bar{v}) - G_j(\bar{v}, \mathbf{P})| + |D_{\bar{v}}(R_j(\bar{v}) - G_j(\bar{v}, \mathbf{P}))| \leq \epsilon/2, \quad j = 1, ..., n+1. \tag{2.13.8}$$

This problem can be resolved by standard algorithms, which perform approximations of functions by multilayered perceptrons (13). Note that these standard methods are based on iteration procedures, which can use a large running time.

We describe here a new variant of the algorithm for this approximation problem, which uses a wavelet-like approach. This approach does not exploit any iteration procedures or linear system solving. All the procedure reduces to a computation of the Fourier and wavelet coefficients. However, this algorithm is numerically effective only for sufficiently smooth R_j with fast decreasing Fourier coefficients and for not too large dimensions n.

The solution of the approximation problem (2.13.8) proceeds in the two steps.

Step 3. We reduce the $n+1$-dimensional problem (2.13.8) to a set of one-dimensional ones as follows. Let us approximate the functions R_j by the Fourier expansion:

$$\sup_{\bar{v} \in D}(|R_j(\bar{v}) - \hat{R}_j(\bar{v})| + |\nabla_{\bar{v}}(R_j(\bar{v}) - \hat{R}_j(\bar{v}))|) < \epsilon/4, \tag{2.13.9}$$

where

$$\hat{R}_j(\bar{v}) = \sum_{k \in K_D} \hat{R}_j(k) \exp(i(k, \bar{v})), \tag{2.13.10}$$

$(k, \bar{v}) = k_1 \bar{v}_1 + k_2 v_2 + ... + k_{n+1} \bar{v}_{n+1}$ and the set K_D of vectors k is a finite subset of the $(n+1)$- dimensional lattice L_D

$$K_D \subset L_D = \{k = (k_1, ..., k_{n+1}) : k_i = (a_i - b_i)^{-1} \pi m_i \ for \ some \ m_i \in \mathbb{Z}\}. \tag{2.13.11}$$

The Fourier coefficients $\hat{R}_j(k)$ can be computed by

$$\hat{R}_j(k) = (volume(D))^{-1} \int_D R_j(\bar{v}) \exp(-i(k, \bar{v})) d\bar{v}.$$

In order to satisfy (2.13.9), we take a sequence of extending sets K_D. For some K_D relation (2.13.9) will be satisfied because the Fourier coefficients $\hat{R}_j(k)$ fastly decrease in $|k|$.

Step 4. We exploit the fact that the problem (2.13.8) is linear with respect to the coefficients $\bar{A}_{ij}$. For each $k \in K_D$ we resolve the following one-dimensional problem. Let

$$g(q, M, a, \beta, \bar{h}) = \sum_{i=1}^{M} a_i \sigma(\beta_i(q - \bar{h}_i)). \tag{2.13.12}$$

We are seeking for integer $M > 0$ and the vectors $a = (a_1, ..., a_M)$, $\beta = (\beta_1, ..., \beta_M)$ and $\bar{h} = (\bar{h}_1, ..., \bar{h}_M)$ such that

$$\sup_{q \in I_k} |W_{j,k}(q) - g(q, M, a, \beta, \bar{h})| < \epsilon(10|K_D|)^{-1}, \tag{2.13.13}$$

$$\sup_{q \in I_k} |dW_{j,k}(q)/dq - g'(q, M, a, \beta, \bar{h})| < \epsilon_1 \leq \epsilon(10|K_D|)^{-1}, \tag{2.13.14}$$

where $|K_D|$ is the number of the elements k in the set K_D,

$$W_{j,k}(q) = \hat{R}_j(k) \exp(iq),$$

$$g'(q, M, a, \beta, \bar{h}) = \sum_{i=1}^{M} a_i \sigma'(\beta_i(q - \bar{h}_i)), \tag{2.13.15}$$

and $q = (k, \bar{v}) \in I_k$, where I_k is the interval $[q_-(k), q_+(k)]$ with

$$q_-(k) = \min_{\bar{v} \in D}(k, \bar{v}), \quad q_+(k) = \min_{\bar{v} \in D}(k, \bar{v}).$$

These approximation problems are indexed by (j, k), where $j = 1, ..., n + 1$ and $k \in K_D$ (we temporarily omit dependence on (j, k) in $a, \beta, \bar{h}, M$ to simplify notation).

To resolve these one-dimensional approximation problems, we apply a method based on the wavelet theory. Notice that this method is numerically effective. First we observe that if (2.13.14) is fulfilled with a sufficiently small ϵ_1, then, to satisfy (2.13.13), it is sufficient to add a constant term of the form $a_{M+1}\sigma(b_{M+1}q)$ with $b_{M+1} = 0$ to the sum in the right hand side of (2.13.12).

Let us define the function ψ by

$$\psi(q) = \sigma'(q) - \sigma'(q - 1). \tag{2.13.16}$$

We observe that

$$\int_{-\infty}^{\infty} \psi(q)dq = 0 \tag{2.13.17}$$

and $\psi(q) \to 0$ as $|q| \to \infty$, therefore, ψ is a wavelet-like function.

Let us introduce the following family of functions indexed by the real parameters r, h:

$$\psi_{r,\xi}(q) = |r|^{-1/2}\psi(r^{-1}(q - \xi)). \tag{2.13.18}$$

For any $f \in L_2(\mathbb{R})$ we define the wavelet coefficients $T_f(r,\xi)$ of the function f by

$$T_f(r,\xi) = \langle f, \psi_{r,\xi}\rangle = \int_{-\infty}^{\infty} \mathrm{d}q f(q)\psi_{r,\xi}(q). \tag{2.13.19}$$

For any smooth function f with a finite support $I_R = (-R,R)$ one has the following fundamental relation:

$$f = c_\psi \int_0^\infty \int_{-\infty}^{\infty} r^{-2}\mathrm{d}r\mathrm{d}\xi T_f(r,\xi)\psi_{r,\xi} = f_{wav}. \tag{2.13.20}$$

for some constant c_ψ. This equality holds in a weak sense: the left hand side and the right hand side define the same linear functionals on $L_2(\mathbb{R})$, i.e., for each smooth, well localized g one has

$$\langle f, g\rangle = \langle f_{wav}, g\rangle.$$

Let $\delta(\epsilon) << \epsilon$ be a small positive number. According to (2.13.20) we can find positive integers p_1, p_2, points $r_1, ..., r_{p_1}$, $\xi_1, ..., \xi_{p_2}$ and a constant $\bar{c}_\psi$ such that the integral in the right hand side of (2.13.20) can be approximated by a finite sum:

$$\sup |f(q) - \bar{f}_{wav}(q)| < \delta, \tag{2.13.21}$$

where

$$\bar{f}_{wav} = \bar{c}_\psi \sum_{l_1=1}^{p_1}\sum_{l_2=1}^{p_2} r_{l_1}^{-2} T_f(r_{l_1},\xi_{l_2})\psi_{r_{l_1},\xi_{l_2}}.$$

In our case for each (j,k) we set $f = W_{j,k}(q)$ for $q \in I_k$ and $f = 0$ for $q \notin I_k$. We can take $r_{l_1} = r_+ l_1/p_1$, where r_+ is large enough, and $\xi_{l_2} = q_{\min} + (q_{\max} - q_{min})l_2/p_2$, where $q_{\min} < q_-(k)$, $q_{\max} > q_+(k)$ are sufficiently large and $l_1 = 1, ..., p_1, l_2 = 1, ..., p_2$. We can renumerate the points (r_{l_1}, ξ_{l_2}) by a single index $l = 1, ..., p$, where $p = p_1 p_2$, that gives us r_l, ξ_l and the wavelet coefficients $T_l = \bar{c}_\psi T_f(r_l, \xi_l)$.

Having p, r_l, ξ_l and the wavelet coefficients T_l, we obtain the following solution of the approximation problem (2.13.12):

$$M(j,k) = p, \quad \bar{h}_{2l-1}(j,k) = r_l^{-1}\xi_l, \quad \bar{h}_{2l}(j,k) = r_l^{-1}(\xi_l + 1),$$

$$\beta_{2l-1}(j,k) = \beta_{2l}(j,k) = r_l^{-1}, \quad a_{2l-1}(j,k) = -a_{2l}(j,k) = T_l,$$

where we have introduced the index (j,k) in notation for the solution $(M, a, \beta, \bar{h})$ to emphasize that problem (2.13.12) depends on this index.

Finally, in the end of this step we obtain the coefficients

$$M(j,k), a_1(j,k), ..., a_{M(j,k)}(j,k), \beta_1(j,k), ..., \beta_{M(j,k)}(j,k), \bar{h}_1(j,k), ..., \bar{h}_{M(j,k)}(j,k). \tag{2.13.22}$$

Step 5. We construct a network with $n+1$ centers $\bar{v}_1, ..., \bar{v}_{n+1}$ and N satellites as follows. Let $\mathbf{C} = 0$ and $\mathbf{D} = 0$, i.e., we assume that the satellites don't interact

among themselves and there are no direct interactions between the centers. The number of satellites is defined by

$$N = \sum_{j=1}^{n+1} \sum_{k \in K_D} M(j,k).$$

Each satellite can be equipped with a triple index (i,j,k), where $j = 1, ...n+1$, $k \in K_D$ and $i \in \{1, ...M(j,k)\}$. We set that all $h_j = 0$, $\tilde{\lambda}_i = 1$, and λ_j are chosen as above. The threshold $h_{i,j,k}$ for the satellite with the index (i,j,k) is defined by

$$h_{i,j,k} = \bar{h}_i(j,k)$$

where $\bar{h}_i(j,k)$ are obtained at the Step 4 (see (2.13.22)).

Furthermore, we define the matrices $\bar{\mathbf{A}}$ and $\mathbf{B}$ as follows. One has

$$B_{(i,j,k),l} = \beta_i(j,k)k_l,$$

(this relation describes an action of the l-th center on the satellite with index (i,j,k)) and

$$\bar{A}_{l,(i,j,k)} = a_l(j,k)$$

(this relation describes an action of the l-th center on the satellite with index (i,j,k)). Here $i \in \{1, ...M(j,k)\}$, $j, l = 1, ..., n+1$ and $k \in K_D$.

Remark. This algorithm can be simplified if instead networks (2.8.1), (2.8.2) we use analogous networks where satellites act on centers in a linear way:

$$\frac{dw_i}{dt} = \sigma\left(\mathbf{B}_i v + \mathbf{C}_i w - \tilde{h}_i\right) - \kappa^{-1}\tilde{\lambda}_i w_i, \tag{2.13.23}$$

$$\frac{dv_j}{dt} = \left(\mathbf{A}_j w - h_j\right) - \lambda_j v_j, \tag{2.13.24}$$

where $i = 1, ..., N_1$, $j = 1, ..., n$, and the fields $Q^{(l)}$ are defined by polynomials (note that Jackson's theorems (6) guarantee that any Q can be approximated by a polynomial field on $\Pi(a,b)$ in C^1-norm). Then we can simplify Step 3 and Step 4 of the algorithm as follows. We observe that we can set $\gamma = 1$ and in this case the functions R_j have the form

$$R_j(\bar{v}) = Q_j(\bar{v}) + \lambda_j \bar{v}_j. \tag{2.13.25}$$

On Step 3 for polynomial functions $R_j(v)$ we can also use simple algebraic transformations, instead of the Fourier decomposition, to reduce the multidimensional approximation problem to one dimensional ones. On step 4 the function ψ defined by (2.13.16) is well localized and therefore alternatively step 4 can be realized by standard programs using radial basic functions and the method of least squares (see an example on the Lorenz system below).

Let us turn now to the problem of *identification* of a neural network from time series produced by a dynamical system $dv/dt = Q(v, \mathbf{P})$, $v \in \mathbb{R}^n$ with unknown

parameters $\mathbf{P}$. Assume that we observe a time series $v(t_1), v(t_2), ..., v(t_K)$ and the time interval between observations is small: $t_{i+1} - t_i = \Delta t << 1$. We want to construct a network with n centers, which produces, in a sense, analogous time series. According to (2.7.4), a suitable criterion of trajectory similarity is as follows. We can approximate the averages $S_{Q,\phi}$ from (2.7.3) by the time series

$$S_{Q,\mathbf{P},\phi} \approx K^{-1}\Delta T \sum_{k=1}^{K} \phi(v(t_k)) = S^{(K)}_{Q,\mathbf{P},\phi}. \tag{2.13.26}$$

Then, if the network identification is correct, the averages defined by time series and the corresponding ones generated by the approximating centralized neural network, should be close for smooth weight functions ϕ:

$$|S^{(K)}_{Q,\mathbf{P},\phi} - S^{(K)}_{G_{anN},\phi}| = Err_{approx} < \delta(\phi) << 1, \tag{2.13.27}$$

where G_{anN} is the approximation of Q by the neural network.

As a first step, we can approximate the unknown field $Q(v)$ by finite differences, for example, using the relation

$$Q(\tilde{v}_i, \mathbf{P}) = (v(t_{i+1}) - v(t_i))\Delta t^{-1}, \quad \tilde{v}_i = (v(t_{i+1}) + v(t_i))/2. \tag{2.13.28}$$

For other values v the field Q can be reconstructed, for example, by a linear interpolation. The neural network approximation of Q can be obtained by applying the steps 2-5 of the synthesis algorithm described above.

We end this section with an illustration of the simplified variant of the identification and synthesis algorithm, see the preceding Remark.

As an example, we describe a solution of the following identification problem. Consider time series generated by the Lorenz system perturbed by noise. The Lorenz system involves a *controller* parameter. Adjusting the values of this parameter, we can obtain chaotic dynamics, time periodic one or dynamics with convergent trajectories. We are going to find a centralized network, which also has a controller parameter and can generate all this rich variety of trajectories. For chaotic and periodic trajectories this neural approximation should exhibit dynamics with analogous ergodic properties (in the sense of (2.13.27).

Recall that the Lorenz system has the form

$$\mathrm{d}x/\mathrm{d}t = \alpha(y - x), \quad \mathrm{d}y/\mathrm{d}t = x(\rho - z) - y, \quad \mathrm{d}z/\mathrm{d}t = xy - \beta z. \tag{2.13.29}$$

This system shows a chaotic behaviour for $\alpha = 10, \beta = 8/3$ and $\rho = 28$. For $\alpha = 10, \beta = 8/3$ and $\rho \in (0, 1)$ this system has a globally attracting rest point.

We introduce new variables $v_1 = x, v_2 = y, v_3 = z$ and $v_4 = \rho$ and consider a more complicated modified Lorenz system with a controller parameter: (compare with the proof of Theorem 2.10.3):

$$\mathrm{d}v_1/\mathrm{d}t = \alpha(v_2 - v_1) = f_1, \quad \mathrm{d}v_2/dt = r_1 v_1(v_4 - v_3) - r_2 v_2 = f_2, \tag{2.13.30}$$

$$\mathrm{d}v_3/\mathrm{d}t = r_3 v_1 v_2 - \beta z = f_3, \quad \mathrm{d}v_4/\mathrm{d}t = \sigma_H(v_4, b_0, h_0) - \xi v_4 = f_4, \tag{2.13.31}$$

where σ_H is a regularized step function defined by $H_1(w) = (1+\exp(-b_0(w-h_0))^{-1}$ with $b_0 >> 1$ and $h_0 = 1$. We set $\xi = 0.5$, $r_1 = 14, r_2 = 1, r_3 = 1$. The initial data for the fourth component $v_0 = v_4(0)$ is a controller parameter. For large b_0 the differential equation for v_4 has two stable equilibria: $v_4^- \approx 0$ and $v_4^+ \approx 2$. Therefore, for $v_0 \in (0,1)$ system (2.13.30), (2.13.31) has a globally attracting rest point and for $v_0 > 1$ the attractor of this system is chaotic Lorenz one. The parameters of this system are $\mathbf{P} = (\alpha, \beta, r_1, r_2, r_3)$.

Suppose we observe trajectories $v(t)$, $t \in [0,T]$ of system (2.13.30) at some time moments $t_0 = 0, t_1 = \mathrm{d}t, ..., t_p = p\Delta t$. In order to simulate experimental errors we have perturbed the system with additive noise. We are going to find a centralized network, which has an attractor with, in a sense, similar statistical characteristics. More precisely, we aim to minimize Err_{approx} from relation (2.13.27). For identification procedure we use a centralized network with 4 centers v_1, v_2, v_3 and v_4. In this case steps 3, 4 can be simplified if we use this specific form of the modified Lorenz system. The last center v_4 serves as a controller.

We state the algorithm for the modified Lorenz system, however, the method is general and feasible for identification by trajectories generated by all low-dimensional dynamical systems defined by polynomial vector fields.

First we set

$$\mathbf{C} = \mathbf{D} = 0. \tag{2.13.32}$$

This means that only satellites act on centers and vice versa. To find the matrices $\mathbf{A}$, $\mathbf{B}$ and the thresholds h_i, we solve the following approximation problems:

$$R(\mathbf{A}, \mathbf{B}, h) \to min, \quad R = \sum_{i=1}^{4} \sum_{j=1}^{p} (Q_i(t_j) - S_i(v(t_j, \mathbf{A}, \mathbf{B}, h))^2 \tag{2.13.33}$$

where

$$Q_i(t_j) = (v_i(t_j + \Delta t) - v_i(t_j))/\Delta t, \quad S_i(v, \mathbf{A}, \mathbf{B}, h) = \sum_{k=1}^{N_i} A_{ik} \sigma(\sum_{j=1} B_{kj} v_j - h_{ik}). \tag{2.13.34}$$

This approximation problem is nonlinear with respect to B and h. We can simplify this problem by the following heuristic method. Each function $f_i(\mathbf{v})$ defined on a open bounded domain can be represented as a linear combination of functions $g_l(\boldsymbol{v} \cdot \boldsymbol{k}_{li})$, where vectors $\mathbf{k}_{li}$ belong to a finite set of vectors K_i. For example, for system (2.13.30), (2.13.31) the components f_j for $j = 1, 2, 3$ can be represented as linear combinations of monomials:

$$f_j(v) = g_j(v) - \lambda_j v_j, \quad g_j(v) = \sum_{l=1}^{11} C(j,l) T_l(v) \tag{2.13.35}$$

where

$$T_l = v_l, \quad l = 1, 2, 3, 4$$

$$T_{2l+1} = (v_1 + v_l)^2, \quad T_{2l+2} = (v_1 - v_l)^2, \quad l = 2, 3, 4, \quad T_{11} = 1.$$

and $\lambda_1 = \alpha, \quad \lambda_2 = 1, \quad \lambda_3 = \beta$. Therefore, $K_1 = \{\mathbf{k}_{11} = (1, 0, 0, 0)\}$, $K_2 = \{\mathbf{k}_{12} = (1, 0, 1, 0), \mathbf{k}_{22} = (1, 0, -1, 0), \mathbf{k}_{32} = (1, 0, 0, 1), \mathbf{k}_{42} = (1, 0, 0, -1)\}$, $K_3 = \{\mathbf{k}_{13} = (1, 1, 0, 0), \mathbf{k}_{23} = (1, -1, 0, 0)\}$, $K_4 = \{\mathbf{k}_{14} = (1, 0, 0, 0)$. Let n_i be the number of the vectors contained in the set K_i, $n_1 = 1, n_2 = 4$, $n_3 = 2$ and $n_4 = 1$. In this case of the modified Lorenz system, the set K_D from (2.13.11) is the union of sets K_i, $i = 1, ..., 4$.

We take a sufficiently large N_L, a large b_0 and define the auxiliary thresholds $\bar{h}_{\mathbf{k}_{li},j}$, where $j = 1, ..., N_L$, by

$$\bar{h}_{\mathbf{k}_{li},j} = \min_{s=1,...,p, l \in K_i} v(t_s) \cdot \boldsymbol{k}_{li} + j(\max_{s=1,...,p, l \in K_i} v(t_s) \cdot \boldsymbol{k}_{li} - \min_{s=1,...,p, l \in K_i} v(t_s) \cdot \boldsymbol{k}_{li})/NL.$$

We seek coefficients $\bar{A}_{il,\mathbf{k}_{li}}$ and C_i, which minimize $R_i(\bar{\mathbf{A}}, C_i)$ for $i = 1, 2, 3, 4$:

$$R_i(\bar{\mathbf{A}}, C_i) \to min, \quad R_i = \sum_{j=1}^{p} (Q_i(t_J) - \tilde{S}_i(v(t_j), \bar{\mathbf{A}}, C_i))^2 \tag{2.13.36}$$

where

$$\tilde{S}_i(v, \bar{\mathbf{A}}, C) = C_i + \sum_{l=1}^{n_i} \sum_{j=1}^{N_L} \bar{A}_{ij,\mathbf{k}_{li}} \sigma(b_0(\mathbf{k}_{li} \cdot v - \bar{h}_{\mathbf{k}_{li},j})). \tag{2.13.37}$$

Note that since $\tilde{S}_i$ are linear functions of $\bar{A}_{il,\mathbf{k}_{li}}$ and C_i, problems (2.13.36) can be solved by the least square method. The important advantage of this approach is that approximations can be done independently for different components i.

This approximation produces a centralized network involving 4 centers and $N = 8N_L + 8$ satellites. Indeed, each vector $\mathbf{k}_{li}$ associated with a quadratic term T_l, gives us N_L sattellites to approximate this term. Moreover, we use 4 satellites for approximations of the linear terms and 4 satellites are necessary for constants C_i in the right hand sides of (2.13.37).

The numerical simulations give the following results. The trajectories to identify are produced by the Euler method applied to the system (2.13.30), (2.13.31) perturbed by noise, where the time step 0.005 on the interval $[0, 50]$, the noise is simulated by $\epsilon_N \omega(t_i)$, where $\omega(t)$ is the standard white noise and $\epsilon_N = 0.05$. As a result of minimization procedure, we have obtained the errors R_i of the order $0.01 - 0.1$. The trajectories of the system (2.13.30), (2.13.31) perturbed by noise and the corresponding neural networks are not close but they have a similar form and statistical characteristics that is confirmed by the value Err_{approx} (defined by (2.13.27)), which is 0.008, where the test function ϕ is $\phi(v) = v_1^2 + v_2^2/2 - 2v_3$. These results are illustrated by Fig. 4.

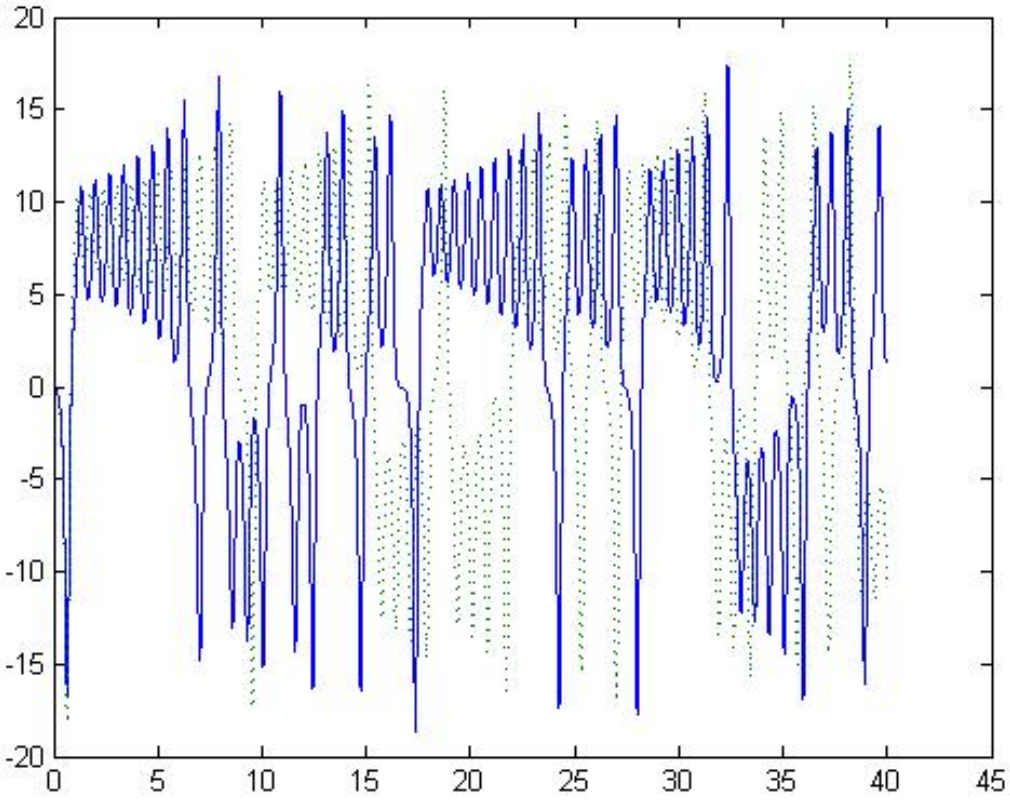

Figure 2.13: This plot shows trajectories of v_1-component of the Lorenz system perturbed by noise (the solid curve) and its neural approximation with $N = 20$ satellites (the dotted curve). The curves are not close but they exhibit almost identical statistical properties ($Err_{approx} = 0.008$ (the white noise level is 0.05, solutions have been obtained by the Euler method with the time step 0.001 on the interval $[0, 40]$).

Chapter 3

Filtration method by autoregression with auxiliary RBF network

3.1 Networks of radial basis functions and autoregression model

Autoregression model. Let us remind some elementary facts about autoregression models. They can be applied to describe stationary random processes. The difference from other models is that probabilistic properties of those model do not change in time – function of distribution of stationary dynamic series does not change as a result of time shift. If the calculated values of the model equations at the moment t are defined as a linear function of the previous values at $t-1, t-2, ...$, then we dealing with an autoregression model. We can classify autoregression models by the number p of the previous values involved in current computation, then we say that the model has the order p.

Models of the first and second order can be represented in the following form:

$$y_t = a_1 y_{t-1} + \varepsilon_t, \tag{3.1.1}$$

(a model of 1 order)

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \varepsilon_t, \tag{3.1.2}$$

(the model of second order) etc. Here ε_t is a noise.

To resolve the problem of model identification, we should, in particular, define its order. The model can be constructed when the values of the stationary process can be described by a number of changes in some trends. Consecutive capture of differences can serve as an example.

Note that the majority of the main autoregression processes can be described by fading autocorrelated functions. In that case, to find the order of model the log of the values of the autocorrelated function can be used.

In practice, identification of processes is a difficult task. Sometimes, it is not sufficient to use such simple models as autoregression ones.

3.1.1 A method of noise filtering by a combined model of autoregression and RBF network

In this subsection, the problem of noise filtering is considered for the time series analysis with use of networks of radial basis functions. Results can be applied for machine vision. Note that a much attention is paid to a problem of signal filtering by means of neural networks, however, in the majority of works either algorithms of linear approximation by means of autoregression, or neural approximations are used. Here we combine both approaches, for the first time a combined approach with simultaneous application of linear model of autoregression and RBF networks is considered.

Noise filtering by artificial neural networks is an important problem studied within the last decades [57,63,65,58,59,60,66]. We are going to apply the combined neural network consisting of linear autoregression and a RBF network for processing the movements of a biological object.

It is known that RBF network are characterized by a single hidden layer, nonlinear activation function for such layer and equality of the input and hidden layer on synoptic scales. The existing models of autoregression are characterized by linear relations connecting values of a time series in this moment from the previous values of the same row. We use both these approaches that allows us to use advantages of models of linear autoregression and nonlinearity of networks of radial and basic functions. For receiving basic data a few of experiments were done. In these experiments, time series giving coordinates of reference points of a biological object were obtained for different object movements. Natural places of a bend of a human hand were chosen as reference points (joints). Data were obtained by means of a video camera. For recognition of reference points it was decided to use different color markers which allowed to select them from a background. In language C++ two programming modules were written. The first module is used for definition of color responses of markers in the HCV color space on any static frame on which all markers are visible. The second module is used to selection of coordinates of an object on video.

In practical problems of processing of signals linear models and networks of nonlinear basis functions for removal noise signal components are used. However the estimates received in these works show that the linear model of data processing gives smaller value of an approximation accuracy, that makes filtration better.

The new method described in this thesis consists that at first filtration of a signal by means of linear model of autoregression is made, and then a residual signal (a difference initial and filtered) we approximate by means of networks of radial basis functions. It allows to reduce an error of filtration of linear autoregression several times that is confirmed experimentally.

3.1.2 Description of method

.

In case of two-dimensional movements the mathematical model has the form:

$$q_{il}(t+\Delta t) = \sum_{k=1}^{m} a_k^{(1)} q_1(t-(k-1)\Delta t) + \sum_{k=1}^{m} a_k^{(2)} q_2(t-(k-1)\Delta t) + b_i + R_i^{(N)}(t) + \varepsilon_i(t),$$

Where:

$q_{is}(t)$ – i- coordinate first reference point at the time of t. Here i=1 corresponds to a horizontal axis and i=2 – a vertical axis;

N – number of reference points;

$m \geqslant 1$ – the natural number, which defines the memory of autoregression model;

$\mathbb{R}_i^{(N)}(t)$ – network of radial basic functions with N neurons for coordinate i, autput signal of a network;

$\varepsilon_i(t)$ – an error in representation of y -coordinate;

$b_i, a_k^{(1)}, a_k^{(2)}$ - coefficients of linear autoregression model.

The main goal consists in selection of such parameters of network for radial basic functions and such coefficients $a_{ks}^{(1)}, a_{ks}^{(2)}$, that the average mean square value of the approximation error takes the minimal value.

The mathematical algorithm of approximate solution of this problem can be described as follows.

(1) At first, in the standard way we seek the solution of the optimization problem for linear model with $\mathbb{R}_i^{(N)} = 0$ (where the number N of neurons in RBF network equals 0). Suppose this model approximates observed output within an error $\varepsilon_i(t)$.

(2) Then errors $\varepsilon_i(t)$, $t = 1, ..., T$ is modelled by RBF network with N neurons and parameter of localization "spread", which can be found by trial and error search.

The program implementation of these steps (I) and (II) in the environment of Matlab is described below.

For store input data, the program is capable to receive coordinates of reference points defined by an motion of a human hand. The two-dimensional coordinate system is used. Implementation of three-dimensional option is possible by of the second video camera and adjustment of the calculations stated above (a mathematical model of two-dimensional movements). The shoulder joint, an elbow and a brush acted as reference points. For this purpose on a hand special color markers (pink, green, white), considerably different from a background were placed (brown), and the record of the videos lasting from 10 to 20 seconds with different models of the movement – trajectories, speed, disappearance of one of markers from a viewing field, etc. is made. The increase of of reference point numbers, even for all body, by introduction of different flowers as markers or by cutting of a frame on areas which can be crossed in small number of cases, for example, feet and knees of the legs of the same name, the head and a coxofemoral joint, etc. is possible. The video Record is made with a frequency of 24 frames per second that during the work with each frame separately allowed to receive enough data for outputs on operability of the offered model. The write enable made $320x240$ pixels that was sufficient and did not cause redundancy in volume of video files.

Further, the video is processed by means of algorithms of Open CV library. At the beginning, the image (a single frame of a video) is transformed to the HSV color space, by means of the cvCvtColor method (image, hsv) that allows us to process each marker on a hand, by certain coordinates of brightness in which the image can be displayed unambiguously. The example of a window for determination of color coordinates of a marker is presented in figures 11 a, b, c.

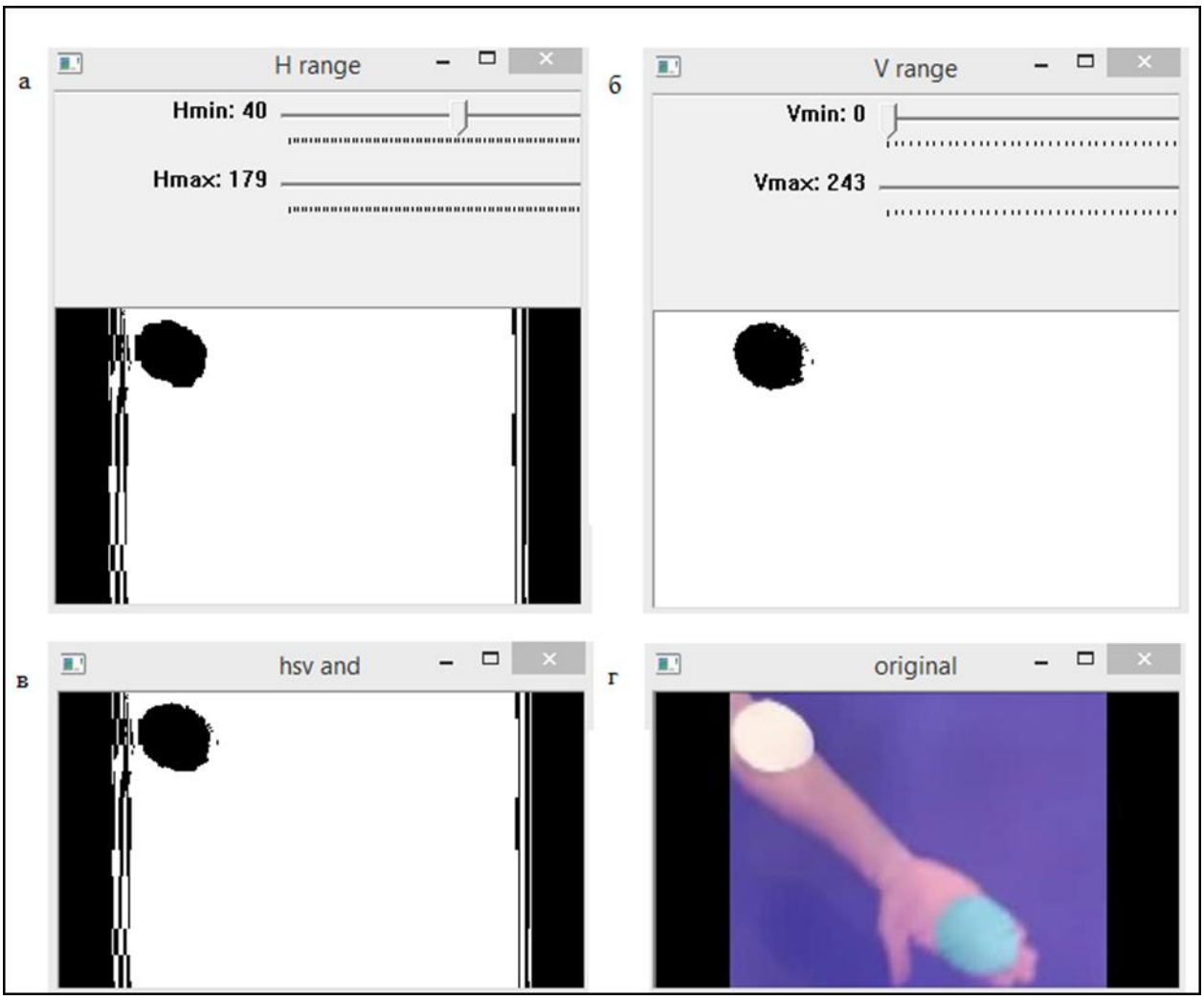

Figure 11. A window of determination of color coordinates of a marker for any frame.

a) selection of a marker in the range of H (hue);

b) selection of a marker in the range of V (color value);

c) the summing-up flow;

d) original frame;

The program representing the main module is written in programming language C++ . It can be described as follows. On a program input the video file which is divided into the sequence of frames by the cycle while proceeding before existence

of the following frame moves and using the cvQueryFrame method. Further, for each frame there is conversion to a color model of HSV, and to similarly auxiliary module, there is selection of area of the image in the borders which once are picked up in the auxiliary module written at a preparatory stage. Thus, for each frame three of its copies only with markers consistently turn out. For the received area of the image on each copy everything is excluded, except the pixels containing one of markers. Then, the frame with a tag will be transformed to the black-and-white image, and there is a calculation of the center of masses for the turned-out figure. The marker looks white on a black background. The algorithm of calculation of the center of masses is necessary for average value of all set of coordinates of a marker and definition of its central point and registers in a vector. The vector as data type, is selected as the program is capable to work with video files of bigger duration, than specified, and it means that the array size for record of coordinates is initially unknown. The values written in a vector were written in output files. For each marker two files with coordinates of X (t) and Y(t), t=1, ..., T were received. These files were processed further among Matlab.

Program implementation of steps (I) and (II) of mathematical algorithm can be described as follows. It is possible to apply the standard NEWLIND program to implement step (I). Here (P, Z) where the target vector of Z is a value $q_{il}(t+\Delta t), t = 1, ..., T-1$ and the explaining P vector – all values of coordinates j, s and t.

When linear neural network is constructed, the procedure of simulation $q_{il} = sim(net, P)$ was used to find an approximation error of the linear model $z_{il}(t)$. This vector is presented in the form of network of the radial basic function $R_i^{(N)}(t, spread)$. There is possible to find spread parameter of localization of basic functions by simple trial and error search for the purpose of minimization of an error of the combined model.

Results.

The efficiency of filtration is visible from the plots given below. These plots are obtained as follows. We start the program which builds plots of linear dependences of coordinates of X and Y from time.

We determine time interval equal to 0.3 sec. which was calculated from quantity of frames and duration of the video file. Schedules of dependences are presented in figure 2.

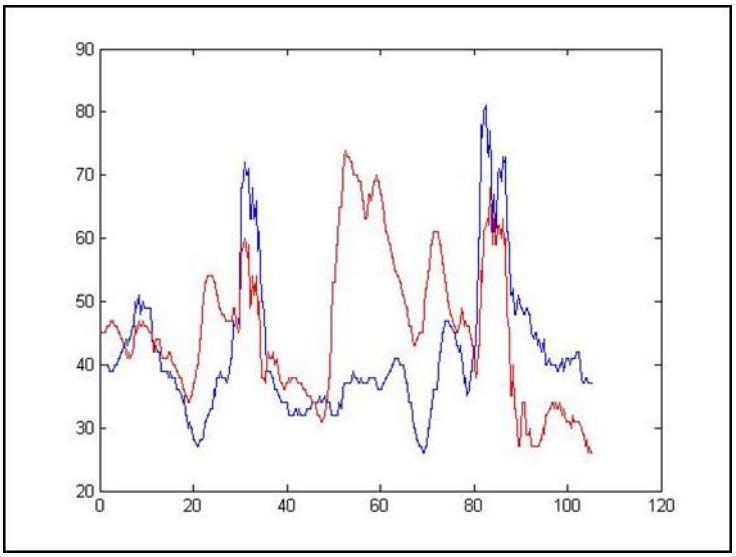

Figure 12. Dependence of coordinates of a point of X and Y on time

At creation of the schedule program designs which carry out reading the coordinates from the file received with video of a fragment were used. Further, using the standard Matlab newlind function which serves for work with linear neural network, calculation of an error of work of such network is made. Then, using the standard Matlab newrb program, the network of radially basic functions is modelled. The maximum number of neurons of $MN = 350$, the accuracy of calculations 10^{-5}.

For comparison, in figure 13, we will provide the schedule constructed at a combination of linear model and network of radially basic functions. For programming of RBF network and evident comparison of value of a mistake with linear model, both algorithms were written in one file.

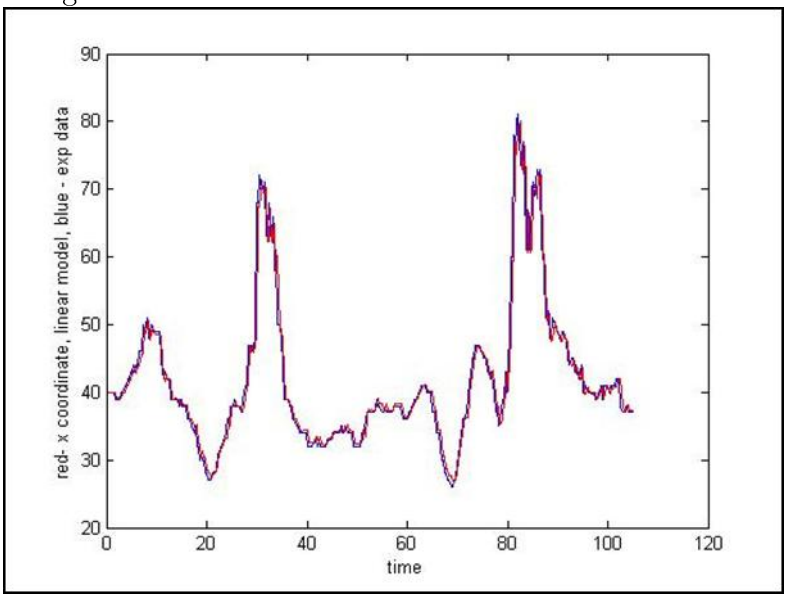

Figure 13. Data processing by linear neural network and network of radial basic functions.

For confirmation of robustness of model series of experiments assuming various input data, increase in number of neurons in RBF networks and increase in an error of calculations were conducted. Results are presented in table.

Table 3.1: Demonstration increase in number of neurons in RBF networks and increase in an error of calculations

n neurons	error of data processing by linear model	Size of an error of processing by RBF network and linear model
351	0.06	0.02
150	0.06	0.03
75	0.06	0.04

Table 1. Comparison of size of an error of data processing by linear model of neural network and processing by RBF network and linear model of neural network.

From table 1 it is possible to is robust that increase in neurons for RBF network in total with linear model of neural network, leads to increase in accuracy of calculations and the model is robastny for different input values. For rather small number of neurons (351) the quality of filtering was significantly increased. It is the main result of this section.

To conclude, let us note that recognition of movements of real biological objects is an important applied task. The combined application of model of linear autoregression with networks of radial basic functions allows us to essentially increase filtering accuracy.

3.2 Search of an object by video camera allowing to reduce number of sensors

In modern society interest in a problem of computer vision increases. The extensive review of applied tasks, which can be solved by means of computer sight lies in the field of medicine, different applied tasks, military solutions and many other areas, see [67].

However, the special attention is required by applied tasks, which arise at human activity. Among many, an important problem is car parking. To resolve this problem different sensors, in particular, a laser ranging sensor can be used. In principle, systems the parctronic and the systems of the automatic car parking can work without participation of the driver work. Nonetheless these technical means are not capable to make beforehand a decision on a possibility of the parking in the conditions of a distance from desirable space, for example, it is required to stop by in the yard and the driver should understand first there is a place, or not. It is especially important in the conditions of the complicated entrance to the destination, lack of an opportunity to be unrolled. The narrow yard or features of an automobile parking in shopping centers can be an example of such place. In the long term developments of this problem the method can be applied to movements of heavy-load cars – defining a possibility of journey with bulky goods.

The solution can be found by the use of cameras which will be able to recognize existence of a free space by means of special algorithms and to notify the driver on expediency of entrance. For recognition it is proposed to use independently written software on the basis of OpenCV library. This library by means of Haar's cascade allows us to look for objects by the principle of the sliding window. To advantages of this method costs carry the low probability of false operation, and to shortcomings – a progressive tense of training of a cascade. The training selection can be on the basis of photos of the surrounding area. Training of a cascade will happen to the help of the utilities delivered complete with OpenCV.

After a preparatory stage connection of a cascade to the software and the USB camera can be made. This module is written in the C# programming language. By means of EmguCV, operation accuracy about 70 % is reached. The space found for the parking is designated by the black rectangle drawn over a frame that reports to the driver about a possibility of the parking of the car.

Descript of method.

In the tasks similar to stated above, one of the main difficulties is that using different technical solutions, such as lasers, cameras, different sensors of volume, we receive the ambiguous, sometimes mutually not interpreted data. It means that when we use one of types of sensors, the wrong result is possible.

The most perspective approach is to use computer vision by a digital video camera. Using mathematical algorithms, in particular, neural networks, we can bring closer "a computer eye" to the present human. One of successful methods in this area Viola-Jones's method [68]. By means of Viola's - Jones method the problem of object recognition is successfully solved by a criteria in which the input cascade of a method is trained. It allows us to create the software for a specific applied objective. Using the description of opportunities of the software by Viola-Jones's method for a task of the analysis of a possibility of the parking of the car in the pre-determined location.

We use so-called Haar's cascade [70]. in Viola's - Jones method, to train the Haar cascade it is necessary to produce the training by the selection of negative and positive examples (superviser training). In case of a solution of other problems, for example, of calculation of number of cars, either facial recognition, or search specific different from an object background, we can apply the same method, but use other training examples. In this task, it is necessary to select a background component, ignoring other objects on the image moving on an input. Further, it is necessary to separate the carriageway from the real parking lot. These, and many other problems arising on the task solution course are considered in our approach. The Open CV library of the last available version at the time of development (OpenCv 3.0 beta) is applied to resolve the task. The program produces up to 70% of positive (faultless) results .

Haar's sign and Viola's – Jones method in Open CV library. The basic principles, to implement our method, are as follows:

- the image is represented in the integrated form. This principle allows us to

select required areas and objects quickly enough;

- we use the signs of Haar selection of a object on the image;

- The busting [73] is applied – by means of this principle the most important signs for the defined object are selected;

- the cascade of all signs arrives on an input of the qualifier which unambiguously selects result of true or false.

The essential defect of the Viola-Jones method is a long training of the qualifier – at rather large volume of the training selection training can reach up to 3 days when using the PC with an average configuration. However, after training, sweep rate of an object is very high therefore Viola's – Jones method can well works, it was one of reasons to select that method to resolve our task. This method is one of the best ones if we will estimate it on the ratio of sweep rate of an object and speed of work. A low probability of wrong detection and false operation are additional advantages. Furthermore we state a general description of the Viola-Jones algorithm.

The principle of the scanning window is the cornerstone of this method functioning. We use the scanning window as follows.

The image is represented by a matrix consisting of pixels of w*h (width and height), each pixel can take values from 0 to 255 in case of black-and-white the image and from 0 to 255^3 in case of the color image.

The problem of detection of a required object as a part of the digital image comes down to the fact that in the course of work the algorithm notes distinctive signs of an object in working area of the image rectangular signs which describe the found object. rectangle = {x,y,w,h,a}, where x, y are the coordinates describing the i-th center of a rectangle, w is width, h is height, an is a tilt angle between a rectangle and a vertical axis of the image, rectangle is a structure, which contains set of four integers defining arrangement and the size of a rectangle.

The method of the scanning window is applied to images: each input part of image of the scanning window requires application of the qualifier. By means of in advance prepared mini-programs as a part of OpenCV libraries the system of training is completely automated and does not require an additional control in the course of the work.

Let us consider now how to represent the image in an integrated form. For carrying out transformations with data in Viola-Jones's method we use integrated representation of the input image. This type of transformation is also used in wavelet-transformations, and other algorithms [71]. The method put in a basis allows to count total brightness for each of rectangles. The feature consists that time of calculation remains invariable for each such rectangle.

The matrix with sizes coinciding with the sizes of the initial image is the integrated representation of the image. Each entry of a matrix contains the total value of intensity of all pixels. Each entry of a matrix can be calculated as follows:

$$L(x,y) = \sum_{i=0,j=0}^{i \le x, j \le y} * I(i,j)$$

where $I(i, j)$ — the brightness of pixel of the initial image.

Thus, for a matrix of $L[x, y]$ each element contains total value of pixels in a required rectangle from $(0, 0)$ to (x, y), at the same time value of pixel (x, y) consists of the sum of all next pixels except standing on the right and to the left of given (x, y). Time, is necessary for calculation of a matrix linearly and in proportion to the number of pixels in the image, proceeding from it the image submitted integrally becomes possible to be counted for one cycle of pass. The matrix can be calculated by the following:

$$L(x, y) = I(x, y) - -I(x - 1, y - 1)I(x, y - 1)I(x - 1, y)$$

By means of this integrated matrix is possible to calculate the required values of the sums of pixels of the rectangle of any area.

Haar cascade

Let us describe Haar's cascade.

Haar's cascade is object X display f: $X- > D_f$, where X –is a set of objects, D is — the set of admissible values of sign. If signs are $f_1, ..., f_n$ then a vector of signs of $f = (f_1(x), ..., f_n(x))$ is the description of object X . Those sign descriptions can be considered, from a formal mathemactical point of view, as objects. During the work of the Viola-Jones method in a classical form rectangular signs are applied. The general view of these primitives of Haar is presented in figure 14:

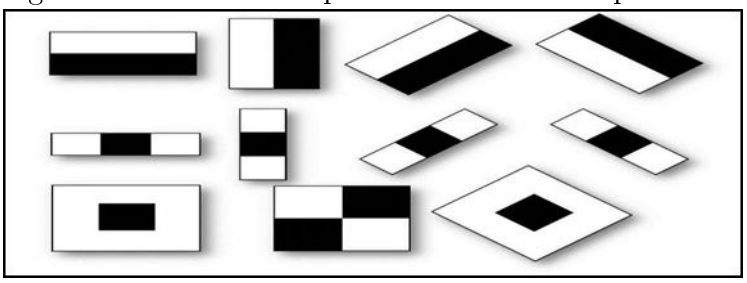

Figure 14. Haar's primitives.

At the heart of OpenCv library besides the main an expanded set of primitives for the modified Viola-Jones's method is used. The general view of primitives is presented in figure 15.

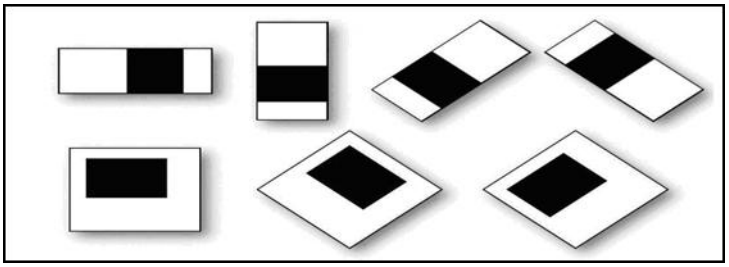

Figure 15. An expanded set of primitives for Viola-Jones's method.

For such sign the calculated value will be F = X-Y where X – total value of brightness of points a zakryvayemykhsvetly part of sign, and Y – total value of brightness of points a part of sign. At calculation of these values integrated

submission of the image is applied. Haar's signs give dot a of brightness on axis X and Y respectively.

Preparation of selected objects for training.

For training of a cascade it is necessary to prepare positive and negative examples. Photos in various weather conditions and various conditions of lighting are required. For the solution of an objective it is required to finish shooting not less than 30 photos which in a consequence will be cut on 230 positive examples and 300 negative ones. Accuracy of work of an algorithm depends on the number of images in the training selection. Experimentally it is found out that for the solution of problem of search of the parking space for the car these values are minimal posssible. Negative examples of selection have to be created in the same environment where recognition will be made and on them there has to be no subject to recognition.

In figure 16 the example of the site of the image used for positive selection is given. The rectangle has allocated places which by the size of the training sample satisfy to the sizes of the average car.

figure 16. Seating of the car in the set space

Options of seating of the car in the specified areas exists much more, than it is presented in the drawing. For the training selection it is necessary to prepare shot for the sliding window with step to 10 pixels. It will allow to train Haar's cascade more effectively. When seating the car to it the side interval for opening of doors from 10% up to 20% of car width was conditionally added.

Training of cascade of Haar. In algorithm the model of machine learning is used. Training of the car is special process of receiving by the module of new knowledge. In Viola's method of Jones training solves problem of classification.

To start training it is necessary to create two folders with examples. For each such folder it is required to create the special text file in which the prepared images will be described. For the file with negative examples just it is necessary to number images from 0 to 300 with the indication of expansion.

For positive examples for training it is required to specify not only expansion and serial number of example, but also area which contains object. For example,

for the file with the name "1.bmp" in the file of the description line "1.bmp 0 0 60 100" registers. Numerical values are coordinates of limits of the image as positive examples are prepared only with the maintenance of required object. It is possible to specify in the file of the description several objects at once if the image supports them. In that case it is necessary to specify line "1.bmp 0 0 60 100 60 0 120 200" where the second four of numbers is coordinates of the second object. The most convenient option is use of one image as example.

Training happens in two stages. At the first stage all positive images are provided to the general format. In the folder, the corresponding digit capacity of the processor in the OpenCV folder there is an opencv_createsamples.exe utility. Work with this utility happens in a console mode and for creation of a vector of images it is necessary to enter the following command:

-info C:\haar\Good.dat -vecsamples.vec -w 40 -h 60, where

- info C:\haar\Good.dat – a way to the file with the description of the list of positive images of selection with the description of coordinates of an object;

- vecsamples.vec – the file name in which the vector containing a set of positive images for selection will be saved. The file will be created in the directory where there is opencv_createsamples.exe;

- w 40 - h 60 — the approximate size of a template for the sliding window which will be able to contain an object. The template should correspond to proportions of the selected object. The size of a template should not be big. The template size directly influences training speed.

As a result of a program runtime the samples.vec file in which there are all positive images is created. For creation of a cascade "opencv_traincascade.exe", Operating time on the computer of the following configuration is used, takes about 6 hours: CPU AMD Phenom II X6 2.6 GHz/6core DDR3 8Gb HDD 1Tb.

For a program runtime it is necessary to enter the following command in a console mode:

-data haarcascade -vecsamples.vec -bg E C:\haar\Bad.dat -numStages 16 -minhitrate 0.999 -maxFalseAlarmRate 0.4 -numPos 200 -numNeg 500 -w 40 -h 60 -mode ALL -precalcValBufSize 2048 -precalcIdxBufSize 2048

where:

- datahaarcascade — the folder address for creation of the trained cascade;

- vecsamples.vec — the file address with positive examples;

- bg C:\haar\Bad.dat — the address of the file description of negative examples;

- numStages 16 — the number of levels of a cascade trained by the program; - minhitrate 0.999 — coefficient of quality of training;

- maxFalseAlarmRate 0.5 — the level of false operations; - numPos 200 — quantity of positive examples. 80% of the available positive files are specified;

- numNeg 300 — quantity of negative examples;

- w 40 - h 60 — the primitive size;

- mode ALL — allows to include use of all set Haar signs;

- precalcValBufSize 2048

- precalcIdxBufSize 2048 — the memory selected under process;

In this task it it is necessary to include full range of signs of Haar as cars can stand at an angle and to complicate search of the parking space.

Files with the description of stages of training and the output trained file of a cascade are by results created.

Work with the trained cascade. For work with a cascade by means of the video camera it was required to write the program which will be able to provide connection of the USB webcam, to create a form for work and to connect algorithms of processing of a cascade. For implementation the C# language is selected.

The new empty C# project is created, then in the Uses module Emgu libraries are connected. Emgu is interpretation of Open CV for work with the C# programming language.

It is required to connect in the project also files of libraries of normal library OpenCv.

After that, in the main function of the program there is a definition of variables.

Then, on the timer there is a capture of a frame with input a flow from the camera which then will be transformed to the black-and-white image. It is caused by the fact that for economy of memory it is necessary to squeeze an input flow (each frame). The frame is compared to a cascade and in case of affirmative answer starts function of posing of a black rectangle over an entrance flow. If visualization of result is not required, this function will allow to make a decision of the notification of the driver on a possibility of the parking.

Testing of the program. The final program allows to define the parking space, with an accuracy about 70% is free or taken. It means that from an entrance flow of the stationary camera at change of external conditions and at redefinition of a signal from the camera with a frequency about 1 second there is a lack of signs of the selected area in 30 of 100 frames. It does not involve dangerous effects as not the positive, but negative result is given. Considering constantly changing conditions – open street space, we believe that this result satisfies to an objective.

One of perspective applications of the program consists in realization of additional modules with use of SMS of informing on demand, with providing information on existence of the empty parking space. This approach will be repaid in case of difficult arrival and impossibility of sharp turn in the conditions of narrow drives.

Use of cascade of Haar allows to define objects with sufficient accuracy. The program created in work satisfies to objectives of search of the parking space for the car.

For increase in accuracy of work of algorithm it is possible to use parallel calculations for increase in speed of training of cascade, and improvement of technology of busting other algorithms. Increase in accuracy of recognition can be increased combination of various methods.

Prospects for improvement of the program can consist that it is possible to create the module which will allow to notify remotely the driver on possibility of the parking, without carrying out arrival to the yards or parkings, difficult for

maneuvering.

Practical application of method also lies in the sphere of collecting statistical data on filling of the parking by cars. Unlike simple recalculation of quantity of the empty fenced seats, becomes possible to make assessment of real number of cars which can be located in the set space.

3.3 Conclusion and discussion

In this thesis, we have proposed a complete analytic theory of maximally flexible and switchable Hopfield networks. We shown that dynamics of a network with n slow components $v_1, ..., v_n$ can be reduced to a system of n differential equations defined by a smooth n dimensional vector field $F(v)$. If these slow components are hubs, i.e., they are connected with a number of other weakly connected nodes (satellites) and center-satellite interactions dominate inter-satellite forces, then the network becomes maximally flexible. Namely, by adjusting only center-satellite interactions we can obtain smooth F of arbitrary forms.

These networks are also maximally switchable. We describe networks of a special architecture, which contains a controller hub. By changing the state of this hub and the hub response time parameter ξ one can completely change the network dynamics from an unique global attractive steady state to any combination of periodic or chaotic attractors.

Our results provide a rigorous framework for the idea that centralized networks are flexible. We also propose mechanisms for switching between attractors of these networks with controller hubs. In functional genomics there are numerous examples when transitions between attractors of gene regulatory networks can be triggered by controller proteins having multiple states sometimes resulting from interactions with micro-RNA satellites (18). Similarly, neurons having multiple internal states can trigger phase transitions of brain networks suggesting that single neuron activation could be used for neural network control (28).

The proofs of our results are constructive and are based on an algorithm allowing the network reconstruction. This algorithm has several potential applications in biology. Identified networks can be used to study emergent network properties such as robustness, controllability and switchability. Gene networks with the desired switchability properties could be build by synthetic biology tools for various applications in biotechnology. Furthermore, maximal switchable network models can be used in neuroscience to relate structure and function in the brain activity, or in genetics to explain how a minimal number of mutations can induce large phenotypic changes from one type of adaptive behavior to another one.

Appendix A

Appendix

VakMorozDMD1

```
function  [X,Y, Xshift] = VakMorozDMD1(Nmark,  T0, M1, Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8,
for m=1:8
q=load([['Z',num2str(m)],'.mat']);
Y=eval (['Z' num2str(m)]);
    for j=1: M1
     time(j)=j;
           X(j, m)=Y(j+T0);

    Xshift(j,m)=Y(j+1+T0);
   end;
end;
   X=transpose(X);
Xshift=transpose(Xshift);
end
```

DMDMine

```
function [Phi,omega,lambda,b,Xdmd, U, S, V, Atilde] = DMDMine(X1,X2,r,dt)
% Computes the Dynamic Mode Decomposition of X1, X2
% INPUTS:
% X1 = X, data matrix
% X2 = X', shifted data matrix
% Columns of X1 and X2 are state snapshots
% r = target rank of SVD
% dt = time step advancing X1 to X2 (X to X')
% OUTPUTS:
% Phi, the DMD modes
```

```
% omega, the continuous-time DMD eigenvalues
% lambda, the discrete-time DMD eigenvalues
% b, a vector of magnitudes of modes Phi
% Xdmd, the data matrix reconstrcted by Phi, omega, b
%% DMD
[U, S, V] = svd(X1, 'econ');
r = min(r, size(U,2));
U_r = U(:, 1:r);
S_r = S(1:r, 1:r);
V_r = V(:, 1:r);
Atilde = U_r' * X2 * V_r / S_r; % low-rank dynamics
[W_r, D] = eig(Atilde);
Phi = X2 * V_r / S_r * W_r; % DMD modes
lambda = diag(D); % discrete-time eigenvalues
omega = log(lambda)/dt; % continuous-time eigenvalues
%% Compute DMD mode amplitudes b
x1 = X1(:, 1);
b = Phi\x1;
%% DMD reconstruction
mm1 = size(X1, 2); % mm1 = m - 1
time_dynamics = zeros(r, mm1);
t = (0:mm1-1)*dt; % time vector
for iter = 1:mm1,
    time_dynamics(:,iter) = (b.*exp(omega*t(iter)));
end;
Xdmd = Phi * time_dynamics;
tildeA=ctranspose(U)*X2*V*inv(S);
```

MHM1DMD

```
function [P,TargetX,TargetY,TargetZ, TappX, TappY, TappZ, Time,epsX,epsY,epsZ]= MHM1DMD(X1,
tic;
X=X1;
Y=Y1;
Z=Z1;
N=length(X);
nomer=1;
MN1=MN(1);
MN2=MN(2);
MN3=MN(3);
DF1=DF(1);
DF2=DF(2);
DF3=DF(3);
```

```
for j=1:Npoint
    Time(j)=j;
    TargetX(j)=X(j+Nbeg, nomer);
    TargetY(j)=Y(j+Nbeg, nomer);
    TargetZ(j)=Z(j+Nbeg, nomer);
    for k=1:kdeg
    P(k,j)=cos(om1(k)*j);
    end;
end;
trained_net=newrb(P, TargetX, goal, spd, MN1, DF1);
hold off;
TappX=sim(trained_net, P);
z=TargetX -TappX;
epsX=norm(z)/norm(TargetX);
trained_net=newrb(P, TargetY, goal, spd, MN2, DF2);
TappY=sim(trained_net, P);
z=TargetY -TappY;
epsY=norm(z)/norm(TargetY);
trained_net=newrb(P, TargetZ, goal, spd, MN3, DF3);
TappZ=sim(trained_net, P);
z=TargetZ -TappZ;
epsZ=norm(z)/norm(TargetZ);
%A_IW = trained_net.IW
%celldisp(A_IW)
%A_LW = trained_net.LW;
%celldisp(A_LW);
%celldisp(b)
toc;
end

frequencygeneration

function [ Om1] = frequencygeneration( kdeg, Ns, om1max );
for j=1:Ns
    'Test', j
 for l=1:kdeg
Om1(l, j)=om1max(l)*rand;
 end;
end

apprToSegOptVar2

tic;
```

```
Nf=5;
kdeg=2;
spd=0.3;
Nbeg=1;
Npoint=2000;
% Npoint is the maximal number of points in the segment;
om1max=[0.03, 0.02];
%om1max(1) is the maximal possible frequency for 1-th oscillator;
%om1max(1) is the maximal possible frequency for 2-th oscillator;
goal=0.000001;
Natt=10;
% Natt is the  number of attempts for the random search of frequencies;
om1opt=om1max;
Eopt=100;
for kk=1:Natt
om1(1)=rand*om1max(1);
om1(2)=rand*om1max(2);
Om(kk,1)=om1(1);
Om(kk, 2)=om1(2);
for i=1 : length(cuts)+1
   X(i,1)=cell(mots(i,1));
   for k=1 : length(X{i}.jointTrajectories)
       J(k,i) = X{i}.jointTrajectories(k,1);
       L=J{k,i};
       A = zeros(length(L),1);
       B = zeros(length(L),1);
       C = zeros(length(L),1);
       for j=1:length(L)
           A(j,1)=L(1,j);
           B(j,1)=L(2,j);
           C(j,1)=L(3,j);
       end
       %[P,TargetX,TargetY,TargetZ, TappX, TappY, TappZ, Time,epsX,epsY,epsZ]= MotionHandle
       [TargetX,TargetY,TargetZ, TappX, TappY, TappZ, Time,epsX,epsY,epsZ]= MotionHandleAM(
       Ex(i, k)=epsX;
       Ey(i,k)=epsY;
       Ez(i,k)=epsZ;
   end
end
n1=length(cuts)+1;
n2=length(X{i}.jointTrajectories);
aa=sqrt(n1*n2);
errorX=norm(Ex)/aa;
```

```
errorY=norm(Ey)/aa;
errorZ=norm(Ez)/aa;
error(kk)=sqrt((errorX^2 + errorY^2 + errorZ^2)/3);
if(error(kk) < Eopt) Eopt=error(kk); om1opt=om1; else end;
kk
error(kk)
end;
toc;

MotionHandleAM

function [TargetX,TargetY,TargetZ, TappX, TappY, TappZ, Time,epsX,epsY,epsZ]= MotionHar
tic;
Nfunc=Nf^2;
nomer=1;
X=A;
Y=B;
Z=C;
N=length(X);
Npoint=min(Npoint, N-Nbeg);
[ P ] = BasicFunc(Npoint, kdeg, om1, Nf, spd);
size(P);
for j=1:Npoint
    Time(j)=j;
    TargetX(j)=X(j+Nbeg, nomer);
   TargetY(j)=Y(j+Nbeg, nomer);
    TargetZ(j)=Z(j+Nbeg, nomer);
end;
trained_net=newlind(P, TargetX);
TappX=sim(trained_net, P);
z=TargetX -TappX;
epsX=norm(z)/norm(TargetX);
trained_net=newlind(P, TargetY);
TappY=sim(trained_net, P);
z=TargetY -TappY;
epsY=norm(z)/norm(TargetY);
trained_net=newlind(P, TargetZ);
TappZ=sim(trained_net, P);
z=TargetZ -TappZ;
epsZ=norm(z)/norm(TargetZ);
end

OptfreqNRBrandM
```

```
function [  epsmin, opt ] = OptfreqNRBrandM( X1, Y1, Z1, Om1, om1max, Ns, Npoint,  Nbeg, om
epsmin=100;
for j=1:Ns
   'Test', j
 for l=1:kdeg
om1(l)=Om1(l,j);
 end;
[P,TargetX,TargetY,TargetZ, TappX, TappY, TappZ, Time,epsX,epsY,epsZ]= MHM1(X1, Y1, Z1, om1
eps=sqrt(epsX^2 + epsY^2 +epsZ^2)/sqrt(3);
if(eps < epsmin) epsmin=eps;
    for kk=1:kdeg
    opt(kk)=om1(kk); end;
else end;
    end;
end

Segdecomposition

function [ Nseg] = Segdecomposition( Nbeg, Npoint, NumberofSeg )
SegLength=(Npoint - Nbeg)/NumberofSeg;
for i=1:(NumberofSeg+1)
   Nseg(i)=round( Nbeg+ (i-1)*SegLength)+1;
end;
end
```

Bibliography

[1] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser. Illustrative Flow Visualization: State of the Art, Trends and Challenges. *In Eurographics 2012 - State of the Art Reports*, 2012.

[2] A. Gams, A. J. Ijspeert, S. Schaal, and J. Lenarcic. On-line learning and modulation of periodic movements with nonlinear dynamical systems. *Autonomous Robots*, 27:3–23, 2009.

[3] A. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *International Journal of Robotics Research*, 21(4):642–653, 2008.

[4] A. Ijspeert. Central pattern generators for locomotion control in animals and robots: A review. *Neural Networks*, 21(4):642–653, 2008.

[5] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. *IEEE International Conference on Robotics and Automation*, pages 1398–1403, 2002.

[6] N. I. Achieser. *Theory of approximation*. Courier Corporation, 2013.

[7] R. Albert and A. L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.

[8] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *nature*, 406(6794):378–382, 2000.

[9] B. L. Ho and R. E. Kalman. Effective construction of linear statevariable models from input/output data. *In Proceedings of the 3rd Annual Allerton Conference on Circuit and System Theory*, pages 449–459, 1965.

[10] B. O. Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[11] B. W. Brunton, L. A. Johnson, J. G. Ojemann, and J. N. Kutz. Extracting spatial temporal coherent patterns in large-scale neural recordings using dynamic mode decomposition. *Journal of neuroscience methods*, (258):1–15, 2016.

[12] Y. Bar-Yam and I. R. Epstein. Response of complex networks to stimuli. *Proceedings of the National Academy of Sciences of the United States of America*, 101(13):4341–4345, 2004.

[13] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *Information Theory, IEEE Transactions on*, 39(3):930–945, 1993.

[14] J. Bascompte. Networks in ecology. *Basic and Applied Ecology*, 8(6):485–490, 2007.

[15] C. Heine, H. Leitte, M. Hlawitschka, F. Iuricich, L. D. Floriani, G. Scheuermann, H. Hagen, and C. Garth. A survey of topologybased methods in visualization. *Computer Graphics Forum*, 35(3):643–667, 2016.

[16] C. W. Rowley, I. Mezic, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, (641), 2009.

[17] J. M. Carlson and J. Doyle. Complexity and robustness. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2538–2545, 2002.

[18] R. W. Carthew. Gene regulation by microrнas. *Current opinion in genetics & development*, 16(2):203–208, 2006.

[19] D. R. Chialvo. Emergent complex neural dynamics. *Nature physics*, 6(10):744–750, 2010.

[20] CMU. Carnegie Mellon University Graphics Lab: Motion Capture Database, 2013.

[21] R. Cohen, K. Erez, D. Ben-Avraham, and S. Havlin. Breakdown of the internet under intentional attack. *Physical review letters*, 86(16):3682, 2001.

[22] S. P. Cornelius, W. L. Kath, and A. E. Motter. Realistic control of network dynamics. *Nature communications*, 4, 2013.

[23] N. J. Cowan, E. J. Chastain, D. A. Vilhena, J. S. Freudenberg, and C. T. Bergstrom. Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PloS one*, 7(6):e38398, 2012.

[24] D. Knight and G. Mallinson. Visualizing unstructured flow data using dual stream functions. *IEEE Transactions on Visualization and ComputerGraphics*, 2(4):355–363, 1996.

[25] G. Deco and V. K. Jirsa. Ongoing cortical activity at rest: criticality, multistability, and ghost attractors. *The Journal of Neuroscience*, 32(10):3366–3375, 2012.

[26] E. Theodorou, J. Buchli, and S. Schaal. A generalized path integral control approach to reinforcement learning. *Journal of machine learning research*, 11:3137–3181, 2010.

[27] R. Edwards, A. Beuter, and L. Glass. Parkinsonian tremor and simplification in network dynamics. *Bulletin of mathematical biology*, 61(1):157–177, 1999.

[28] S. Fujisawa, N. Matsuki, and Y. Ikegaya. Single neurons can induce phase transitions of cortical recurrent networks with multiple internal states. *Cerebral Cortex*, 16(5):639–654, 2006.

[29] J. Gao, Y.-Y. Liu, R. M. D'Souza, and A.-L. Barabási. Target control of complex networks. *Nature communications*, 5, 2014.

[30] H. Hoffmann, P. Pastor, D.-H. Park, and S. Schaal. Biologically inspired dynamical systems for movement generation: automatic realtime goal adaptation and obstacle avoidance. *International Conference on Robotics and Automation*, 2009.

[31] H. K. Khalil. Nonlinear systems. *Prentice-Hall, Upper Saddle River*, 2002.

[32] H. Kimura, Y. Fukuoka, and A. Cohen. Adaptive dynamic walking of a quadruped robot on natural ground based on biological concepts. *International Journal of Robotics Research*, 26(5):475–490, 2007.

[33] D. Henry and D. Henry. Geometric theory of semilinear parabolic equations. 1981.

[34] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 4 1982.

[35] J. J. Hopfield, D. W. Tank, et al. Computing with neural circuits- a model. *Science*, 233(4764):625–633, 1986.

[36] Huang, Sui and Ernberg, Ingemar and Kauffman, Stuart. Cancer attractors: A systems view of tumors from a gene network dynamics and developmental perspective. *Seminars in Cell Developmental Biology* , 20(7):869 – 876, 2009.

[37] I. Kovacic, M. Brennan. The Duffing equation: Nonlinear oscillators and their behaviour. *Wiley*, 2011.

[38] I. Mezi'c. Analysis of fluid flows via spectral properties of the Koopman operator. *Annual Review of Fluid Mechanics*, (45):357–378, 2013.

[39] J. Gosek and J. N. Kutz. Dynamic mode decomposition for realtime background/foreground separation in video. 2013.

[40] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.

[41] J. N. Auke Jan Ijspeert and S. Schaal. Learning rhythmic movements by demonstration using nonlinear oscillators. *Proceedings of the IEEE/RSJ Int. Conference on Intelligent Robots and Systems*, pages 958–963, 2002.

[42] J. N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance*, 8(5):620–627, 1985.

[43] J. N. Kutz, J. Grosek, S. Brunton, X. Fu, and S. Pendergrass. Using dynamic mode decomposition for real-time background/foreground separation in video. *CoRR*, 2014.

[44] J. N. Kutz, S. L. Brunton, B. L. Brunton, J. L. Proctor. Dynamic Mode Decomposition. *SIAM*, 2016.

[45] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. Online movement adaptation based on previous sensor experiences. *Robotics and Autonomous Systems*, 47:79–91, 2003.

[46] H. Jeong, S. P. Mason, A. L. Barabási, and Z. N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411(6833):41–2, 5 2001.

[47] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási. The large-scale organization of metabolic networks. *Nature*, 2000.

[48] T. Jia and A.-L. Barabási. Control capacity and a random sampling method in exploring controllability of complex networks. *Scientific reports*, 3, 2013.

[49] Johannes Ernesti, Ludovic Righetti, Martin Do, Tamim Asfour, Stefan Schaal. 2012.

[50] A. V. Jorg Menche and R. Lipowsky. Dynamical processes on dissortative scale-free networks. *EPL (Europhysics Letters)*, 89(1):18002, 2010.

[51] K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, (56):345–353, 1987.

[52] Y. Kifer. General random perturbations of hyperbolic and expanding transformations. *Journal d'Analyse Mathematique*, 47(1):111–150, 1986.

[53] B. Krüger, A. Vögele, T. Willig, A. Yao, R. Klein, and A. Weber. Efficient unsupervised temporal segmentation of motion data. *IEEE Transactions on Multimedia*, 19(4):797–812, Apr. 2017.

[54] L. John. The Structure of Inhomogeneous Turbulent Flows. *In Atmospheric turbulence and radio wave propagation*, pages 166–178, 1967.

[55] L. Righetti and A. Ijspeert. Programmable central pattern generators: an application to biped locomotion control. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1585–1590, 2006.

[56] Y.-C. Lai. Controlling complex, non-linear dynamical networks. *National Science Review*, 1(3):339–341, 2014.

[57] L. Li, J. Xu, D. Yang, X. Tan, and H. Wang. Computational approaches for microRNA studies: a review. *Mammalian Genome*, 2010.

[58] Z. Li and J. Hopfield. Modeling the olfactory bulb and its neural oscillatory processings. *Biological cybernetics*, 61(5):379–392, 1989.

[59] C. T. Lin. Structural controllability. *Automatic Control, IEEE Transactions on*, 19(3):201–208, 1974.

[60] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[61] M. Grilli, P. J. Schmid, S. Hickel, and N. A. Adams. Analysis of unsteady behaviour in shockwave turbulent boundary layer interaction. *Journal of Fluid Mechanics*, (700):16–28, 2012.

[62] W. Maass, G. Schnitger, and E. D. Sontag. On the computational power of sigmoid versus boolean threshold circuits. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 767–776. IEEE, 1991.

[63] A. E. Motter. Networkcontrology. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(9):097621, 2015.

[64] N. B. Erichson, S. L. Brunton, and J. N. Kutz. Compressed dynamic mode decomposition for background modeling. *Journal of Real-Time Image Processing*, 2016.

[65] T. Nepusz and T. Vicsek. Controlling edge dynamics in complex networks. *Nature Physics*, 8(7):568–573, 2012.

[66] M. E. Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[67] H. A. Orr. The genetic theory of adaptation: a brief history. *Nature Reviews Genetics*, 6(2):119–127, 2005.

[68] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, (656):5–28, 2010.

[69] P. J. Schmid. Application of the dynamic mode decomposition to experimental data. *Experiments in Fluids*, (50):1123–1130, 2011.

[70] P. J. Schmid, D. Violato, and F. Scarano. Decomposition of timeresolved tomographic PIV. *Experiments in Fluids*, (52):1567–1579, 2012.

[71] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal. Online movement adaptation based on previous sensor experiences. *International Conference on Robotics and Automation*, 2011.

[72] Y. Pan and X. Li. Structural controllability and controlling centrality of temporal networks. *PloS one*, 9(4):e94998, 2014.

[73] F. Pasqualetti, S. Zampieri, and F. Bullo. Controllability metrics, limitations and algorithms for complex networks. *Control of Network Systems, IEEE Transactions on*, 1(1):40–52, 2014.

[74] T. Pereira. Hub synchronization in scale-free networks. *Physical Review E*, 82(3):036201, 2010.

[75] T. Pereira, D. Eroglu, G. B. Bagci, U. Tirnakli, and H. J. Jensen. Connectivity-driven coherence in complex networks. *Physical review letters*, 110(23):234103, 2013.

[76] S. Perumal and A. A. Minai. Stable-yet-switchable (sys) attractor networks. In *Neural Networks*, pages 2509–2516, 2009.

[77] P. Poláčik. Complicated dynamics in scalar semilinear parabolic equations in higher space dimension. *Journal of differential equations*, 1991.

[78] R. Bujack, M. Hlawitschka, and K. I. Joy. Topology-inspired galilean invariant vector field analysis. *IEEE Pacific Visualization Symposium (PacificVis)*, 33(3):72–79, 2016.

[79] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in flow visualization: Dense and texturebased techniques. *Computer Graphics Forum*, 23(2):203–221, 2004.

[80] D. Ruelle and C. Dewitt-Morette. Elements of differentiable dynamics and bifurcation theory. *Physics Today*, 1990.

[81] D. Ruelle and C. Dewitt-Morette. Elements of differentiable dynamics and bifurcation theory. *Physics Today*, 1990.

[82] J. Ruths and D. Ruths. Control profiles of complex networks. *Science*, 343(6177):1373–1376, 2014.

[83] S. A. Vakulenko. A system of coupled oscillators can have arbitrary prescribed attractors. *Journal of Physics A: Mathematical and General*, 27(7):2335–2349, 1994.

[84] S. Abu and J. S. Hyung. Dynamic mode decomposition of turbulent cavity flows for self-sustained oscillations. *International Journal of Heat and Fluid Flow*, 32(15):1098–1110, 2011.

[85] C. Stam, B. Jelles, H. Achtereekte, S. Rombouts, J. Slaets, and R. Keunen. Investigation of eeg non-linearity in dementia and parkinson's disease. *Electroencephalography and clinical neurophysiology*, 95(5):309–317, 1995.

[86] Stefan Schaal. Dynamic Movement Primitives. *A Framework for Motor Control in Humans and Humanoid Robotics*.

[87] J. Stroud, M. Barahona, and T. Pereira. Dynamics of cluster synchronisation in modular networks: Implications for structural and functional networks. In *Applications of Chaos and Nonlinear Dynamics in Science and Engineering-Vol. 4*, pages 107–130. Springer, 2015.

[88] J. Sun and A. E. Motter. Controllability transition and nonlocality in network control. *Physical Review Letters*, 110(20):208701, 2013.

[89] M. Talagrand. Rigorous results for the hopfield model with many patterns. *Probability theory and related fields*, 110(2):177–275, 1998.

[90] T. Tanizawa, G. Paul, R. Cohen, S. Havlin, and H. E. Stanley. Optimization of network robustness to waves of targeted and random attacks. *Physical review E*, 71(4):047101, 2005.

[91] J. Tautges, A. Zinke, B. Kruger, J. Baumann, A. Weber, T. Helten, M. Muller, H.-P. Seidel, and B. Eberhardt. Motion reconstruction using sparse accelerometer data. *ACM Trans. Graph.*, 30(3):18:1–18:12, May 2011.

[92] R. Temam. Infinite Dimensional Dynamical Systems in Mechanics and Physics. *Applied Mathematical Sciences*, 1988.

[93] S. Vakoulenko. Complexité dynamique de reseaux Hopfield. *C. R. Acad. Sci. Paris Sér. I Math.*, 335, 2002.

[94] S. Vakulenko. Dissipative systems generating any structurally stable chaos. *Advances in Differential Equations*, 5(7-9):1139–1178, 2000.

[95] S. Vakulenko and O. Radulescu. Flexible and robust patterning by centralized gene networks. *Fundamenta Informaticae*, 118(4):345–369, 2012.

[96] S. A. Vakulenko and O. Radulescu. Flexible and robust networks. *Journal of bioinformatics and computational biology*, 10(02):1241011, 2012.

[97] M. Viana. Dynamics : A probabilistic and geometric perspective. *Documenta Mathematica*, Extra Volume ICM:557–578, 1998.

[98] J. VOHRADSKÝ. Neural network model of gene expression. *The FASEB Journal*, 15(3):846–854, 2001.

[99] F.-X. Wu, L. Wu, J. Wang, J. Liu, and L. Chen. Transittability of complex networks and its applications to regulatory biomolecular networks. *Scientific reports*, 4, 2014.

[100] G. Yan, J. Ren, Y.-C. Lai, C.-H. Lai, and B. Li. Controlling complex networks: How much energy is needed? *Physical review letters*, 108(21):218703, 2012.

[101] L.-S. Young. Stochastic stability of hyperbolic attractors. *Ergodic Theory and Dynamical Systems*, 6(02):311–319, 1986.

[102] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai. Exact controllability of complex networks. *Nature communications*, 4, 2013.

[103] Z. Ma, S. Ahuja, and C. W. Rowley. Reduced order models for control of fluids using the eigensystem realization algorithm. *Theoretical and Computational Fluid Dynamics*, 25(1):233–247, 2011.