Applications of Input-to-State Stability theory to chaos and non-linear adaptive control.

A. Franci, A. Bicchi, R. Mannella

October 6, 2008

Introduction

In this work we consider the problem of stabilizing complex nonlinear dynamical systems where feedback signals are subject to delays and bandwidth limitations due to the less-than-ideal nature of communication channels used to close the high-level loop, by exploiting a feed forward-feedback scheme inspired to vertebrate-like adaptive control system.

Chapter 1 is a short introduction to control theory and related problems. Section 1.1 presents the concepts of open and closed loop control, the concept of stability and the problem of stabilization through feedback (closed loop) control. A brief sketch of the solution of this problem for linear systems is presented. Section 1.2 presents ideas and motivations of *adaptive* control, i.e control of partially unknown or changing dynamics, and some classical solutions to this problem (direct and indirect adaptive control) along with two examples of adaptive control scheme (the high-gain adaptive control and adaptive sliding control). Finally we present open problems linked to communication constraints of real control schemes and a possible solution, the Embedded Control Design.

Vertebrate movement learning represents a fascinating example of such a control and **Chapter 2** includes a survey of known results about it. Main features are the existence of a *virtual trajectory* of instantaneous fixed point (section 2.3.2) stabilized by a small set of *motor primitives* codified in the spinal cord and activated by feed forward motor commands descending from the central nervous system (section 2.3.3), and the role of reflexes in achieving adaptation and learning (section 2.3.5). In the light of these results, a vertebrate-like adaptive control system is presented (section 2.4). We introduce problems linked with this working scheme, that are (i) stability of nonlinear closed-loop system in the presence of control parameter uncertainties and input disturbances, (ii) possible models and implementations of the neural representation of motor commands and reflexes in the spinal cord and in the central nervous system, looking in particular at the chaotic nature of neural dynamics and possible "meaning representation" applications of *chaos control.*

Chapter 3 is a technical chapter which introduces a powerful tool for analyzing stability properties of nonlinear systems under persistent excitations, the Input-to-State Stability (ISS) (section 3.1) and its relations with control system (3.2), including a possible application to analyze and synthesize a feedback control design with minimal bit rate and a practical example, the nonlinear inverted pendulum on a cart with unknown length.

Chapter 4 is an introduction to chaotic systems and to some of their more peculiar characteristics, in order to exploit the neural-chaotic representation problem. In particular, section 4.1 introduces *ergodicity* and *topological transitivity*, which will turn out to be fundamental for controlling chaos; sections 4.2-4.4 describe instruments and results for analyzing stability properties of nonlinear dynamics, with a strong emphasis on the concept of *unstable periodic orbit*; section 4.5 points out how chaotic dynamics can be seen as shadowing an infinity of unstable periodic orbits (the *chaos skeleton*) in terms of the expectation value of some observables, while section 4.6 does the same in a more heuristic way.

Chapter 5 introduces chaos control and shows how it is naturally (local) Input-to-State Stable, due to the very own nature of chaotic dynamics (sections 5.1-5.2), which would imply the robustness of any neural representation based on it. Section 5.3 verifies this result on a pair of coupled FitzHugh-Nagumo equations, chosen as model neurons. Looking forward to synthesizing a possible vertebrate-like adaptive control system, including reflex adaptation, we also study chaotic synchronization of two pairs of coupled chaotic equations through small perturbations (section 5.4.1).

Finally, in **Chapter 6** we propose a possible adaptive control system based on an ISS synergy controller. We describe problems linked with the approximation of real parameters through synergy activation modification (as in the vertebrate), and propose two possible tuning procedures. The first makes use of an analogical synergy controller (such as vertebrate muscle-skeletal system), while the second is based on a synergy digital controller with minimal bit-rate (more useful for practical implementation).

The main **original contributions** in this thesis are results of sections 3.2.1-3.2.4, theorem 2 (although the proof follows the same line of [2], it has been generilized here to include parameter uncertainties) and Theorem 5. The relative simulations (sections 3.2.7 and 5.3.1) are original contributions. Even though it does not include any new results, section 2.4 proposes an original (up to our knowledge) practical implementation of a vertebrate-like adaptive control system, as well as Chapter 6, where this model (including new results about ISS of control systems and chaos control) is further developed and tested on the inverted pendulum on a chart.

Contents

Chapter 1

Adaptive control theory and open problems

1.1 Control theory

This section is based on [1]

Since the beginnings of Calculus, differential equations have provided an effective mathematical model for a wide variety of physical phenomena. Consider a system whose state can be described by a finite number of real-valued parameters, say $x = (x_1, \ldots, x_n)$. If the rate of change $\dot{x} = dx/dt$ is entirely determined by the state x itself, then the evolution of the system can be modeled by the ordinary differential equation

$$\dot{x} = g(x). \tag{1.1}$$

If the state of the system is known at some initial time t_0 , the future behavior for $t > t_0$ can then be determined by solving a Cauchy problem, consisting of (1.1) together with the initial condition $x(t_0) = x_0$.
We are here taking a spectators point of view: the mathematical model allows us to understand a portion of the physical world and predict its future evolution, but we have no means of altering its behavior in any way. Celestial mechanics provides a typical example of this situation. We can accurately calculate the orbits of moons and planets and exactly predict time and locations of eclipses, but we cannot change them in the slightest amount. Control theory provides a different paradigm. We now assume the presence of an external agent, i.e. a controller, who can actively influence the evolution of the system. This new situation is modeled by a control system, namely

$$\dot{x} = f(x, u), \quad u(\cdot) \in U, \tag{1.2}$$

where U is a family of admissible control functions. In this case, the rate of change $\dot{x}(t)$ depends not only on the state x itself, but also on some external parameters, say $u = (u_1, \ldots, u_m)$, which can also vary in time. The control function $u(\cdot)$, subject to some constraints, will be chosen by a controller in order to modify the evolution of the system and achieve certain preassigned goals, g (for example, steer the system from one state to another, maximize the terminal value of one of the parameters, minimize a certain cost functional, etc...)
The control law can be assigned in two basically different ways. In open loop (Figure 1.1A) form, as a function of time: $t \rightarrow u(t)$, and in closed loop or feedback (Figure 1.1B), as a function of the state: $x \rightarrow u(x)$. Implementing an open loop control $u = u(t)$ is in a sense easier, since the only information needed is provided by a clock, measuring time. On the other hand, to implement a closed loop control $u = u(x)$ one constantly needs to measure the state x of the system.

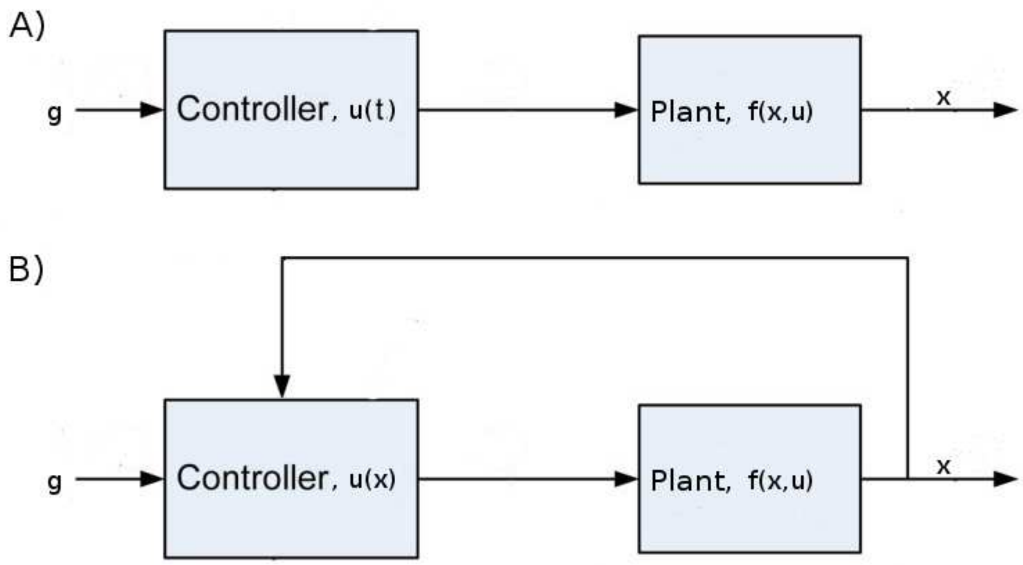


Figure 1.1: Block representation of A) open loop control and B) closed loop control

Designing a feedback control, however, yields some distinct advantages. In particular, feedback controls can be more robust in the presence of random perturbations. For example, assume that we seek a control $u(\cdot)$ which steers the system from an initial state P to the origin. If the behavior of the system is exactly described by (1.1) , this can be achieved, say, by the open loop control $t \rightarrow u(t)$. In many practical situations, however, the evolution is influenced by additional disturbances which cannot be predicted in advance. The actual behavior

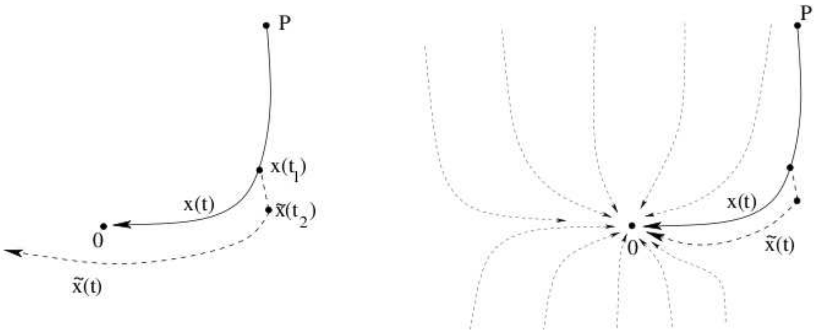


Figure 1.2: External disturbances effect on, left, open loop control and, right, closed loop control

of the system will thus be governed by

$$\dot{x} = f(x, u) + \eta(t), \tag{1.3}$$

where $t \to \eta(t)$ is a perturbation term. In this case, if the open loop control $u = u(t)$ steers the system (1.2) to the origin, this same control function may not accomplish the same task in connection with (1.3), when a perturbation is present. In Figure 1.2(left) the solid line depicts the trajectory of the system (1.2), while the dotted line illustrates a perturbed trajectory $x(\cdot)$. We assumed here that the disturbance $\eta(\cdot)$ is active during a small time interval $[t_1, t_2]$. Its presence puts the system off course, so that the origin is never reached.

Alternatively, one can solve the problem of steering the system to the origin by means of a closed loop control. In this case, we would seek a control function $u = u(x)$ such that all trajectories of the O.D.E.

$$\dot{x} = g(x) = f(x, u(x)) \tag{1.4}$$

approach the origin as $t \to \infty$. This approach is less sensitive to the presence of external disturbances. As illustrated in Figure 1.2(right), in the presence of an external disturbance $\eta(\cdot)$, the trajectory of the system does change, but our eventual goal, steering the system to the origin, would still be attained.

In what follows we will always deal with closed loop systems of the form (1.4). Assume that $f(\overline{x}, 0) \equiv 0$, so that $\overline{x}$ is an equilibrium point when the null control is applied. In general, this equilibrium may not be stable: a trajectory which starts at a point $x_0 \sim \overline{x}$ may get very far from $\overline{x}$ at large times. For many practical situation, an important problem is to design a feedback control

$$u = k(x) \tag{1.5}$$

such that the resulting system

$$\dot{x} = g(x) = f(x, k(x)) \tag{1.6}$$

is asymptotically stable at the point $\overline{x}$.
A formal way to define global asymptotic stability is the Lyapunov stability.

Definition 1 *Consider the differential equation*

$$\dot{x} = g(x) \tag{1.7}$$

$x \in \Re^n$, *let* $x(t, x_0)$ *be the solution with initial data* $x(0) = x_0$. *We say that the system (1.7) is* Lyapunov stable *at the origin if the following holds (see fig.)*

1. *For every* $\epsilon > 0$ *there exists* $\delta > 0$ *such that if* $|x_0| < \delta$ *then for every* $t \geq 0$ *we have* $|x(t, x_0)| < \epsilon$.

2. *For every* $x_0 \in \Re^n$ *we have* $\lim_{t\to\infty} x(t, x_0) = 0$

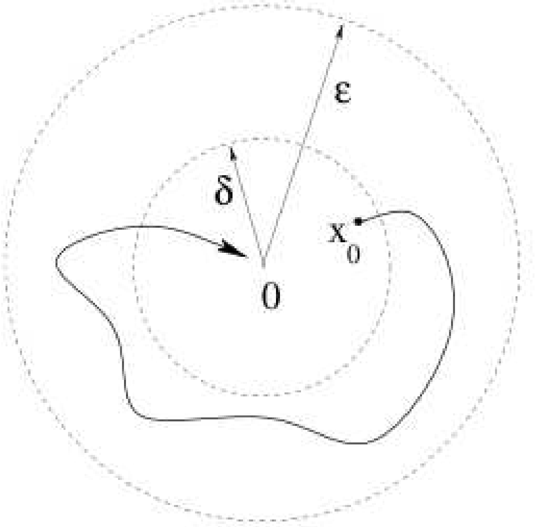


Figure 1.3: Lyapunov stability

For a general nonlinear system, checking its stability is not an easy task. A standard method for proving Lyapunov stability is to construct a positive function that decreases monotonically along all trajectories of the system. We review here the basic ideas of this approach.
Given an open set $Q \in \Re^n$ a $\mathcal{C}^1$ function $V : Q \to \Re$ is called a *Lyapunov function* for (1.7) on Q if following conditions hold.

1. V is proper, i.e. for every $r > 0$ the sub-level set $x : V(x) \leq r$ is compact;

2. V is positive definite, i.e. $V(0) = 0$ and $V(x) > 0$ for every $x \neq 0$;

3. V is strictly decreasing along trajectories of the system: For $x \neq 0$ we have $\nabla V(x)g(x) < 0$.

It can be shown that *Lyapunov stability is equivalent to the existence of a Lyapunov function*(see [1]).
A complete characterization of Lyapunov stability can be given in the case of a linear system with constant coefficients:

$$\dot{x} = Ax, \tag{1.8}$$

which is Lyapunov stable if and only if all the eigenvalues of A have strictly negative real part (see [1]).
The last equivalence can be used to stabilize a linear control system

$$\dot{x} = Ax + Bu, \quad x \in \Re^n, \quad u \in \Re^m, \tag{1.9}$$

where A is an $n \times n$ matrix and B an $n \times m$ matrix. If all the eigenvalues of the matrix A have negative real part, the system is already stable in connection with the null control $u \equiv 0$.
In the case where the uncontrolled system (1.8) is unstable, our aim is to find a linear feedback $u = Fx$, with F an $n \times m$ matrix, such that the resulting linear system

$$\dot{x} = (A + BF)x$$

is Lyapunov stable at the origin. In this case, the control $u = Fx$ is called a *stabilizing feedback*. From the characterization of Lyapunov stability for linear systems, we can conclude that this will be the case if and only all eigenvalues of the matrix $A+BF$ have negative negative real part. The existence of the matrix F is equivalent to the *controllability* of the system, that is the possibility of reaching any point in $\Re^n$ at a given time $t > 0$ starting from the origin through a suitable control (see [1]). In the case of nonlinear control systems of the kind (1.4), one can apply the last result to the linearized system at the origin, in the case where $f(0, 0) = 0$, by setting

$$A = \frac{\partial f}{\partial x}(0, 0), \quad B = \frac{\partial f}{\partial u}(0, 0). \tag{1.10}$$

If the linearized controlled system is Lyapunov stable, then there exists a neighborhood of the origin W, such that the system restricted to W admits a continuous stabilizing feedback.
We will come back to stability properties of nonlinear systems in Chapter 3.

1.2 Adaptive control and communication constraints

In many practical situations the dynamics to be controlled will depend upon a set of parameters. In general, the stabilizing law will depend on this parameters as well. It can happen that these parameters are not exactly known, or that they can change in time (think of a self-guided aircraft which looses weight by consuming fuel). *Adaptive control deals with system having a controller with adjustable parameters and a mechanism for adjusting the parameters . To adapt means to change oneself so that one's behavior will conform to new or changed circumstances.*
Like classical control theory, the performance of adaptive control is generally assessed according to certain rigorous criteria such as stability, convergence and robustness. Stability means that when a system is sufficiently close to equilibrium, the system states can be kept arbitrarily close to the equilibrium point under perturbation and return to the equilibrium point when the perturbation is removed ([13]). Convergence means that given a bounded input, the output will be bounded and tends to a steady state over time. Besides output convergence, convergence of the parameter estimates for the plant (i.e., the process to be controlled) and controller to their true values is also of major concern in adaptive control and system identification problems. Finally, an otherwise stable controller may become unstable in the presence of small disturbances or unmodeled dynamics. Robustness describes the amount of such uncertainty the system can tolerate before controller performance is significantly compromised ([14]; [15]; [18]). We will come back to these concepts in Chapter 3, Section 3.2.
For the moment we will just describe classical solution to face adaptability and related problems. Adaptive control can be mainly classified into direct and indirect adaptive control.
In direct adaptive control, the adaptation law is implemented directly whereas in indirect adaptive control, the adaptation law is implemented by first estimating some unknown plant parameters and/or state variables. Figure 1.4 shows the basic architecture of direct adaptive control. Here, the controller parameters $C(t)$ are adapted directly based on the input/output and control signals, with no explicit estimation of the plant parameters or state variables necessary. In contrast, in indirect adaptive control (Fig. 1.4) the unknown plant parameters or state variables are continuously updated based on some parameter estimation or state estimation law. The resultant estimates are then used to compute the controller parameters at each time t.

In general, indirect adaptive control is more flexible than direct adaptive control because the estimation law and control law can be designed separately ([15]). However, stability and convergence are not guaranteed because the estimated plant

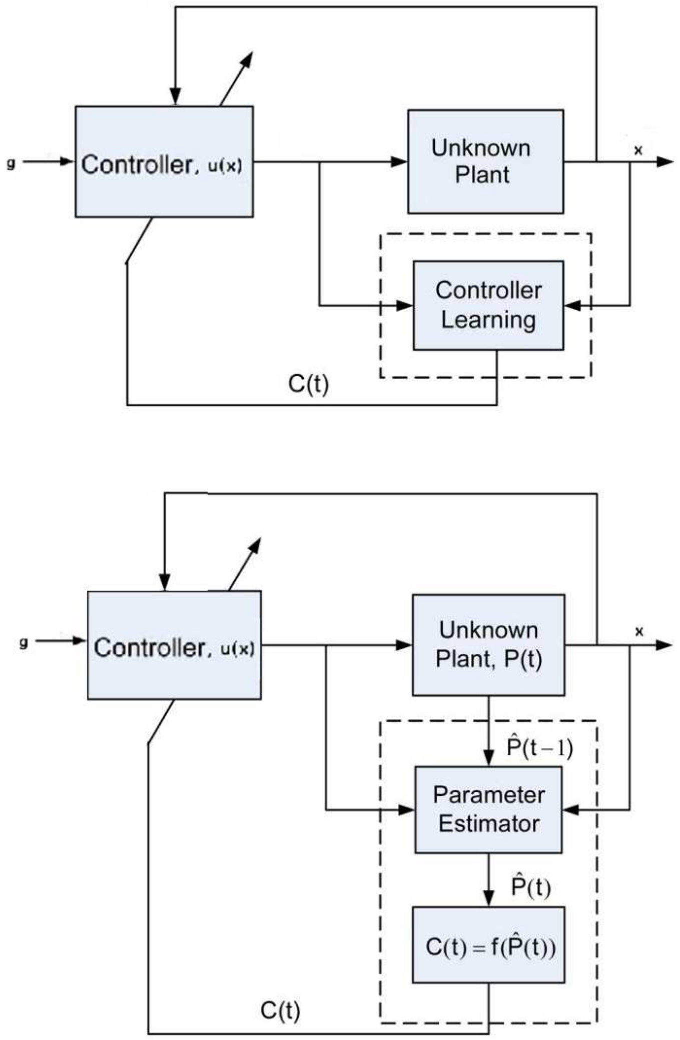


Figure 1.4: Top: direct adaptive control. The controller is adapted directly from the input/output signals. Bottom: indirect adaptive control. The controller is adapted by first estimating the plant parameters and/or state variables.

model may not always satisfy controllability and stability conditions which are requisite for controller design ([15]). Persistent excitation of the input signal is generally required for unbiased identification of the system parameters. However, identifiability could be lost under closed-loop conditions ([14]). For direct adaptive control, parametrization of the plant model into the controller parameters is not necessary ([15]). Also, stability and convergence can be guaranteed with proper design of the controller. However, designing an adaptive controller directly is a challenging task that is not always feasible. The following provides a few examples of direct and indirect adaptive control.

Self-tuning regulator. In classical feedback control, the controller design is determined as a function of the plant. When the plant model is unknown, the controller parameters are adapted using an (implicit or explicit) estimate of the plant model based on measurements of the instantaneous output and control signals. The estimation and adaptation procedures are performed online automatically. This construction is called self-tuning regulator (STR) to emphasize the automatic tuning of the controller. STR can be direct or indirect. In STR, learning is unsupervised, i.e., without the need for an explicit teacher or reference model.

High-gain adaptive control. High-gain adaptive control (HGAC) is the simplest STR in the form of direct adaptive control. In classical feedback control of an unstable plant, a proportional controller with a sufficiently high gain can stabilize the closed-loop system even when the plant is nonlinear, provided the plant satisfies certain strictly positive realness (SPR) condition [1]([16]). This is in contrast to non-SPR systems, where high gain may lead to instability especially when the feedback has significant delays. This idea is employed in HGAC with a continuously adapting controller gain K (Fig. 1.5). Here, stabilizing the plant output is equivalent to tracking a zero reference input, i.e., $r(t) = 0$. In this case the control and adaptation laws can be formulated, respectively, as follows:

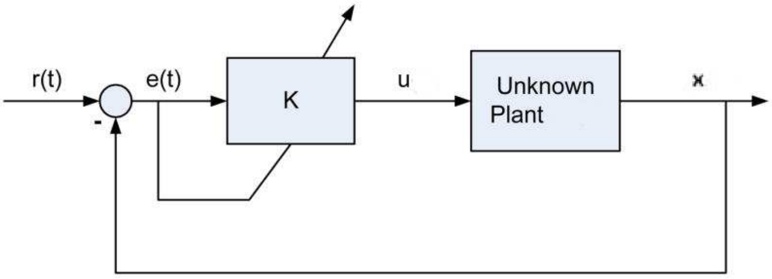


Figure 1.5: A proportional gain controller can stabilize an unstable plant by increasing the feedback gain, K, until the feedback error tends to zero, provided the SPR conditions are satisfied.

$$u = -Ke$$

$$\dot{K} = \alpha e^2$$

[1] A system is said to be SPR if its transfer function, $G(s)$, is analytical on $\mathrm{Re}(s) > 0$ and $G(jw) + G^*(jw) > 0$ for $w \in [0, \infty)$.

where K is the controller gain, $e(t) = r(t) - x(t)$ is the error signal, and α is the adaptation rate. It can be easily verified that $e(t)$ will converge to zero asymptotically as K increases to some large but finite value. Estimation of the plant parameters is not necessary. Indeed, convergence is guaranteed even when the plant input-output relationship is nonlinear (but bounded). However, convergence is restricted to the reference input $r(t) = 0$ only. For any non-zero and/or time-varying reference input, $e(t)$ generally does not converge to zero. Moreover, this design scheme is not robust to measurement errors ([17]).
Adaptive sliding control. Slotine and Li ([19]) have presented a STR algorithm (Fig. 1.6) which consists of a proportional-differential feedback loop and a full dynamics feedforward loop, with the unknown system parameters estimated online. By assuming a linearly parametrized dynamic model, the adaptive controller is proved with Lyapunov stability analysis to be globally stable with zero tracking errors. This is achieved by introducing a time-varying sliding surface, s, defined as,

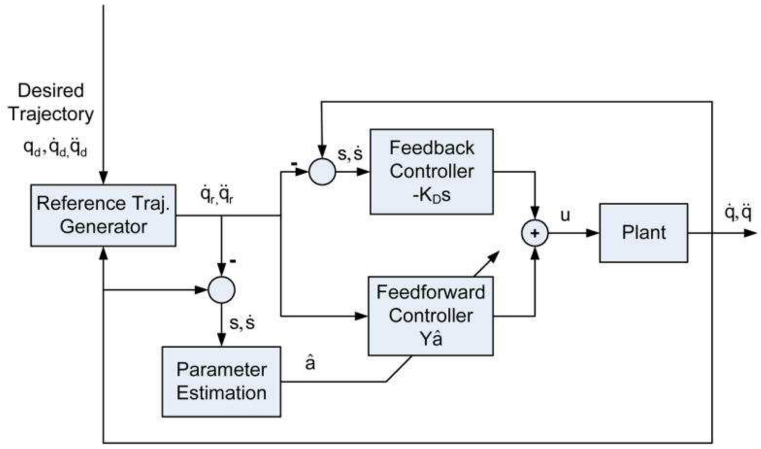


Figure 1.6: The algorithm comprises a fixed feedback and an adaptive feedforward loop. The sliding variable, s, drives the feedback controller and also adapts the feedforward controller. Unknown system parameters are estimated online. Convergence of tracking error and estimation error is guaranteed by Lyapunov analysis.

$$s = \dot{\tilde{q}}_r = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda \tilde{q},$$
$$\tilde{q} = q - q_d,$$

where $\tilde{q}$, $\dot{\tilde{q}}$ are the error signals in position and velocity, q_r is a reference trajectory acting as an intermediate variable to compute the sliding variable s, and Λ is a symmetric positive definite matrix. The control law is defined as,

$$u = Y\hat{a} + K_D s$$

where u is the control signal, $Y = Y(q, \dot{q}, \dot{q}_r, \ddot{q}_r)$ is generally a matrix function, $\hat{a}$ is a vector of the parameter estimates, $Y\hat{a}$ denotes the estimated inverse dynamics of the plant and serves as the feedforward controller, and K_D is the gain of the feedback controller. Lyapunov stability analysis then guarantees that the errors $(\tilde{q},\ \dot{\tilde{q}})$ converge asymptotically to the sliding surface $s = 0$, which corresponds to zero tracking errors in both velocity and position.
The sliding variable describes the dynamics of a reference model. Adaptation is achieved by adjusting the parameters in the feedforward control loop, which identifies a coarse model for the inverse dynamics of the plant. The algorithm provides a computationally simple method for designing an STR controller that is stable and robust with fast convergence even for nonlinear plant dynamics.
Adaptive sliding control has important implications in providing a rigorous engineering basis for the feedback error learning paradigm (Fig. 1.7) proposed by Kawato et al. ([20]) for inverse internal model adaptation during motor learning (see section 2.3.5).

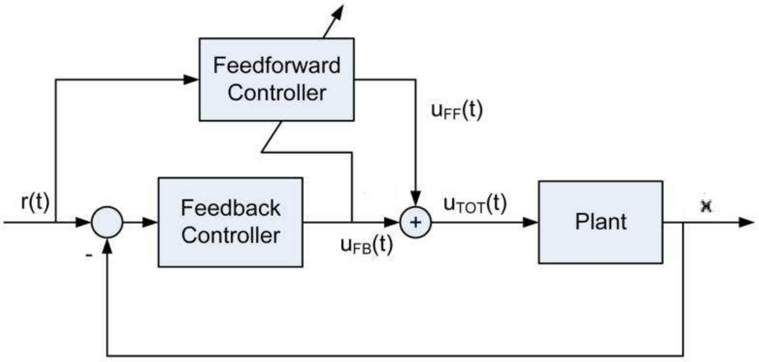


Figure 1.7: The algorithm is similar to adaptive sliding control except the output of the feedback controller is used to adapt the feedforward controller which forms the inverse internal model of the system.

In these classical schemes the state of the system and the used control are completely available both to the feedback controller and to the learning controller, with no delays. In practical situations, however, there can be limitation to the amount of information which can be transmitted through the different data-lines. Limited information implies digital controller, that is the need to sample and quantize measures on the state and controls. This is particularly true in the communication between the plant and the central controller. Also in the case of non adaptive control, very often the plant, along with its sensors and actuators, will not have an on-site controller/learner, which is localized physically far from it. Thus, the needing to implement codification (i.e sampling and quantization) and information transmission algorithms which efficiently close the feedback control loop through some kind of network. This implies long delays (possibly of the same order of the time scale of the controlled dynamics and non constant) and measurement errors due to the codification. As we saw, the high-gain adaptive control, for example, is not robust to these circumstances, while adaptive sliding controls require for their same structure that there are no large delays. For example, the feed forward control in the feedback error learning scheme must be almost synchronized with the feedback error, in order to correctly derive on-line parameters estimation.

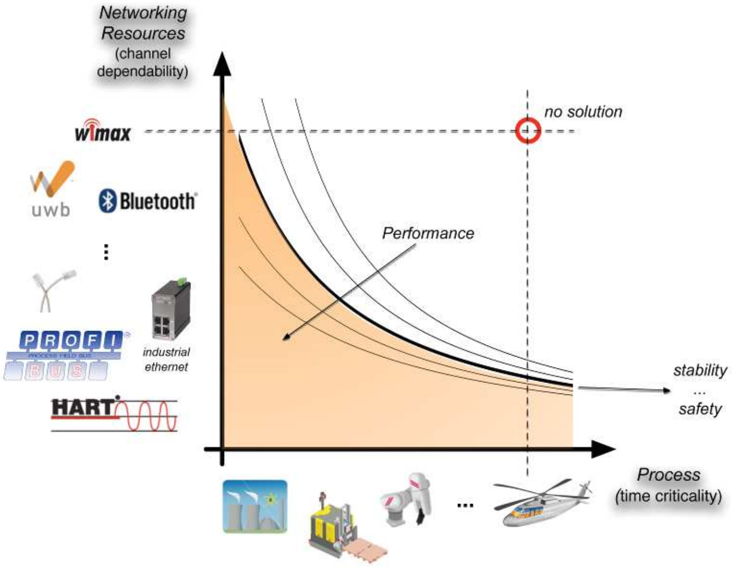


Figure 1.8: A figurative illustration of the traditional view of networked control systems design

In the traditional view of networked control, sketched in fig.1 below, the design is done in a two dimensional conceptual space. On one axis the physical processes to be controlled are considered; on the other axis, resources made available by the network are represented. A loose ordering is alluded to in the figure, whereby systems of increasing time criticality are reported rightmost, while networking technologies are arranged according to their dependability, in terms of timing reliability, predictability, availability etc. At the same intuitive level of description, requirements in this space can be visualized as iso-performance (i.e stability, safety, etc...) curves. As an example, in the control of a highly time-critical, open-loop unstable plant, even the most basic requirement of stabilization can not be robustly achieved in the presence of low network performances, i.e long delays, information looses, etc...

In the wider view of an Embedded Control Design space, depicted in fig. 2, a third axis is added to the picture, representing the amount of intelligence (made of sensing, computational, actuation resources) that the designer can embed in the

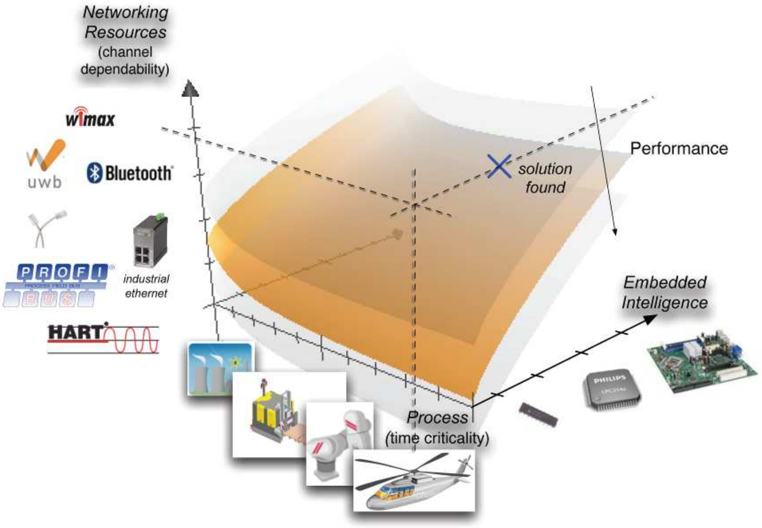


Figure 1.9: The Networked Embedded Control Design space

system. The design goal is hence to evaluate what is the exact amount of decentralized control capabilities to be embedded in the subsystem, that are needed to guarantee fulfillment of specified requirements (represented here as iso-performance manifolds). In the examples above, this amounts to establishing the simplest local controller, or the simplest set of distributed collision avoidance rules, that can guarantee safe execution of higher-level plans received from the network.

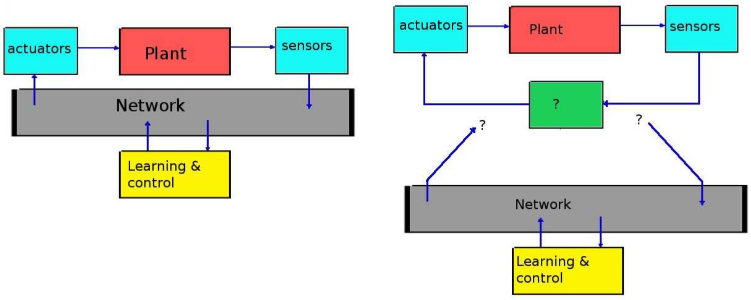


Figure 1.10: Left: classical networked control; Right: embedded-networked control design

Figure 1.10 shows classical networked control (left) and embedded-networked control design. Question marks allude to open possibilities about the kind of embedded control (green square) and the protocol/contents of the information lines to and from the central controller.

A possible solution is to synthesize an embedded control law robust to param-

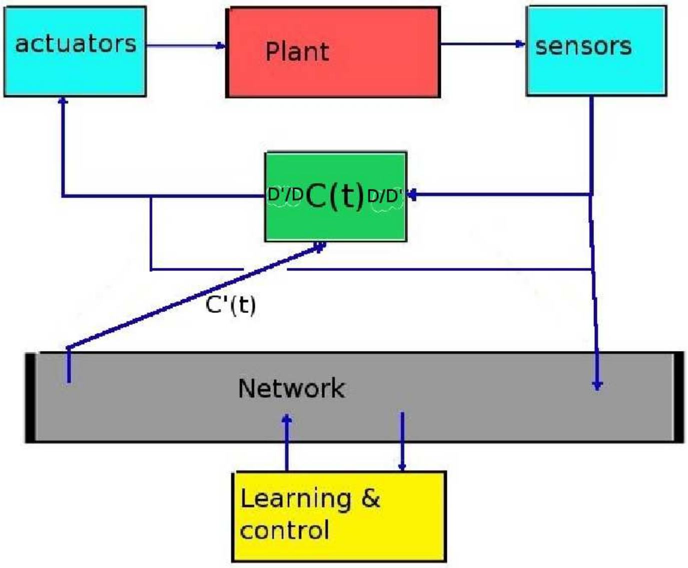


Figure 1.11: Possible embedded-networked control design with minimal data rate embedded controller. The D/D' and D'/D symbols represent the codification from the high bit rate signal sent through the network to the minimal bit rate input of the embedded controller and the decodification from the codified output of the embedded controller to the high bit rate input of actuators, respectively. $C(t)$ is the present set of control parameters, while $C'(t)$ is the upgrade due to the learning controller.

eters uncertainties and measurement errors, so that it is possible to use a digital controller that use as little as computational and observational power as possible, i.e minimal bit rate, (we will analyze and synthesize such feedback controller in subsection 3.2.5), while the full state (i.e high bit rate) of the system and relative control are passed, with very long delays due to the big amount of information, to the learning apparatus, which will then upgrade the feedback parameters (fig. 1.11) by learning the inverse dynamic of the system. While being very convenient for the low costs required by the embedded controller, this solution is not very clever from the point of view of information transmission between the plant and the central controller, as one has to constantly send big amount of information to make learning possible.

In the following chapter we will give a survey of known results about movement learning in the vertebrates and propose a similar inspired adaptive controller.

Chapter 2

Movement learning modeling

Section 2.1-2.3 are based on materials from ([58]).

2.1 The problem of inverse dynamics

When we learn to move our limb and act upon the environment, our brain becomes to all effect an expert in physics. To illustrate the complexities of ordinary motor behaviors, let us consider the task that the central nervous system (CNS) must solve every time a planned gesture is transformed into an action. If the goal is to move the hand from an initial position to another point in space, then clearly there are a number of possible hand trajectories that could achieve this goal: the solution of this elementary problem is not unique. Even after the CNS has chosen a particular path for the hand, its implementation can be achieved through multiple combinations of joint motions at the shoulder, elbow and wrist- again the solution is not unique. Finally, because there are many muscles around each joint, the net force generated by their activation can be produced by a variety of combinations of muscles.

According to the laws of the Newtonian physics, if we want to impress a motion upon a stone with mass m, we must apply a force F, that is directly proportional to desired acceleration, a. This is the essence of Newton's equation $F = ma$. A desired motion may be expressed as a sequence of positions, x, that we wish the stone to assume at subsequent instant of time. t. This sequence is called a 'trajectory' and is mathematically represented as a function, $x = x(t)$. To use Newton's equation for deriving the needed time sequence of forces, we must calculate the first derivative of the trajectory, the velocity and then the second temporal derivative, the acceleration. The above calculation is an example of what in robotics is called an *inverse dynamic problem.* The 'direct dynamic problem is that of computing the trajectory resulting from the application of forces, $F =$

$F(t)$. The solution of this problem requires a complex computational process called integration.

Direct problems are the bread and butter of physicists, who may be concerned, for example, with predicting the motion of a comet from the known pattern of gravitational forces. Unlike physicists, the brain deals most often with inverse problems: we routinely recognize object and people from their visual images -an inverse optical problem- and we find effortlessly how to distribute the forces carried out by several muscles to move our limb in the desired way -an inverse dynamics model.

In the biological context, the inverse dynamic problem assumes somewhat different form from the case of the moving stone. One of the central questions in motor control is how the CNS may form motor commands that guide our limb. One proposal is that the CNS solves an inverse dynamic problem. A system of second order nonlinear differential equations is generally considered to be an adequate description of passive limb dynamics. A compact representation of such system is:

$$D(q, \dot{q}, \ddot{q}) = \tau(t), \tag{2.1}$$

where q,$\dot{q}$,$\ddot{q}$ represent the limb configuration vector -for example, the vector of configuration angles- and its first and second time derivatives. The term $\tau(t)$ is a vector of generalized forces, for example, joint torques, at time t. Conceptually, this expression is nothing else than Newton's law applied to a multi-articular rigid body. In practice the expression for D may have a few terms for a two-joint planar arm or it may take several pages for more realistic models of arm's multi joint geometry. The inverse dynamic approach to the control of multi joint limbs consist in solving explicitly for a torque trajectory, $\tau(t)$, given a desired trajectory of the limb, $q_D(t)$. This is done by plugging $q_D(t)$ on the left side of equation (2.1):

$$\tau(t) = D(q_D(t), \dot{q}_D(t), \ddot{q}_D(t)). \tag{2.2}$$

Another significant computational challenge comes from the need to perform changes of representation, or, more technically, 'coordinate transformation, between the description of a task and the specification of the body motion. Tasks, such as 'bring the hand to the glass of water on the table', are often described most efficiently and parsimoniously with respect to fixed reference point in the environment. For example, the glass may be 10cm to the left of a corner of the table. The hand may be 20cm to the right of the same corner. So, the hand will need to be displaced 30cm along a straight line in the left direction. This is a very simple description of the needed movement. However, this description cannot be used to derive the joint torques, as specified by equation (2.2). To this end, one must represent the trajectory of the hand in terms of the corresponding angular motion of each joint. This is a complex transformation known in robotics as 'inverse kinematic'.

Does the brain carry out similar inverse dynamic calculations for moving the arm on a desired trajectory? A clear-cut answer is still to come but several alternatives have emerged from studies in robotics and computational neuroscience.

2.2 Solution based on feedback

Many of the problems that the brain must face to control movements are indeed similar to those that engineers must solve to control robots. In spite of great differences between the multi joint vertebrate system and current robotic arms, the field of neuroscience, unquestionably, has derived benefits from the theories and procedures that have guided the construction of man-made limbs. For instance, from early on, neuroscientists have been influenced by the notion of feedback. Feedback control is a way to circumvent the computation of inverse dynamics. At each point in time, some sensory signal provides the information about the actual position of the limb. This position is compared with a desired position and the difference between the two is a measure of the error at any given time. Then, a force may be produced with amplitude approximately proportional to the amplitude of the error in the direction of the desired position. This method of control is appealing because of its great simplicity.

Multiple feedback mechanisms have been found in both vertebrates and invertebrates. These mechanisms were discovered by Sherrington at the beginning of the last century ([21]). They have been shown to control the muscles level of contraction, the production of force and the position of joints. Sherrington observed that when a muscle is stretched the stretch is countered by an increase in muscle activation. This stretch reflex is caused by sensory activity that originates in the muscle spindles-receptors embedded within the muscle fibres. Sherrington put forward the daring hypothesis that complex movements may be obtained by combining stretch reflexes as well as other reflexes in a continuous sequence or chain. In this way, movement patterns as complex as the locomotion cycle could be generated by local reflexes, without central supervision. A similar idea was later proposed by Merton ([22]), who suggested that central commands via the gamma muscle spindle system might initiate the execution of movement, not by directly activating the muscles, but by triggering a stretch reflex through the modulation of muscle spindle activities. Both Sherrington and Mertons hypotheses are attempts at explaining movements as automatic responses to sensory feedback, thus limiting the role and the arbitrariness of voluntary commands. However, both Sherringtons ideas on compounding of reflexes and Mertons hypothesis have taken a new form following subsequent experiments which clearly demonstrated the generation of movements in the absence of sensory activities. For example, Taub & Berman ([23]) found that monkeys can execute various limb movements after the surgi-

cal section of the pathways that convey all sensory information from the limb to the nervous system. Shortly thereafter, Vallbo ([24]) was able to record muscle spindle discharges in human subjects and to compare these discharges with the activation of the muscles, as revealed by electromyography (EMG). Vallbos study showed that, in a voluntary movement, muscle activation does not lag but leads the spindle discharges, contrary to the predictions of Mertons hypothesis.
In addition to the experimental findings described above, the idea that biological movements may be carried out by feedback mechanism has been challenged based on consideration about limb stability and reflex delays. It takes more than 40ms before a signal generated by the muscle spindles may reach the supra spinal motor centers and it takes 40-60 ms more before a motor command may be transformed into a measurable contraction of the muscles. These transmission delays may cause instability (Hogan et al. 1987). The effects of delays are even greater when the limb interacts with the environment. For example, if a robotic arm were to contact a rigid surface, a delay of 3Oms would initiate a bouncing motion also known as chattering instability. This instability is again due to the fact that the control system could detect the contact only after it has occurred. This would cause a back-up motion that would move the arm away from the surface. Then, the controller would move again towards the surface and so on in a repeated bouncing motion.

2.3 Solution based on feedforward

An alternative to feedback control would be for the CNS to pre-program the torques that the muscles must generate for moving the limbs along the desired trajectories. This method is often referred to as feed-forward control. The torques needed to move the arm can only be computed after the angular motions of the shoulder, elbow and wrist have been derived from the desired movement of the hand-that is after an inverse kinematics problem has been solved. Investigations in robot control in the late 1970s and early 1980s showed that both the inverse kinematic and inverse dynamic problems may be efficiently implemented in a digital computer for many robot geometries ([25]). On the basis of these studies, Hollerbach & Flash ([26]) put forward the hypothesis that the brain may be carrying out inverse kinematic and dynamic computations when moving the arm in a purposeful way. Their experimental investigation of arm-reaching movements, combined with inverse dynamics calculations, showed that all components of the joint torque played a critical role in the generation of the observed hand trajectories. In particular, Hollerbach & Flash found that while executing reaching movements the subjects were accurately compensating for the dynamic interactions between shoulder and elbow joints.

Evidence that the brain is carefully compensating for the interaction torques was further provided by more recent studies of Ghez and of Thach and their co-workers. Sainhurg et al. ([27]) studied the movements of subjects suffering from a rare peripheral neuropathy. A consequence of this disease is the complete loss of proprioceptive information from the upper and lower limbs. These investigators found that the abnormal motions observed in these subjects could be accounted for by lack of compensation for the joint interaction torques. A similar conclusion was reached later by Bastian et al. ([28]) about the movements produced by patients suffering from cerebellar lesions. In summary, a substantial body of evidence suggests that the CNS generates motor commands that effectively represent the complex dynamics of multi joint limbs. However, there are different ways for achieving this representation.

2.3.1 Memory based computation

A rather direct way for a robot to compute inverse dynamics is based on carrying out explicitly the algebraic operations after representing variables such as positions, velocity , acceleration, torque and inertia. Something similar to this approach had been first proposed by Raibert ([29]). He started from the observation that inverse dynamic can be represented as the operation of a memory that associates a vector of joint torques to each value of joint angles, angular velocities and angular accelerations. A brute-force approach to dynamics would simply be to store a value of torque for each possible value of position, velocity and acceleration-a computational device that computer scientists call a look-up table. This approach is extremely simple and in fact look-up tables were implicit in early models of motor learning, such as those proposed by Albus ([30]) and Marr ([31]). However, a closer look at the demands for memory size in a reasonable biological context shows that the look-up table approach may be impracticable.

The number of entries in a look-up table grows exponentially with the number of independent components that define each table entry. Being well aware of this problem, Raibert suggested splitting the arm dynamics computations in a combination of smaller subtables: one can obtain the net torque by adding (i) a term that depends on the joint angles and on the angular accelerations to (ii) a term that depends on the joint angles and on the angular velocities. These two terms may be stored in separate tables. Assuming a resolution of only ten values per variable, the control of a two joint limb would require two tables with 10^4 entries each. For a more complete arm model, with seven-joint coordinates, each table would have 10^{14} entries. These are still exceedingly large numbers. A method for reducing the size of look-up tables was suggested by Raibert & Horn ([32]), who represented the dynamic problem as a sum of three elements, each one requiring a table that depended only on the joint angles. Thus, the two-joint limb involved tables with 100 entries and the seven-joint limb tables with 10^7 entries.

2.3.2 The equilibrium point hypothesis

The work of Raibert & Horn ([32]) showed that inverse dynamics of complex limbs may be computed with a reasonable number of operations and with reasonable memory requirements. However, this work did not provide any direct evidence that the brain is ever engaged in such computations. Furthermore, on a purely theoretical basis, explanations based on computing inverse dynamics are unsatisfactory because there is no allowance for the inevitable mechanical vagaries associated with any interaction with the environment. For instance, if an external force perturbs the trajectory of the arm, dramatic consequences may follow. When we pick up a glass of water, we must update the pattern of torques that our muscles must apply to generate a movement of the arm. When we open a door, we must deal with a constraint, the hinge, whose location in space is only approximately known. One may say that most of our actions are executed upon a poorly predictable mechanical environment. It would then be erroneous to suggest that a stored pattern of neuromuscular activations corresponds to some particular movement. Instead, the movement that arise from that pattern is determined by the interaction of the muscle forces with the dynamics of the environment.

Hogan ([33]) developed this concept in a theory known as impedance control. Hogans ideas relate to earlier experiments of Feldman ([34]) and Bizzi and coworkers. In one of these experiments, Polit & Bizzi ([35]) trained monkeys to execute movements of the forearm towards a visual target. The monkey could not see their moving arm nor could they perceive it as their proprioceptive inflow had been surgically interrupted by the transection of cranial and thoracic dorsal roots- a procedure called deafferentation. Surprisingly, Polit & Bizzi found that, despite such radical deprivation of sensory information, the monkeys could successfully reach the visual targets. What was more unexpected was that the monkeys could reach the intended target even when their arm had been displaced from the initial location just prior to the initiation of an arm movement. This result did not seem to be compatible either with the idea that goal-directed movements are executed by a pre-programmed sequence of joint torques or with the hypothesis that sensory feedback is essential to reach the desired limb position.

The performance of the deafferented monkey can be accounted for by the hypothesis that the centrally generated motor commands modulate the stiffness and rest length of muscles that act as flexors and extensors about the elbow joint. As a consequence, the elastic behavior of the muscles, like that of an opposing spring, defines a single equilibrium position of the forearm. A position that ultimately is reached in spite of externally applied perturbations, without need for feedback corrections. This result led to a question concerning the execution of target- directed movements. Are these movements executed ,just by setting the equilibrium point of a limb to the final target? Or does the descending motor command specify an entire trajectory as a smooth shift of the same equilibrium point? Bizzi et al. ([36]) addressed this question in another experiment. If, as suggested by the first hypothesis, there is a sudden jump of the limbs equilibrium to the target location, an elastic force driving the hand toward the target would appear from the onset of the movement. This force would be directed all the time towards the target. The experiment of Bizzi and co- workers disproved this hypothesis. As in the work of [35], they instructed deafferented monkey to execute arm movements towards a visual target but with the vision of the arm blocked by an opaque screen. As soon as the EMG activity indicated the onset of a movement, a motor drove the arm right on the target. If this were the equilibrium position specified by the muscle commands at that time, the arm should have remained in place. On the contrary, the experimenters could observe an evident motion backward towards the starting position followed by a forward motion toward the target. This finding indicate that the muscular activation does not specify a force or a torque, as suggested by the inverse dynamic models, nor a final target position. Instead, the response to the initial displacement suggests that the activation of the muscles produces a gradual shift of the limbs equilibrium from the start to end location. Accordingly, *at all times the limb is attracted by an elastic force towards the instantaneous equilibrium point.* If during a goal-directed movement, the limb is forcefully moved ahead towards the target, the elastic force will drive it towards the lagging equilibrium point, as observed in the experiment.

The sequence of equilibrium positions produced during movement by all the muscular activations has been called by Hogan ([37]) a virtual trajectory. The virtual trajectory is a sequence of points where the elastic forces generated by all the muscles cancel each other. By contrast, the actual trajectory is the result of the interaction of these elastic forces with other dynamic components such as limb inertia, muscle velocity-tension properties and joint viscosity. To intuitively illustrate this distinction, consider a ball attached to a rubber band. When the hand is displaced from its equilibrium position, a restoring force is generated with amplitude proportional to the displacement. If we move the free end of the rubber band, we control the equilibrium position. As we move the rubber band along a trajectory the ball will follow a trajectory that results from the interaction of the elastic force with the mass of the ball.

The idea of a virtual trajectory provides a new unified perspective for dealing with (i) the mechanics of muscles, (ii) the stability of movement, and (iii) the solution of the inverse dynamic problem. In fact, *a strictly necessary and sufficient condition for a virtual trajectory to exist is that the motor commands directed to the muscle define a sequence of stable equilibrium positions.* If this requirement is met, then there exists a single well-defined transformation from the high-dimensional representation of the control signal as a collection of muscle activations, to a low-dimensional sequence of equilibrium points. An advantage of this low-dimensional representation is that, unlike muscle activations, the virtual trajectory may be directly compared with the actual movement of the limb.

The relationship between actual and virtual trajectory is determined by the dynamics of the system and by the stiffness, which transform a displacement from the equilibrium into a restoring force. In the limit of infinite stiffness, the actual trajectory would match exactly the virtual trajectory. On the other hand, with low stiffness values, the difference between virtual and actual trajectory may become quite large. In a work that combined observations of hand movements and computer simulations, Flash ([38]) tested the hypothesis that multi joint arm movements are obtained by the CNS shifting the equilibrium position of the hand along a straight and rectilinear motion from the start to end position. As shown by Morasso ([39]), approximately straight hand paths characterize planar hand movements between pairs of targets. If the same movements are analyzed at a finer level of detail, however, the paths present certain degrees of inflection and curvature, depending on the direction of movement and on the work-space location. In the simulations Flash made the assumption that the hand equilibrium trajectories (but not necessarily the actual trajectories) are invariably straight. In addition, she assumed that the equilibrium trajectory had a unimodal velocity profile. The results obtained from the simulation captured the subtle inflections and the curvatures of the actual trajectories. Moreover, the direction of curvature in different work-space locations and with different movement directions matched quite closely the observed movements.

It must be stressed that the stiffness values used in this simulation were taken from measurements that had been performed not during movements but while subjects were maintaining their arm at rest in different locations ([40]). Gomi & Kawato ([41]) repeated Flashs simulation using lower values of stiffness and found, not surprisingly, that in order to reproduce the actual trajectory of the hand, the virtual trajectory had to follow a much more complicated pathway. The results obtained by Gomi & Kawato are at variance with those of Won & Hogan ([42]), who were able to show that for relatively slow and low-amplitude arm trajectories the virtual equilibrium point was close to the actual trajectory. Clearly, the complexity of the virtual trajectory depends critically upon the elastic field surrounding the equilibrium point. Experimental estimates of the elastic field under static conditions have shown that the local stiffness, i.e. the ratio of force and displacement, changes at different distances from the equilibrium point ([43]). Specifically, it was found that the stiffness decreased with this distance. This is a nonlinear feature of the elastic field. Accordingly if, as in ([41]), one attempted to derive the equilibrium point

using a linear estimate based on the stiffness at the current position, one would overestimate the distance between current and equilibrium position. At present, however, there is not yet an acceptable technique for measuring the elastic force field generated by the muscles during movement. But, if the shape of the virtual trajectory is a complex path, as in Gomi & Kawatos simulations, then the apparent computational simplicity of the earlier formulation of the equilibrium-point hypothesis is lost.
Another challenge to the equilibrium-point hypothesis comes from the work of Lackner & Dizio ([44]) who asked subjects to execute reaching hand movements while sitting at the center of a slowly rotating room. Because of this rotation, a Coriolis force proportional to the speed of the hand perturbs the subjects arm. The Coriolis force acts perpendicularly to the direction of motion. Lackner & Dizio found that, under this condition, there is a systematic residual error at the final position in the direction of the Coriolis force. This finding seems incompatible with the equilibrium-point hypothesis because the Coriolis force depends upon hand velocity but not upon hand position. Therefore, it should not alter the location of the final equilibrium point. However, the experimental results of Lackner & Dizio are in apparent contrast with other experimental findings obtained with similar force fields. In particular, Shadmehr & Mussa-Ivaldi ([45]) used an instrumented manipulandum for applying a velocity-dependent field to the hand of the subjects. In this paradigm the perturbation acted specifically on the arm dynamics and did not affect in any way other systems, such as the vestibular apparatus. Shadmehr & Mussa-Ivaldi found that the final position of the movement was not substantially affected by the presence of velocity-dependent fields, in full agreement with the equilibrium-point hypothesis. The cause of the discrepancy between these results and those of Lackner & Dizio ([44]) has yet to be determined.

2.3.3 The building blocks for computation of dynamics: spinal force fields

Recent electrophysiological studies of the spinal cord of frogs and rats by Bizzi and co-workers ([46]; [47];[48]) suggest a new theoretical framework that combines some features of inverse dynamic computations with the equilibrium-point hypothesis. In these studies, electrical stimulation of the interneuronal circuitry in the lumbar spinal cord of frogs and rats has been shown to impost a specific balance of muscle activation. The evoked synergistic contractions generate forces that direct the hindlimb towards an equilibrium point in space (figure 2.1).

To measure the mechanical responses of the activated muscles, Bizzi et al. ([46])and Giszter et a1. ([47]) attached the right ankle of the frog to a force transducer. To record the spatial variations of the force vectors generated by the

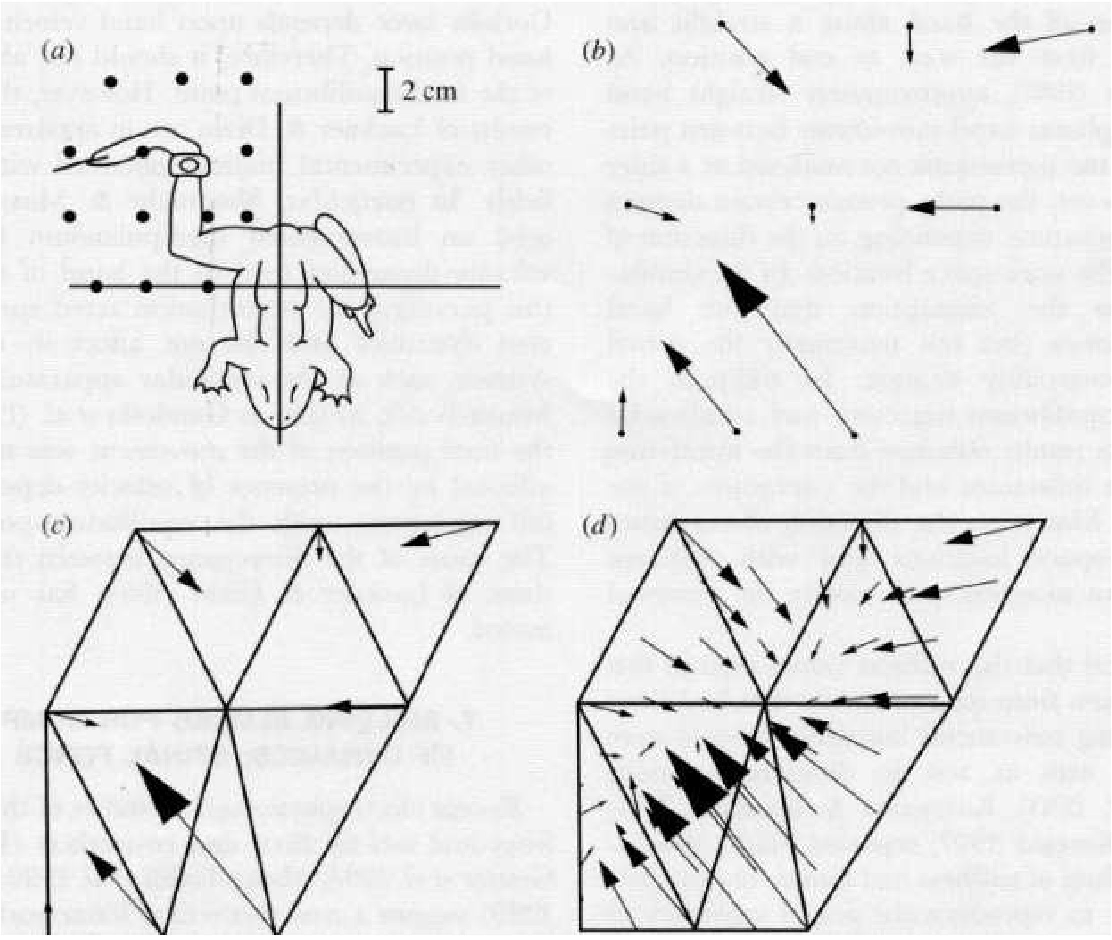


Figure 2.1: Force fields induced by microstimulation of the spinal cord in spinalized frogs. (a) The hindlimb was placed at a number of locations on the horizontal plane (indicated by the dots). At each location a stimulus was derived at a fixed site in lumbar spinal cord. The ensuing force was measured by a six-axes force transducer. (b) Peak force vectors recorder at the nine locations shown in (a). (c) The work-space of the hindlimb was partitioned into a set of non-overlapping triangles. Each vertex is a tested point. The force vectors on the three vertices are used to estimate, by linear interpolation the force in the interior of the triangle. (d) Interpolated force fields.

leg muscles, they placed the frogs leg at a location within the legs work-space. Then, they stimulated a site in the spinal cord and recorded the direction and amplitude of the elicited isometric force at the ankle. This stimulation procedure was repeated with the ankle placed at each of nine to 16 locations spanning a large portion of the legs work-space. The collection of the measured forces corresponded to a well-structured spatial pattern, called a vector field. In most instances, the spatial variation of the measured force vectors resulted in a field that was at all times both convergent and characterized by a single equilibrium point.

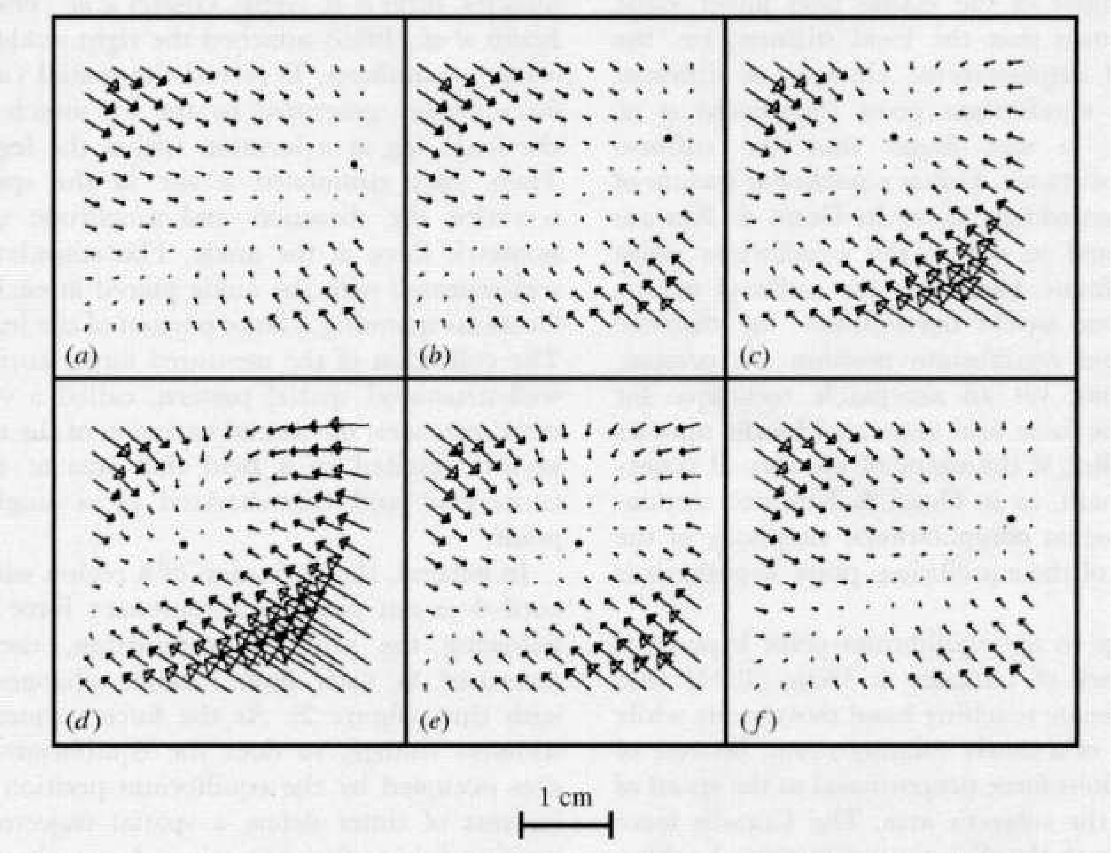


Figure 2.2: Temporal evolution of a spinal force field. Following the stimulation of a site in the spinal cord, the force vectors change in a continuous fashion. The result is a mechanical wave, described here by a sequence of frames ordered by increasing latency from the onset of the stimulus. The frames are separated by intervals of 86 ms. The dot indicates the location of the static equilibrium point (where the estimated force vector vanishes) in each frame.

In general, the activation of a region within the spinal cord does not produce a stationary force field. Instead, following the onset of stimulation, the force vector measured at each limb location changes continuously with time (figure 2.2). As the force vectors elicited by a stimulus change, so does the equilibrium position: the sites occupied by the equilibrium position at subsequent instants of times define a spatial trajectory. The time- varying field is the expression of a mechanical wave that summarizes the combined action of the muscles that are affected by the stimulation. Mechanical waves of the same kind can be used to describe the operation of central pattern generators and of other natural structures involved in the control of motor behavior. At all latencies after the onset of stimulation, the force field converges towards an equilibrium position. The temporal sequence of these equilibrium positions provides us with an image of a virtual trajectory, as in the sequence of frames of figure 2. Sometimes we found that the virtual trajectories observed after electrical stimulation followed circular pathways starting and ending at the same point ([49]). In contrast, the virtual trajectories inferred by Flash ([38]) from reaching arm movements followed rectilinear and smooth pathways, from start to final position of the hand. This is not a surprising discrepancy given the great difference in experimental conditions, limb mechanics and neural structures involved in these studies. Despite these differences, however, it is remarkable that the essential biomechanics of the moving limb is the same for the hindlimb of the spinalized frog and for the arm of the human subject. In both cases, *movement is described as a smooth temporal evolution of a convergent force field produced by the spring-like properties of the neuromuscular apparatus.*

Perhaps the most interesting aspect of the investigation of the spinal cord infrogs and rats was the discovery that the fields induced by the focal activation of the cord follow a principle of vectorial summation (figure 2.3). Specifically, Mussa-Ivaldi el al. ([50]) developed an experimental paradigm involving the simultaneous stimulation of two distinct sites in the frogs spinal cord. They found that the simultaneous stimulation of two sites led to vector summation at the ankle of the forces generated by each site separately. When the pattern of forces recorded at the ankle following co-stimulation were compared with those computed by summation of the two individual fields, Mussa-Ivaldi et al. ([50]) found that co-stimulation fields and summation fields were equivalent in more than 87% of cases. Similar results have been obtained by Tresch & Bizzi ([48]) by stimulating the spinal cord of the rat. Recently, Kargo & Giszter ([51]) showed that force field summation underlies the control of limb trajectories in the frog.
Vector summation of force fields implies that the complex nonlinearity that characterizes the interactions both among neurons and between neurons and muscles is in some way eliminated. More importantly, this result has led to a novel hypothesis for explaining movement and posture based on combinations of a few basic elements. The few active force fields stored in the spinal cord may be viewed as representing motor primitives from which, through superposition, a vast number of movements can be fashioned by impulses conveyed by supra spinal pathways. Through computational analysis, Mussa-Ivaldi & Giszter ([52]) and Mussa-Ivaldi ([53]) verified that this view of the generation of movement and posture has the competence required for controlling a wide repertoire of motor behaviors.
The fields generated by focal activation of the spinal chord are nonlinear functions of limb position, velocity and time: $\phi_i(q, \dot{q}, t)$, where the velocity dependence describes viscous property of the muscle-skeletal system. Consistent with the observation that these fields add vectorially, one may modify the formulation of the inverse dynamic problem by replacing the generic torque function, $\tau(t)$, with a

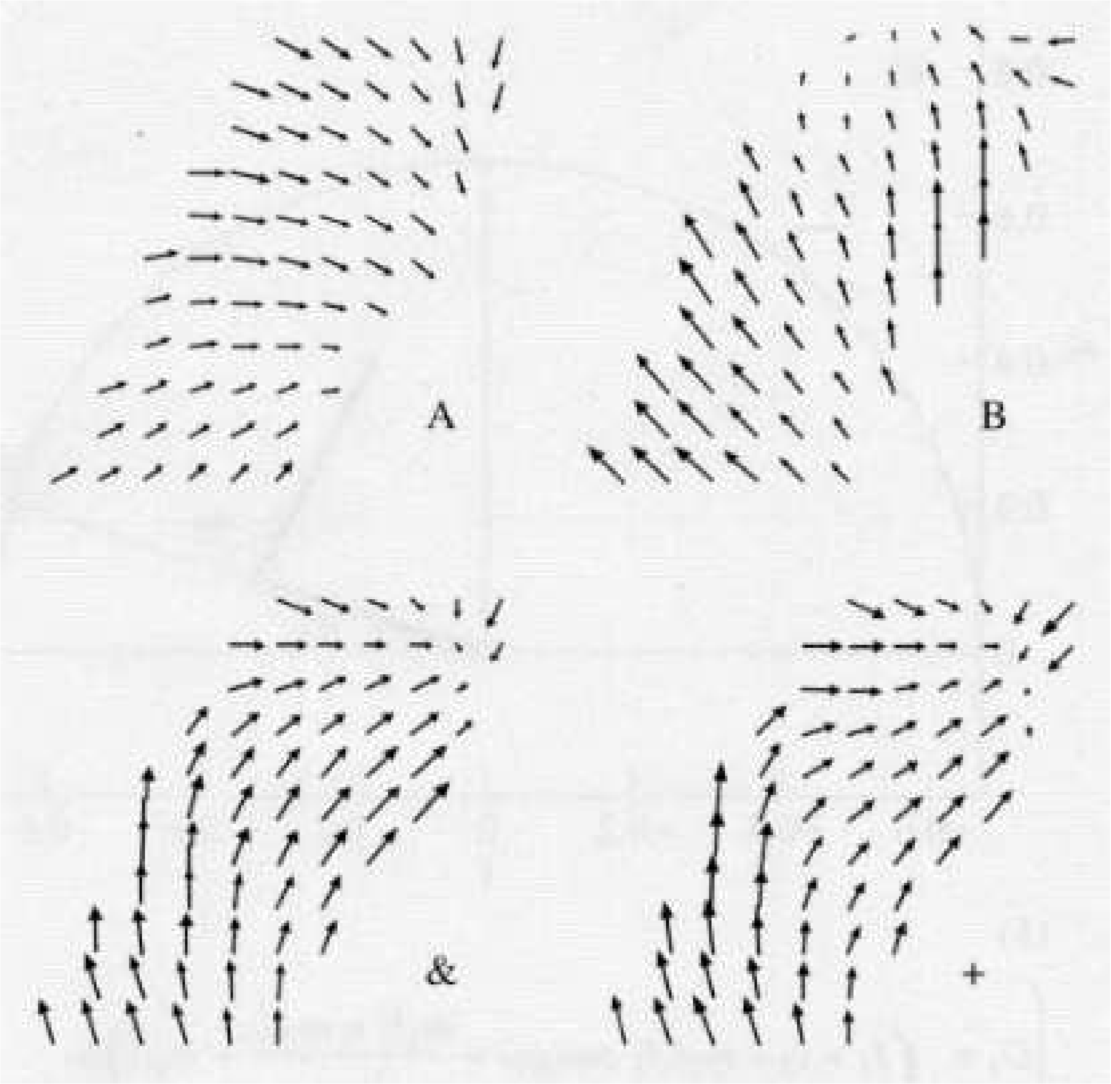


Figure 2.3: Spinal force fields add vectorially. Fields A and B were obtained in response to stimulations delivered to two different spinal sites. The & field was obtained by stimulating simultaneously the same two sites. It matches closely (correlation coefficient larger than 0.9) the force + force field, which was derived by adding pairwise the vectors in A and B. This highly linear behavior was found to apply to more than 87% of dual stimulation experiments (from [50])

superposition of spinal fields:

$$D(q, \dot{q}, \ddot{q}) = \sum_{i=1}^{K} c_i \phi_i(q, \dot{q}, t). \tag{2.3}$$

Here each spinal fields is tuned by a (non-negative) scalar coefficient, c_i that represent a descending supraspinal command. We should stress that in this model,

the descending commands do not alter the shape of the fields- that is their dependence upon state and time. This is consistent with the empirical observation that the pattern of force orientation of spinal fields remained invariant in time and with different intensities of stimulation ([47]). Thus, it is plausible to assume that the supraspinal signals select the spinal fields by determining how much each one contributes to the total field. The computational model of equation (2.3) is simply a reformulation of inverse dynamics, with the additional constraint that joint torque is produced by the modulation of a set of pre-defined primitives, the fields $\phi_i(q, \dot{q}, t)$.
The vector fields generated by the spinal cord offer a clear example of the impedance control that has been discussed in section 2.3.2. The experiments suggest that the circuitry in the spinal cord-and perhaps also in other areas of the nervous system-is organized in independent units, or modules. While each module generates a specific field, more complex behaviors may be easily produced by superposition of the fields generated by concurrently active modules. Thus, we may regard these force fields as independent elements of a representation of dynamics, what are commonly referred to as central pattern generators, circuitries capable of generating motor patterns without sensory input.

2.3.4 Evidence for internal model

The findings on the spinal cord suggest that the CNS is capable of representing the dynamic properties of the limbs as in a classical *internal model* scheme. The term 'internal model' refers to two distinct mathematical transformations: (i) the transformation from a motor command to the consequent behavior, and (ii) the transformation from a desired behavior to the corresponding motor command ([54];[55];[56]). A model of the first kind is called a 'forward model'. Forward models provide the controller with the means not only to predict the expected outcome of a command, but also to estimate the current state in presence of feedback delays. A representation of the mapping from planned actions to motor commands is called an 'inverse model'. Studies by Wolpert et al ([57]) proposed that the neural structures within the cerebellum perform sensory-motor operations equivalent to a combination of multiple forward and inverse models. Strong experimental evidence for the biological and behavioral relevance of internal models has been offered by the experimental results obtained by Shadmehr & Mussa-Ivaldi ([45]), which clearly demonstrate the formation of internal models. Their experimental subjects were asked to make reaching movements in the presence of externally imposed forces. These forces were produced by a robot whose free end-point was held as a pointer by the subject (figure 2.4). The subjects were asked to execute reaching movements towards a number of visual targets. Since the force field produced by the robot significantly changed the dynamics of the reaching movements, the subjects' movements initially were grossly distorted when compared with the undisturbed movements. However with practice, the subjects' hand trajectories in the force field converged to a path similar to that produced in absence of any perturbing force (figure 2.5top)

Subjects' recovery of performance is due to learning. After the training had been established, the force field was unexpectedly removed for the duration of a single hand movement. The resulting trajectories (figure 2.5bottom), named 'after-effects', were approximately mirror images of those that the same subjects produced when they had initially been exposed to the force field. The emergence of after-effects indicates that the CNS had composed an internal model of the external field. For example, in motor learning, neurons in the primary motor cortex (M1) and the supplementary motor area (SMA) ([59]) are found to encode the changes in movement dynamics. Such changes in neuronal activities represent the adaptation outcomes of kinematics-to-dynamics transformation, which could represent either direct or indirect adaptive control. The internal model was generating patterns of force that effectively anticipated the disturbing force that the moving hand was encountering. The fact that these learned forces compensate for the disturbances applied by the robotic arm during the subjects' reaching movement indicates that the CNS programs these forces in advance. The after-effects demonstrate that these forces are not the product of some reflex compensation of the disturbing field. They are, instead, a feed forward motor command.

2.3.5 The role of sensory feedback: adaptation and learning

In ([60]) it was studied how deafferentation, that is the elimination of sensory inflow, modifies muscles' synergy structure and activation.
It was found that the spinalized frog was still able to swim and jump almost in the same way, again confirming the central pattern generators hypothesis. Moreover, it was found that the functional form of most of the synergies is invariant under sensory feedback. This means two things: first, that their dependence upon space and time is not changed if a feedback signal, like a spinal reflex, is present; second, that the supraspinal motor centers are organized in modules, in the same way as the spinal chord motor primitives are.
So, what is the real role, if there is any, of sensory feedback?
Sensory inflows adapt the recruitment of synergies to the unpredicted constraints imposed by the task. That is they can *finely* tune the activation coefficient, c_i of 2.3. This is particularly true for fast movement, such as jumping, when the CNS does not have the time to compensate for uncertainties in the feed forward way, due to long delays between sensory apparatus and the CNS. In particular it is

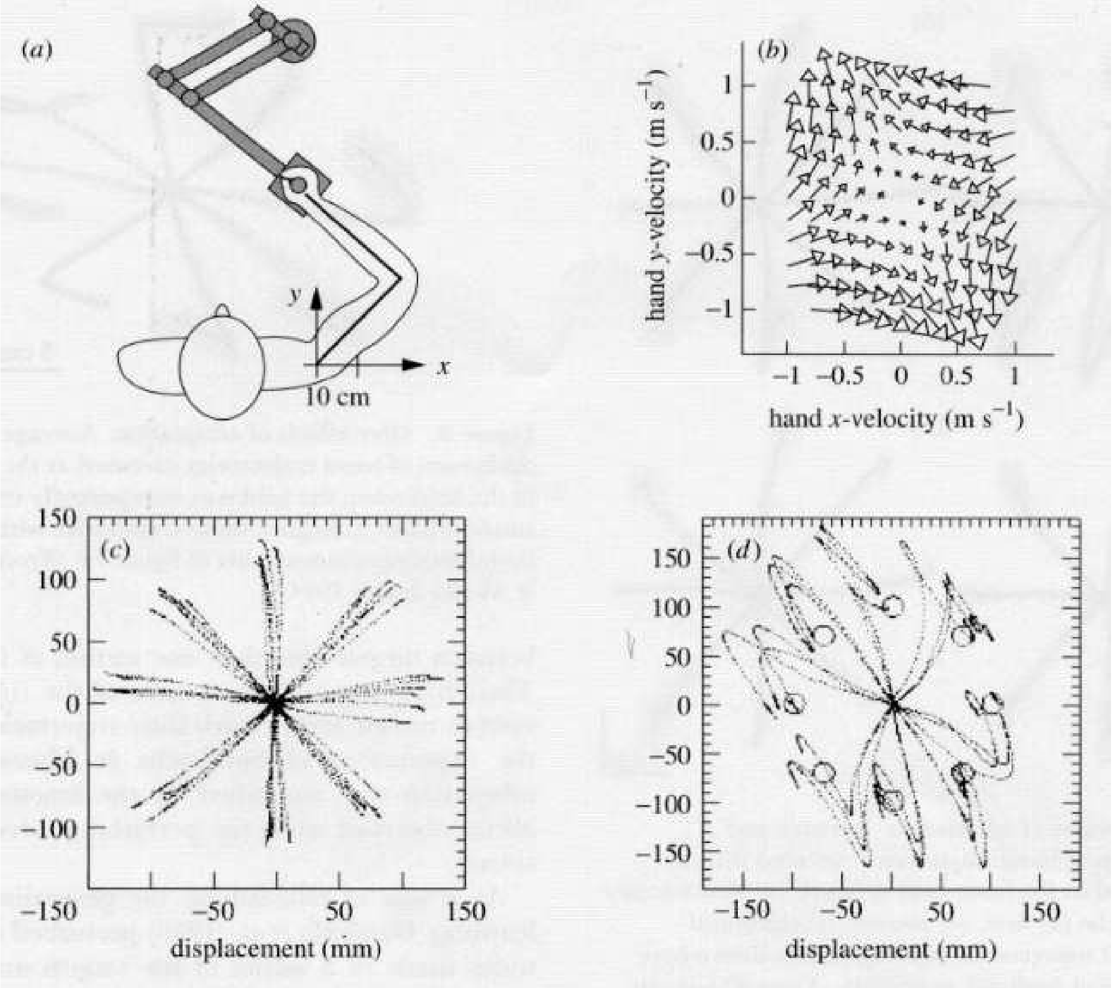


Figure 2.4: Adaptation to external force fields. (a) Sketch of the experimental apparatus. Subjects executed planar arm movements while holding the handle of an instrumented manipulandum. A monitor (not shown) placed in front of the subjects and above the manipulandum displayed the location of the handle as well as targets of reaching movements. The manipulandum was equipped with two computer-controlled torque motors, two joint-angle encoders and a six-axes force transducer mounted on the handle. (b) Velocity-dependent force field corresponding to the expression

$$F = B\mathbf{v}, \quad withB = \begin{bmatrix} -10.1 & -11.2 \\ -11.2 & 11-1 \end{bmatrix}$$

The manipulandum was programmed to generate a force F that was linearly related to the velocity of the hand, $\mathbf{v} = [v_x, v_y]$ Note that the matrix B has a negative and a positive eigenvalue. The negative eigenvalue induces a viscous damping at 23 whereas the positive eigenvalue induces an assistive destabilizing force at 113. (c) Unperturbed reaching trajectories executed by a subject when the manipulandum was not producing disturbing forces (null field). (d) Initial responses observed when the force field shown in (b) was applied unexpectedly. The circles indicate the target locations(modified from [45]).

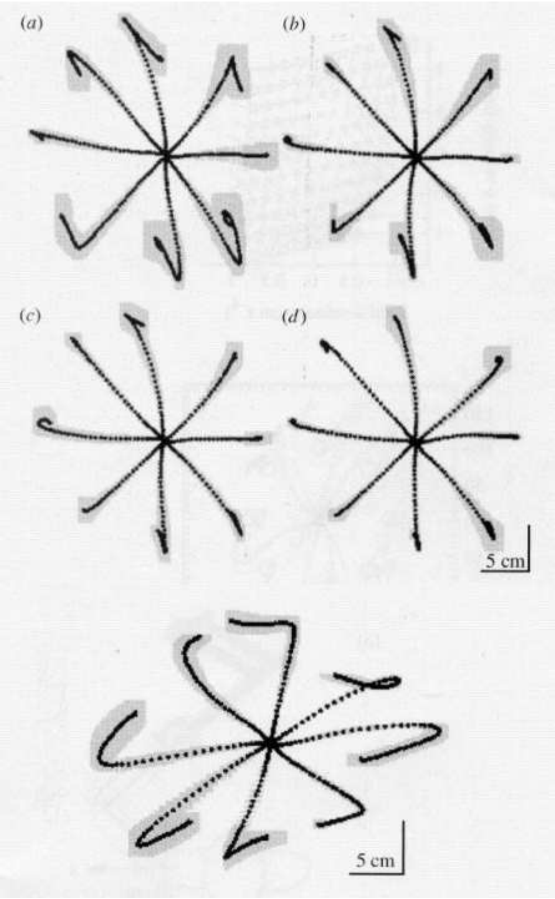


Figure 2.5: Top: Time-course of adaptation. Average and standard deviation of hand trajectories executed during the training period in the force field of figure 2.4b. Performance is plotted during the (a) first, (b) second, (c) third and (d) final set of 250 movements. All trajectories shown here were under no-visual feedback condition (from [45]). Bottom: After-effects of adaptation. Average and standard deviations of hand trajectories executed at the end of training in the field when the field was unexpectedly removed on random trials. Compare these trajectories with the initial-exposure movements of figure 2.4d (from [45]).

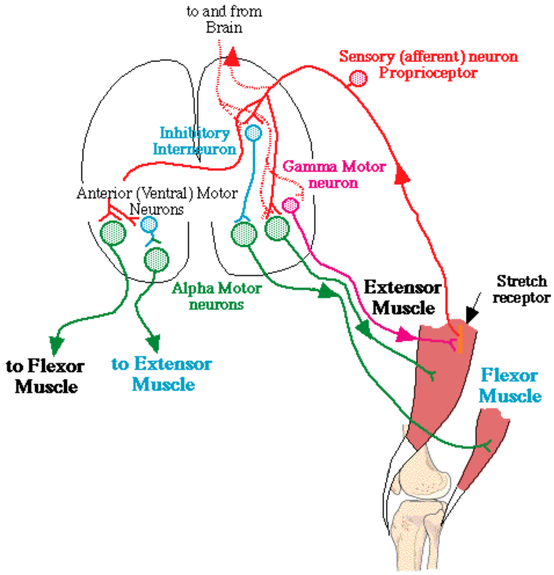


Figure 2.6: Schematic representation of a spinal reflex. Each kind of neuron represented should be interpreted as the neuron pool relative to a certain muscle synergy.

found that, during jump extension, feedback plays a prominent role in enhancing the activation amplitude of the related synergies. During jump extension, the frog needs to generate more power to accelerate the body against gravity within the shortest possible time independently of external dynamical condition, for example ground reactivity. Power generation might be enhanced by increased activation of the Golgi tendon organs and Ib interneurons, assisting in maximal activation of the extensors' synergies through a positive feedback loop.

The analysis of coefficient amplitudes also reveals that feedback uncouples multiple synergies that are centrally organized; therefore, afferents might allow for more individuated control of each synergy. Conversely, the synergies coupled by feedback may be subservant to similar biomechanical functions.

Sensory feedback is also due to discrepancies between expected and actual behavior of the muscular system. In particular, different type of neurons and relative fibers bring to the spinal chord information about muscle length (II-muscle spindle, static response), and length and rate of change of length, (Ia-muscle spindle, dynamic response). The second kind of response is modulated by the gamma muscle spindle, which alters the sensitivity of the Ia-muscle spindle (an introduction to these concepts can be found in ([61])). This way it is possible to set an activation level that induces a *task-dependent sensory inflow*. Let us make it clear with an example. If the planned motion is to move the right hand in a straight line from the chest to a glass of water in front of us at a certain velocity, and an unpredicted force, like a strong wind (or the force field induced by the manipulandum of section

2.3.4), slightly deviates our arm to the right, then the Ia-muscle spindle will fire in consequence of such deviation and a spinal reflex motor command, due to some changes in synergy activation (fine tuning), will contrast this deviation to bring the hand back in track, i.e no Ia spindle firing (see figure 2.6).

We stress that higher loop reflexes are also possible at the supraspinal level (through visual feedback, and slower reflexes, for example), but here we are interested only in the fast, low level adaptation due to the muscle-skeletal system proprioceptors and spinal motor synergy tuning through sensory feedback. Neilson et al. suggested that motor learning may be modeled as an internal model representation by some adaptive filter networks in the brain. Kawato et al. ([20]) proposed a feedback error learning model as a possible mechanism of adapting internal models (see section 1.2, figure 1.7). To support this hypothesis in vertebrate motor learning, R.Osu et al. ([62]) in the same experiment as that of Figure 2.4 measured the hand path error between actual and desired trajectory. Their results completely support the feedback error learning hypothesis.

Furthermore, in the case of vertebrate motor learning the error signal sent to the CNS will not be the reflex motor command itself, but the change in synergy activations.

2.4 The proposed model and relative problems

2.4.1 Survey of movement learning and related problems

From what was explained in the last section, it seems that the vertebrate movement learning system can be devised in two parts:

- the muscle-skeletal system and the spinal cord
- the central nervous system (motor cortex, supraspinal motor center, etc ...)

The first one serves to actuate motor commands through the combination of a few modules on the base of feedforward supraspinal activation commands, to retrieve information from the environment and to adapt to unpredictable changes (fine tuning). The second one serves to learn appropriate activation patterns in order to execute planned movements and to compensate for external dynamic constraints in feed forward manner, that is learning the inverse dynamic.

Their common aim is to stabilize the (expected) instantaneous virtual trajectory point.

The CNS is very likely to learn the inverse dynamic through feedback error learning. In any case, the supraspinal descending commands do not deal with actual muscle inputs, but with codified synergy activation patterns. Indeed, there are no

reasons to think that the feedback error signal will be given in term of changes in muscle inputs, or sensory inflows, rather than in changes in the activation of motor primitives. From this point of view, we can imagine the CNS to be completely disconnected from afferents and muscle activation, as the only necessary signals between the actuator (muscle-skeletal system & spinal cord) and the learning apparatus (CNS) are codified muscle synergy activation patterns (see figure 2.8).
Let us make this clearer with the perturbed hand example. Let $\mu \in \Re^p$ be a real vector which parametrize the environment dynamics and $\nu \in \Re^p$ the present knowledge that the CNS has of it. Before learning, in order to move the hand in a straight line a set of motor primitive activation coefficients, parametrized by the present knowledge of the inverse dynamics, $\{c_i^\nu\}_{i=1,\ldots,K}$, descends from the CNS to the spinal chord, where i runs over the synergies. When the hand starts to move the unexpected perturbing force field deviates it and some muscles will have an unexpected elongation or elongation's rate of change. This generates a sensory response which reaches the spinal cord. Here, the only possible feedback reaction is to modulate the feed forward activation coefficients of the synergy/ies relative to the spiking muscles, in order to compensate for the deviation, that is

$$c_i^\nu \to \tilde{c}_i^\nu, \; i = 1, \ldots, K. \tag{2.4}$$

Spinal reflexes require some 10ms. Moreover, the changes in activation that result will be small (a fine tuning), and most of the time will not suffice to completely compensate for the unknown dynamics. This is why reflexes alone are not sufficient to stabilize the desired trajectory. In any case, changes due to their activation are sent to the central nervous system too. From computations, the CNS will derive a change in its knowledge of the system parameters and in the next trial of the same movement it will send a descending motor command based on the upgraded set of parameters. As long as the approximate knowledge does not suffice to correctly predict unknown dynamical properties some sensory inflows will tend to modify the feed forward activation coefficients, and so on until the internal model of the CNS is close enough to the actual environment dynamics, so as to stabilize the desired virtual trajectory.

This scheme implies different problems. First of all, limb, or plant, stability problems. In general, the exact knowledge of the system parameters can be only approximated through the internal representation, and physical disturbances (neural noise, trial to trial variability, etc ...) can perturb the muscle activations, or inputs. During all the learning phases, that is independently of the present knowledge of the inverse dynamics and perturbations, the system needs to be contained around the desired trajectory, even if not stabilized or, equivalently, there must exist a neighborhood W of the instantaneous virtual trajectory point which attracts the state of the system (see figure 2.7). This way, the system tends to

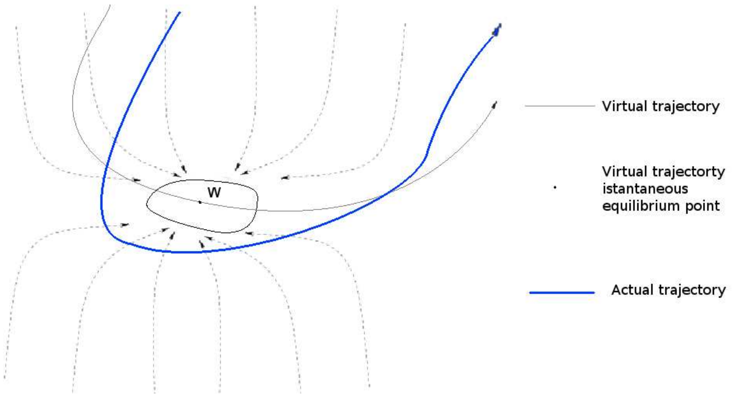


Figure 2.7: Independently from the initial conditions of the system, spring like characteristics of the muscle skeletal system and feed forward activation of its force fields create an attracting set, W, around the instantaneous point of the desired virtual trajectory which guarantees the boundness of actual trajectories. In the limit of exact knowledge and representation of the inverse dynamic W reduces to the instantaneous point of the desired virtual trajectory.

stay near the desired trajectory. Failure of this condition would imply injuries, or, in general, plant damage.

Then, the neurons in the CNS and the spinal neurons have to represent in some way the internal model and the set of activation coefficients. We do not want to enter the details of such representation. In any case, some basic knowledge on neuron dynamics brings us to model them as groups of coupled nonlinear oscillators ([63],[64]). This representation must be encoded in some way in the state of these oscillators ([65],[66],[67]) and this representation has to be stable under disturbances and uncertainties. Coupled nonlinear oscillators can show chaotic dynamics ([68]). As we will see in chapter 4, chaotic systems embed an infinity of unstable different dynamics, both periodic and chaotic, and the periodic ones can be stabilized through small perturbations (*chaos control*, [74]). We would point out that this not the case for non chaotic dynamics, where, in general, one would need perturbations of the same order of the unperturbed dynamic. The idea of assigning a different meaning to different periodic orbits has already been applied in Chaotic Neural Networks ([83][79]), and it is also the guiding idea of our neural representation model (as we will explain in section 5.3, a very easy and robust practical chaos control algorithm may also turn out to be biological plausible). In any case, it would also be important that this control is robust, otherwise even very small changes in the characteristics of the oscillators, that occur normally for real neurons, would compromise the stability of the representation.

Finally, the communication between the spinal cord and the brain can have long delays, compared to the time scales of the controlled dynamic. Hence, the CNS cannot send control input directly to the muscle skeletal system, but every descending command has to first be codified in the synergies activation coefficient stored in the spinal cord.

2.4.2 Control theory approach

The muscle skeletal system interacting with the environment can be modeled through a nonlinear control system, the plant, parametrized by a vector of real parameters, $\mu \in \Re^p$, which defines the changing conditions,

$$\dot{x} = f(x, u, \mu),$$

where $x \in \Re^n$, and $u \in \Re^m$ is the control variable. As we said, the problem of movement learning can be reduced to the problem of stabilizing the instantaneous virtual trajectory point. In analyzing this problem in the control theory framework we will then limit our attention to the stabilization of the origin of the system, and will drop time dependence. We further assume that there exists a stabilizing feedback law $u = k_\mu(x)$, which renders the closed loop system asymptotically stable at the origin. It is then natural to adopt an adaptive control point of view when trying to describe it. Vice-versa, adaptive control theory can inspire from vertebrate movement learning in order to synthesize efficient controllers. In what follows and in figure 2.8 we propose a possible vertebrate inspired adaptive control.

The activation of muscle synergies generates a nonlinear force field, $k_\nu(x)$, where $\nu \in \Re^p$ is the internal model approximation of the real of parameter vector μ. This force field will be given by the superposition of some invariant nonlinear force fields, $\{\phi_i\}_{i=1,\ldots,K}$, that is

$$u = k_\nu(x) = \sum_{i=1}^{K} c_i^\nu \phi_i(x) \tag{2.5}$$

where c_i^ν, $i = 1, \ldots, K$, is the feed forward activation coefficient, stored in the synergy activation tuner. In this case there is no state quantization, as the controller the controller is analogical. On the other hand, its output will be in general affected by strong disturbances, p. Moreover, the fact that only the combination of few modules is available will lead to further approximations of the parameter vector (see section 6.1). If we use a digital controller with bandwidth limitations, the effect of state quantization can be still reduced to a perturbation term on the output of an "ideal controller" (see section 3.2 for more details on this point).

In the case of an analogical controller, sensors contained in the plant will codify possible divergences from the origin, similarly to muscle spindles, and send a signal, r (sensory inflow), to the adaptive controller with a short delay. This is possible by defining a suitably measured (by the sensors) variable $\tilde{x}(t) = m(x(t)) \in \Re^s$ and a sensor activation level $S_0 \in \Re^s$ for this variable. This way, we define an admissible region, $\mathcal{A} \subset \Re^s$. If the system leaves this region a codified error signal is sent to the synergy tuner and synergy activation coefficients will be changed in accordance, through a suitable algorithm, thus leading to adaptation and, by sending these changes to the central controller as well, learning. In general the more synergies that are used, the more it will be possible to approximate the exact stabilizing feedback law, $k_\mu(x)$ (see section 6.1). Note that, if the feed forward coefficients already suffice to stabilize the desired dynamics no signal is necessary from sensors to the adaptive controller.

In the case of a digital controller, in order to define the role of sensors, we first have to specify the particular kind of controller we are using. As we will explain in the next chapter, we will choose a controller which makes use of minimal bit rate. Chapter 6 explains a possible sensor implementation for this controller, which makes use only of a particular statistical property of the coded variable and allows for finer parameter vector estimation.

Even if it is possible to store and modify activation coefficients in many ways, a possible one, which takes inspiration from chaotic neural dynamics, makes use of chaos control. Beside the biological aspect, there are other reasons to use this method:

- a single chaotic system can have an infinity of different ordered states (one for each unstable periodic dynamic); each of them can represent a different meaning and be modulated to carry different messages (such as standard sinusoidal waves modulation [82])

- positive entropy and symbolic dynamics makes chaotic systems natural digital information sources

- we will show that chaos control procedures are robust to noise and parameter uncertainties (see Chapter 5)

- chaos control can be implemented through simple microelectronic circuitry with low power consumption even at very high frequencies (some 10Mhz) (see [80] and the literature therein)

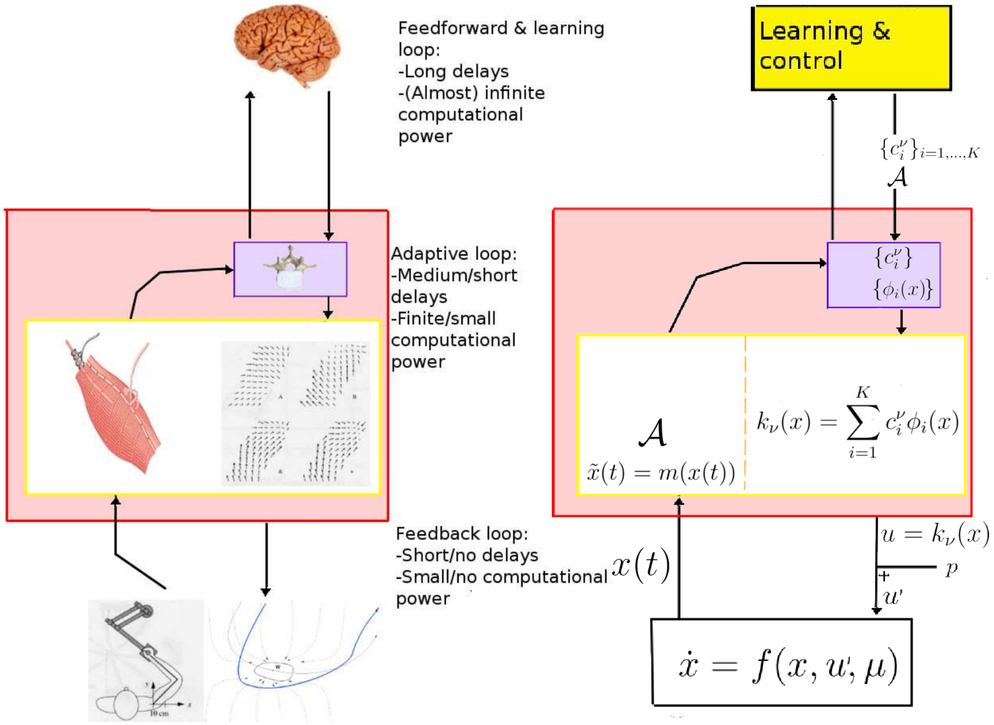
- chaos synchronization allows for secure and robust information transmission ([75])

Chaos control has already been used for chaotic neural networks, in which the chaotic dynamics represent the ground state, no stimulus, while different UPOs carries information about the stimulus (see [79],[83]). This system has been found to be surprisingly robust to noise (in accordance with the results we will give in chapter 5). In the following we will not focus on specific communication aspects

between the plant and the central controller, nor on specific learning algorithms, nor on specific activation coefficient codification through chaos control. Instead we will be interested in stability properties, both of the closed loop dynamics in case of parameter uncertainties, input disturbances and state quantization, and of the controlled chaotic dynamics, which will assure robust codification, in case of changing conditions or noise. Moreover we will try to specify efficient adaptation algorithms, both for the analogical and for the digital synergy controller.
Finally, the open "technical" problems are:

- find a general formalism to exploit stability analysis of nonlinear closed loop dynamics of the kind $\dot{x} = f(x, k_\nu(x) + p, \mu)$, where p is some disturbance, and, in general, $\nu \neq \mu$
- study chaos control stability

We will exploit these problems in Chapters 3-5.
Finally in Chapter 6 we describe two embedded adaptive control algorithms, one for the analogical controller (which includes a possible simple way of using chaos synchronization to implement reflex modification of synergy activation) and one for the digital controller (which makes use of a statistical property of the coded variable to tune synergy activation).
This way we can partially neglect the network problems (delays and communication constraints) explained in section 1.2, as the simple embedded controller already suffices to keep the system bounded near the origin, to obtain (partial) adaptation and to send appropriate signals to the learning controller (the brain).



Feedforward & learning loop:
-Long delays
-(Almost) infinite computational power
Adaptive loop:
-Medium/short delays
-Finite/small computational power
Feedback loop:
-Short/no delays
-Small/no computational power
10 cm
Learning & control
$\{c_i^\nu\}_{i=1,\dots,K}$
$\mathcal{A}$
$\{c_i^\nu\}$
$\{\phi_i(x)\}$
$\mathcal{A}$
$\tilde{x}(t) = m(x(t))$
$k_\nu(x) = \sum_{i=1}^{K} c_i^\nu \phi_i(x)$
$x(t)$
$u = k_\nu(x)$
p
+
u'
$\dot{x} = f(x, u', \mu)$

Figure 2.8: Left: Biological model of vertebrate movement control and learning through feedforward activations of spinal force fields and the relative reflex adaptation. Right: inspired adaptive control scheme. For the biological model stabilization refers to the desired instantaneous virtual trajectory point, while for the control model it refers to the origin. Explanation: a task dependent activation command, composed by a feedforward synergy activation pattern, $\{c_i^\nu\}_{i=1,\dots,K}$, parametrized by the present knowledge of the system parameter vector ν is sent from the learning/control apparatus (CNS) to synergy feedback controller (red region, the muscle skeletal system and the spinal cord). In particular it is stored in the synergy tuner (blu region, the spinal cord) and serves to activate a fixed set of nonlinear force fields, $\{\phi_i\}_{i=1,\dots,K}$ (spinal cord motor primitives) which add vectorially, weighted with the relative activation coefficient. In the figure the i-th synergy is activated with coefficient c_i, and not c_i^ν, because the possible reflex modification of the feedforward activation was already included. The resulting nonlinear force field, $k(x)$ (muscle co-contraction force field) is the input, u of a nonlinear control system, $\dot{x} = f(x, u, \mu)$, that models the interaction of the limb with the external dynamics. In real cases there will be input perturbations p, due to different causes (neural noise, trial to trial variability, state quantization in the case of a digital controller, ...). The state of the system, $x(t)$ is fed back to the controller. Here it has two roles: (i) obviously it determines the control input $u(t) = k(x(t))$, (ii) it is measured by *adaptive* sensors (muscle proprioceptors: muscle spindles), $\tilde{x}(t) = m(x(t))$. The state of $\tilde{x}$ is compared to a task dependent admissible region, $\mathcal{A}$ (determined by the gamma muscle spindle), and in case it leaves this region an error signal (sensory inflow) is sent to the synergy activation tuner (the spinal cord). On the base of a predetermined tuning algorithm, the synergy activation tuner will modify the activation coefficient of some synergies (sensory feedback). These changes are sent to the learning/control apparatus, too, which will then upgrade its internal model representation, similarly to the feedback error learning scheme, through some adaptive filter (neural network, Kalman filter, ...). Delays and computational power refer to the relative loop: feedforward & learning, adaptive, feedback. Delay magnitude is compared to typical dynamic time scales.

Chapter 3

Input-to-state stability and consequences

We want to build a formal framework to exploit stability analysis of nonlinear dynamical systems. We introduce a general definition for stability with respect to a given set, which allows for very powerful technical procedures. To exploit the power of such formalism, we show how it is possible to synthesize a digital feedback controller which makes use of minimal data rate and which succeeds in stabilizing unstable dynamics in the case of parameter uncertainties, too. As a paradigmatic example, we synthesize such control for the inverted pendulum on a cart with unknown length.

3.1 Characterizations of Input-to-state stability

This section is based on [70].

Consider a general dynamical system of the form

$$\dot{x} = f(x, \omega), \tag{3.1}$$

where $x \in \Re^n$ is the states variable, $\omega \in \Re^m$ is some disturbance acting on the system and $f : \Re^n \times \Re^m \to \Re^n$ is a locally Lipschitz function, with solution $x(t, x_0, \omega)$. The questions to be addressed all concern the study of the size of each solution as a function of the initial condition $x_0 = x(0)$ and the magnitude of the perturbation $\omega(\cdot)$.
For nonlinear systems, there is no complete agreement as yet regarding what are the most useful formulations of system stability with respect to input perturbations. One candidate for such a formulation is the property called ”input-to-state stability” (ISS), introduced in ([69]).

In some cases, authors have suggested apparent variations of the ISS property. Here we want to point out that ISS is theoretically equivalent to many other stability formulations for nonlinear systems, and, thus, a quite general assumption. For practical purposes, a weaker formulation, the local ISS (LISS), is equivalent to the trivial assumption that the unperturbed system is globaly asymptotic stable (0-GAS).

All the results in this section are taken from ([70]) and can easily be generalized to discrete time systems.

3.1.1 Basic definition and Notations

Euclidean norm in $\Re^n$ or in $\Re^m$ is denoted simply as $|\cdot|$. Instead of single fixed point, we will study the notions relative to nonempty subset $\mathcal{A} \subset \Re^n$; for such set and $\xi \in \Re^n$, $|\xi|_{\mathcal{A}} = \inf\{d(\eta, \xi), \eta \in \mathcal{A}\}$ denotes the point to set distance from ξ to $\mathcal{A}$. We also let, for each $\epsilon > 0$:

$$B(\epsilon, \mathcal{A}) := \{\xi \mid |\xi|_{\mathcal{A}} < \epsilon\}, \quad \overline{B}(\epsilon, \mathcal{A}) := \{\xi \mid |\xi|_{\mathcal{A}} \leq \epsilon\}.$$

By input (disturbance) we mean a measurable and locally essentially bounded function $u : \mathcal{I} \to \Re^m$, where $\mathcal{I}$ is a subinterval of $\Re$ which contains the origin, so that $u \in U \subset \Re^m$ for almost all t.

Given any input u defined on an interval $\mathcal{I}$ and $\xi \in \Re^n$, there is a unique maximal solution of the initial value problem $\dot{x} = f(x, u),\ x(0) = \xi$. This solution is defined on some maximal open subinterval of $\mathcal{I}$, and it is denoted by $x(\cdot, \xi, u)$. A *forward complete system* is one such that, for each u defined on $\mathcal{I} = \Re_{\geq 0}$, and each ξ, the solution $x(t, \xi, u)$ is defined on the entire interval $\Re_{\geq 0}$. The L^m_∞-norm of an input u is denoted by $\|u\|_\infty$. That is, $\|u\|_\infty$ is the smallest number c such that $|u(t)| \leq c$ for almost all $t \in \mathcal{I}$.

A function $F : S \to \Re$, $S \subset \Re^n$ containing 0 is *positive definite* if $F(x) > 0$ for all $x \in S$, $x \neq 0$, and $F(0) = 0$. A function $\gamma : \Re_{\geq 0} \to \Re_{\geq 0}$ is *of class $\mathcal{N}$* (or an *$\mathcal{N}$ function*) if it is continuous and non-decreasing; it is an *$\mathcal{N}_0$ function* if in addition it satisfies $\gamma(0) = 0$. A function $\gamma : \Re_{\geq 0} \to \Re_{\geq 0}$ is a *$\mathcal{K}$ function* if it is continuous, positive definite, and strictly increasing, and a *$\mathcal{K}_\infty$ function* if it is also unbounded. Finally, recall that $\beta : \Re_{\geq 0} \times \Re_{\geq 0} \to \Re_{\geq 0}$ is said to be a function of class $\mathcal{KL}$ if for each $t \geq 0$, $\beta(\cdot, t)$ is of class $\mathcal{K}$ and for each fixed $s \geq 0$,$\beta(s, t)$ decreases to zero as $t \to \infty$, and of class $\mathcal{KK}$ if for each $t \geq 0$, $\beta(\cdot, t)$ is of class $\mathcal{K}$ and for each fixed $s \geq 0$, $\beta(s, \cdot)$ is of class $\mathcal{K}$

3.1.2 Catalog of properties

A *zero-invariant set* $\mathcal{A}$ for a system (3.1) is a subset $\mathcal{A} \subseteq \Re^n$ with the property that $x(t, \xi, 0) \in \mathcal{A}$ for all $t \geq 0$ and all $\xi \in \mathcal{A}$.

From now on, all definitions are with respect to a given *forward complete* system as in equation (3.1), and a given compact zero-invariant set $\mathcal{A}$ for this system. The main definitions follow.

We first recall the definition of the ISS property:

$$\exists\gamma \in \mathcal{K}, \beta \in \mathcal{KL} \ : \ \forall\xi \in \Re^n \ \forall u(\cdot) \ \forall t \geq 0$$

$$|x(t, \xi, u)|_{\mathcal{A}} \leq \beta(|\xi|_{\mathcal{A}}, t) + \gamma(\|u\|_\infty). \tag{3.2}$$

It is known that a system is ISS if and only if it satisfies a *dissipation inequality*, that is to say, there exists a smooth $V : \Re^n \to \Re_{\geq 0}$ and there are functions $\alpha_i \in \mathcal{K}_\infty, \ i = 1, 2, 3$ and $\sigma \in \mathcal{K}$ so that

$$\alpha_1(|\xi|_{\mathcal{A}}) \leq V(\xi) \leq \alpha_2(|\xi|_{\mathcal{A}}) \tag{3.3}$$

and

$$\nabla V(\xi) f(\xi, v) \leq \sigma(|v|) - \alpha_3(|\xi|_{\mathcal{A}}) \tag{3.4}$$

for each $\xi \in \Re^n$ and $v \in \Re^m$. A very useful modification of this characterization is the fact that ISS is also equivalent to the existence of a smooth V satisfying (3.3) and Equation(3.4) replaced by an estimate of the type $\nabla V(\xi) f(\xi, v) \leq -V(\xi) - \alpha_3(|\xi|_{\mathcal{A}})$. A weaker version, yet very useful in practice, of the ISS property is the *local* ISS (LISS), which requires that both the initial state and control are small, that is:

$$\exists\rho > 0, \gamma \in \mathcal{K}, \beta \in \mathcal{KL} : \forall |\xi|_{\mathcal{A}} \leq \rho, \forall \|u\|_\infty \leq \rho$$

$$|x(t, \xi, u)|_{\mathcal{A}} \leq \beta(|\xi|_{\mathcal{A}}, t) + \gamma(\|u\|_\infty) \ \ \forall t \geq 0. \tag{3.5}$$

For the unforced system ($u = 0$) we will use the prefix 0-. We can express *0-global asymptotic stability*(0-GAS) as

$$\exists\beta \in \mathcal{KL} : \forall\xi \in \Re^n \ \forall t \geq 0$$

$$|x(t, \xi, 0)|_{\mathcal{A}} \leq \beta(|\xi|_{\mathcal{A}}, t), \tag{3.6}$$

and *0-asymptotic stability*(0-AS) as

$$\exists\rho > 0, \beta \in \mathcal{KL} : \forall\xi \in \Re^n \ \forall t \geq 0$$

$$|x(t, \xi, 0)|_{\mathcal{A}} \leq \beta(|\xi|_{\mathcal{A}}, t). \tag{3.7}$$

Next we introduce some concepts about the perturbed system. The *limit property* (LIM) holds if every trajectory must at some time get to within a distance of $\mathcal{A}$ which is a function of the magnitude of the input:

$$\exists\gamma \in \mathcal{N}_0 : \forall\xi \in \Re^n \ \forall u(\cdot)$$

$$\inf_{t\geq 0} |x(t,\xi,u)|_{\mathcal{A}} \leq \gamma(\|u\|_\infty). \tag{3.8}$$

The *asymptotic gain property*(AG) holds if every trajectory must ultimately stay not far from $\mathcal{A}$, depending on the magnitude of the input:

$$\exists \gamma \in \mathcal{N}_0 : \forall \xi \in \Re^n \ \forall u(\cdot)$$

$$\overline{\lim_{t\to\infty}} |x(t,\xi,u)|_{\mathcal{A}} \leq \gamma(\|u\|_\infty). \tag{3.9}$$

Now we can state some of the most notable ISS equivalences:

Theorem 1 (E.D Sontag, [70]) *Assume given any forward complete system as in equation (3.1) and a compact zero-invariant set $\mathcal{A}$ for this system. The following properties are equivalent:*

a) ISS

b) LIM & 0-AS

c) LIM & 0-GAS

d) AG & 0-GAS

e) AG & LISS

We find it convenient to give also the following implications:

$$\text{ISS} \Rightarrow \text{0-GAS} \Rightarrow \text{LISS}. \tag{3.10}$$

3.2 ISS and control system

We want to deal with a general nonlinear control system, whose dynamics is parametrized by a real vector μ

$$\dot{x} = f(x,u,\mu) \quad (\ x(l+1) = f(x(l),u(l),\mu)\ ) \tag{3.11}$$

where $x \in \Re^n$ is the state variable, $u \in \Re^m$ is the control variable, $\mu \in \Omega \subset \Re^p$ is a parameter which changes the dynamics of the system, and $f : \Re^n \times \Re^m \times \Re^p \to \Re^n$ is a Lipschitz function satisfying $f(0,0,\mu) = 0 \ \forall \mu$. Note that f is Lipschitz also with respect to the parameter.

All the results can be generalized to discrete time systems, for which we will only give results without demonstration, but explaining the necessary adjustments to notation in round brackets.

In what follows, we find it convenient to use the infinity norm $\|x\|_\infty := \max\{|x_i| :$

$1 \leqslant i \leqslant n\}$ on $\Re^n$. We let $B^n_\infty(x_0, r)$ denote a ball with respect to this norm with radius r and center x_0.
First of all we formalize the ISS assumption to our system. In this case the perturbation will sum to the feedback function in the control variable and we can write:

***Assumption* 1** *: System (3.11) admits a Lipschitz, both in the state and in the parameter vector dependence, feedback law* $u = k_\mu(x)$ *(* $u(l) = k_\mu(x(l))$ *),* $\mu \in \Omega$, *which satisfies* $k_\mu(0) = 0$ *and renders the close-loop system input-to-state stable (ISS).*

Written in terms of the infinity norm and for piecewise continuous inputs (which is sufficient for our purposes), this condition means that there exist functions $\beta \in \mathcal{KL}$ and $\gamma_{\text{pert}} \in \mathcal{K}$ such that for every initial condition $x(t_0)$ and every piecewise continuous signal p the corresponding solution of the system

$$\dot{x} = f(x, k_\mu(x) + p, \mu) \quad (\ x(l+1) = f(x(l), k_\mu(x(l)) + p(l), \mu)\ ) \tag{3.12}$$

satisfies

$$\|x(t)\|_\infty \leqslant \beta(\|x(t_0)\|_\infty, t - t_0) + \gamma_{\text{pert}}\left(\sup_{t_0 \leq s \leq t} \|p(s)\|_\infty\right) < \infty,\ \forall t \geqslant t_0 \tag{3.13}$$

$$\left(\ \|x(l)\|_\infty \leqslant \beta(\|x(l_0)\|_\infty, l - l_0) + \gamma_{\text{pert}}\left(\sup_{l_0 \leq s \leq l} \|p(s)\|_\infty\right) < \infty,\ \forall l \geqslant l_0\ \right). \tag{3.14}$$

Now we want to show some consequences of the ISS. In particular we will derive the following implications:

- ISS ⇒ ISS with respect to measurement error (ISSwrme)
- ISS ⇒ *containability* (i.e the system remains bounded around the unstable fixed point) when there are uncertainties over system parameters (PC)
- ISS ⇒ ISC (Input-to-state containability)
- ISS ⇒ ISC with respect to measurement error (ISCwrme)

For each point, we will use the fact that the control function $k_\mu(\cdot)$ is Lipschitz, both in the state and in the parameter vector.

3.2.1 ISS⇒ ISS with respect to measurement error

ISSwrme means that the perturbation does not act directly on the input u of the system, but on the measure of the state used to apply the control law

$$k_\mu(x) \to k_\mu(x+e) \quad (\ k_\mu(x(l)) \to k_\mu(x(l)+e(l))\ ).$$

In any case, we can reduce this measure error to an input perturbation by writing

$$k_\mu(x+e) = k_\mu(x) + \int_0^1 \left[\frac{\partial k_\mu(y)}{\partial y}\right]_{y=x+\alpha e} d\alpha \cdot e = k_\mu(x) + p_e(x) \tag{3.15}$$

$$\Bigg(k_\mu(x(l)+e(l)) = k_\mu(x(l)) + \int_0^1 \left[\frac{\partial k_\mu(y)}{\partial y}\right]_{y=x(l)+\alpha e(l)} d\alpha \cdot e(l)$$
$$= k_\mu(x(l)) + p_{e(l)}(x(l)) \Bigg) \tag{3.16}$$

with $p_e(x)$ ($p_{e(l)}(x(l))$) bounded due to the Lipschitz property of the control function.
Now we can define the function $\gamma_{\text{err}} \in \mathcal{K}$ by

$$\gamma_{\text{err}}(\|e\|) = \sup_{x\in A}\left(\sup_{x'\in B^n_\infty(x,\|e\|)} \gamma_{\text{pert}}\left(\|p_{e'}(x)\|\right)\right), \tag{3.17}$$

where A is a suitable set and $e' = x' - x$.
Thus, written in terms of the infinity norm and for piecewise continuous inputs , the ISSwrme condition, means that there exist functions $\beta \in \mathcal{KL}$ and $\gamma_{\text{err}} \in \mathcal{K}$ such that for every initial condition $x(t_0)$ and every piecewise continuous signal e the corresponding solution of the system

$$\dot{x} = f(x, k_\mu(x+e), \mu) \quad (\ x(l+1) = f(x(l), k_\mu(x(l)+e(l)), \mu)\ ) \tag{3.18}$$

satisfies

$$\|x(t)\|_\infty \leqslant \beta(\|x(t_0)\|_\infty, t-t_0) + \gamma_{\text{err}}\left(\sup_{t_0\le s\le t}\|e(s)\|_\infty\right) < \infty,\ \forall t \geqslant t_0, \quad A = \{x(s)\}_{t_0\le s\le t} \tag{3.19}$$

$$\Bigg(\|x(l)\|_\infty \leqslant \beta(\|x(l_0)\|_\infty, l-l_0) + \gamma_{\text{err}}\left(\sup_{l_0\le s\le l}\|e(s)\|_\infty\right) < \infty,\ \forall l \geqslant l_0,$$
$$A = \{x(s)\}_{l_0\le s\le l} \Bigg). \tag{3.20}$$

3.2.2 ISS $\Rightarrow$ Containability

Uncertainties over system parameters means that in computing the feedback function, which will depend in some way on them, we are using a wrong parameter vector. That is

$$k_\mu(x) \to k_\nu(x) \quad (\ k_\mu(x(l)) \to k_\nu(x(l))\ )$$

Setting $k(x,\mu) \equiv k_\mu(x)$, also in this case we can reduce this error to an input perturbation by writing

$$k_\nu(x) = k_\mu(x) + \int_0^1 \left[\frac{\partial k(x,\eta)}{\partial \eta}\right]_{\eta=\mu+\alpha(\nu-\mu)} d\alpha \cdot (\nu - \mu) = k_\mu(x) + p_{\mu\nu}(x) \quad (3.21)$$

$$k_\nu(x(l)) = k_\mu(x(l)) + \int_0^1 \left[\frac{\partial k(x(l),\eta)}{\partial \eta}\right]_{\eta=\mu+\alpha(\nu-\mu)} d\alpha \cdot (\nu - \mu) = k_\mu(x(l)) + p_{\mu\nu}(x(l)) \quad (3.22)$$

with $p_{\mu\nu}(x)$ ($p_{\mu\nu}(x(l))$) bounded due to the Lipschitz property of the control function.
Now we can define the function $\gamma_{\text{par}} \in \mathcal{K}$ by

$$\gamma_{\text{par}}(\Delta\mu) = \sup_{x\in A}\left(\sup_{\nu'\in B^p_\infty(\mu,\Delta\mu)} \gamma_{\text{pert}}\left(\|p_{\mu\nu'}(x)\|\right)\right), \quad (3.23)$$

where A is a suitable set, $\Delta\mu = \|\nu - \mu\|_\infty$.
Thus, written in terms of the infinity norm and for piecewise continuous inputs, the Containability condition means that there exist functions $\beta \in \mathcal{KL}$ and $\gamma_{\text{par}} \in \mathcal{K}$ such that for every initial condition $x(t_0)$ the corresponding solution of the system

$$\dot{x} = f(x, k_\nu(x), \mu) \quad (\ x(l+1) = f(x(l), k_\nu(x(l)), \mu)\ ) \quad (3.24)$$

satisfies

$$\|x(t)\|_\infty \leqslant \beta(\|x(t_0)\|_\infty, t - t_0) + \gamma_{\text{par}}(\Delta\mu) < \infty,\ \forall t \geqslant t_0, \quad A = \{x(s)\}_{t_0\leq s\leq t} \quad (3.25)$$

$$(\ \|x(l)\|_\infty \leqslant \beta(\|x(l_0)\|_\infty, l - l_0) + \gamma_{\text{par}}(\Delta\mu) < \infty,\ \forall l \geqslant l_0, \quad A = \{x(s)\}_{l_0\leq s\leq l}\ ). \quad (3.26)$$

3.2.3 ISS $\Rightarrow$ Input-to-State Containability (ISC)

In this case we have a perturbation acting on the input and a wrong parameters vector, that is

$$k_\mu(x) \to\ k_\nu(x) + p \quad (\ k_\mu(x(l)) \to\ k_\nu(x(l)) + p(l)\ ).$$

We define the functions $p_{\mu\nu}(x)$ as in the case of containability and define the function $\tilde{\gamma} \in \mathcal{KK}$ by

$$\tilde{\gamma}(\Delta\mu, \|p\|) = \sup_{x \in A} \left(\sup_{\nu' \in B^p_\infty(\mu, \Delta\mu)} \left(\sup_{p' \in B^m_\infty(0, \|p\|)} \gamma_{\text{pert}} \left(\|p_{\mu\nu'}(x) + p'\|\right) \right) \right), \qquad (3.27)$$

where A is a suitable set, $\Delta\mu = \|\nu - \mu\|_\infty$.
To separate the dependence from parameter uncertainties and input perturbation we can write

$$\tilde{\gamma}(\Delta\mu, \|p\|) = r_{\mu\nu} + (\tilde{\gamma}(\Delta\mu, \|p\|) - r_{\mu\nu}) = r_{\mu\nu} + \gamma_{\mu\nu}(\|p\|), \quad \text{where } r_{\mu\nu} = \tilde{\gamma}(\Delta\mu, \|p\|)\big|_{p=0}$$

Thus, written in terms of the infinity norm and for piecewise continuous inputs, the ISC condition, means that there exist functions $\beta \in \mathcal{KL}$ and $\gamma_{\mu\nu} \in \mathcal{K}$ such that for every initial condition $x(t_0)$ the corresponding solution of the system

$$\dot{x} = f(x, k_\nu(x) + p, \mu) \quad (\ x(l+1) = f(x(l), k_\nu(x(l)) + p(l), \mu)\ ) \qquad (3.28)$$

satisfies

$$\|x(t)\|_\infty \leqslant \beta(\|x(t_0)\|_\infty, t - t_0) + \gamma_{\mu\nu}\left(\sup_{t_0 \le s \le t} \|p(s)\| \right) + r_{\mu\nu} < \infty, \ \forall t \geqslant t_0, \quad A = \{x(s)\}_{t_0 \le s \le t} \qquad (3.29)$$

$$\left(\ \|x(l)\|_\infty \leqslant \beta(\|x(l_0)\|_\infty, l - l_0) + \gamma_{\mu\nu}\left(\sup_{l_0 \le s \le l} \|p(s)\| \right) + r_{\mu\nu} < \infty, \ \forall l \geqslant l_0, \right.$$
$$\left. A = \{x(s)\}_{l_0 \le s \le l} \ \right). \qquad (3.30)$$

3.2.4 ISS ⇒ ISC with respect to measurement error

In this case we have an error in the measurement of the state and a wrong parameters vector, that is

$$k_\mu(x) \to k_\nu(x + e) \quad (\ k_\mu(x(l)) \to k_\nu(x(l) + e(l))\ ).$$

Again, we set $k(x, \mu) \equiv k_\mu(x)$, and reduce both errors to an input perturbation by writing

$$k_\nu(x + e) = k_\mu(x) + \int_0^1 \left[\frac{\partial k(y, \eta)}{\partial y} \right]_{\substack{y = x + \alpha e \\ \eta = \mu + \alpha(\nu - \mu)}} d\alpha \cdot e +$$

$$+\int_0^1 \left[\frac{\partial k(y,\eta)}{\partial \eta}\right]_{\substack{y = x+\alpha e \\ \eta = \mu + \alpha(\nu-\mu)}} d\alpha \cdot (\nu - \mu) =$$

$$= k_\mu(x) + p_{e,\mu\nu}(x) + p_{\mu\nu,e}(x). \tag{3.31}$$

with $p_{e,\mu\nu}(x)$ and $p_{\mu\nu,e}(x)$ bounded due to the Lipschitz property of the control function.

Now we can define the function $\tilde{\gamma} \in \mathcal{KK}$ by

$$\tilde{\gamma}(\Delta\mu, \|e\|) = \sup_{x\in A}\left(\sup_{\nu'\in B^p_\infty(\mu,\Delta\mu)}\left(\sup_{x'\in B^n_\infty(x,\|e\|)} \gamma_{\text{pert}}\left(\|p_{e',\mu\nu'}(x) + p_{\mu\nu',e'}(x)\|\right)\right)\right), \tag{3.32}$$

where A is a suitable set, $\Delta\mu = \|\nu - \mu\|_\infty$ and $e' = x' - x$.

To separate the dependence from parameter uncertainties and input perturbation we can write

$$\tilde{\gamma}(\Delta\mu, \|e\|) = r_{\mu\nu} + (\tilde{\gamma}(\Delta\mu, \|e\|) - r_{\mu\nu}) = r_{\mu\nu} + \gamma_{\mu\nu}(\|e\|), \quad \text{where } r_{\mu\nu} = \tilde{\gamma}(\Delta\mu, \|e\|)\big|_{e=0} \tag{3.33}$$

Thus, written in terms of the infinity norm and for piecewise continuous inputs, the ISCwrme condition, means that there exist functions $\beta \in \mathcal{KL}$ and $\gamma_{\mu\nu} \in \mathcal{K}$ such that for every initial condition $x(t_0)$ the corresponding solution of the system

$$\dot{x} = f(x, k_\nu(x+e), \mu) \quad (\ x(l+1) = f(x(l), k_\nu(x(l)+e(l)), \mu)\ ) \tag{3.34}$$

satisfies

$$\|x(t)\|_\infty \leqslant \beta(\|x(t_0)\|_\infty, t - t_0) + \gamma_{\mu\nu}\left(\sup_{t_0\le s\le t}\|e(s)\|\right) + r_{\mu\nu} < \infty,\ \forall t \geqslant t_0, \quad A = \{x(s)\}_{t_0\le s\le t} \tag{3.35}$$

$$\left(\ \|x(l)\|_\infty \leqslant \beta(\|x(l_0)\|_\infty, l - l_0) + \gamma_{\mu\nu}\left(\sup_{l_0\le s\le l}\|e(s)\|\right) + r_{\mu\nu} < \infty,\ \forall l \geqslant l_0,\right.$$

$$\left. A = \{x(s)\}_{l_0\le s\le l}\ \right). \tag{3.36}$$

3.2.5 Quantized digital feedback with minimal bit rate

In this section we want to show how to define a digital controller that preserves the stability properties (asymptotic stability and containability) of the system. This will allow implementation of the ISS feedback laws more easily in a practical situation.

First, we will describe the codification and control algorithm, then we will prove

how this algorithm suffices to stabilize or, in the presence of parameter uncertainties, contain the system.
The statement of the main Theorem (2) is equal for the continuous and discrete time case. The few differences in the proof will be written in round brackets.

The control strategy and the proof of theorem 2 are inspired to [2]. The introduction of parameter uncertainties is an original result.

Control Strategy and Assumption

We assume that a finite subset $B^n_\infty(0, E_0)$ of $\Re^n$ to which the initial condition belongs is known (if this is not the case we can find it by *zooming out* the system, see ([2])).
From what we have seen in Section 3, ISS is equivalent to substituting Assumption 1 with the following

***Assumption* 2** *: There exists $\Omega \in \Re^p$ such that system (3.11) admits a locally Lipschitz feedback law $u = k_\nu(x)$ which satisfies $k(0) = 0$ and renders the close-loop system input-to-state containable (ISC) with respect to measurement errors, if $\nu \neq \mu$ and $\nu \in \Omega$, $\mu \in \Omega$ (see (3.40)) for the correct definition of Ω), and input-to-state stable (ISS) with respect to measurement errors, if $\nu = \mu$.*

Again, written in terms of the infinity norm and for piecewise continuous inputs (which is sufficient for our purposes), this condition means that there exists functions $\beta \in \mathcal{KL}$ and $\gamma \in \mathcal{K}_\infty$ such that for every initial condition $x(t_0)$ and every piecewise continuous signal e the corresponding solution of the system

$$\dot{x} = f(x, k_\nu(x+e), \mu) \quad (\ x(l+1) = f(x(l), k_\nu(x(l) + e(l)), \mu)\ ) \tag{3.37}$$

satisfies

$$\|x(t)\|_\infty \leqslant \beta_{\mu\nu}(\|x(t_0)\|_\infty, t - t_0) + \gamma_{\mu\nu}\left(\sup_{s\leqslant t} \|e(s)\|_\infty\right) + r_{\mu\nu} < \infty \forall t \geqslant t_0$$

$$\text{with } r_{\mu\nu} \geqslant 0, r_{\mu\nu} = 0 \,\text{if}\, \mu = \nu. \tag{3.38}$$

$$\left(\ \|x(l)\|_\infty \leqslant \beta(\|x(l_0)\|_\infty, l - l_0) + \gamma_{\mu\nu}\left(\sup_{s\leq l} \|e(s)\|\right) + r_{\mu\nu} < \infty\ \forall l \geqslant l_0.\ \right) \tag{3.39}$$

Take κ to be some class $\mathcal{K}_\infty$ function with the property that $\kappa_\nu(r) \geqslant \max_{\|x\|\leqslant r} \|k_\nu(x)\|$ for all $r \geqslant 0$. Then we have $\|k_\nu(x)\|_\infty \leqslant \kappa_\nu(\|x\|_\infty)\forall x$.
Let L be the Lipschitz constant for the function f on the region

$$\{(x, u) : \|x\|_\infty \leqslant D, \|u\|_\infty \leqslant \kappa_\nu(D)\} \times \Omega \tag{3.40}$$

where

$$D := \beta_{\mu\nu}(E_0, 0) + \gamma_{\mu\nu}(N^{1/n}E_0) + N^{1/n}E_0 + r_{\mu\nu} \tag{3.41}$$

and Ω is a convex subset of $\Re^p$. Define

$$\Lambda := e^{L\tau} \geq 1 \quad (\ \Lambda = L \geq 1,\ \text{if } L < 1 \text{ the system is already stable }) \tag{3.42}$$

Define

$$E_0' = \Lambda E_0$$

We know $x(0) \in B^n_\infty(0, E_0^{'})$, for sure.

***Assumption* 3** *The number $N^{1/n}$ is an odd integer. If it is not an integer, choose $N' \leqslant N$ such that $N'^{1/n}$ is an integer instead.*

The last assumption allows to define the encoding function q_0 as follows: Divide $B^n_\infty(0, \Lambda E_0)$ into N equal hipercubic boxes, numbered from 1 to N in some specific way, and let $q_0(x)$ be the number of the box that contains x, if $x \in B^n_\infty(0, \Lambda E_0)$, and 0, otherwise (actually, a 0 in this case means an error occurs, in this case we can shift to the "zooming out" stage of [2]). If $q_0(x(0)) > 0$, then the encoded measurement specifies a box with edges at most $2\Lambda E_0/N^{1/n}$ which contains $x(0)$. Let $\hat{x}(0)$ be the center of this box, we obtain

$$\|\hat{x}(0) - x(0)\| \leqslant \frac{\Lambda E_0}{N^{1/n}}.$$

For $t \in [0, \tau)$, we apply the control law

$$u(t) = k_\nu(\hat{x}(t)) \tag{3.43}$$

where $\hat{x}(\cdot)$ is the solution of $\dot{\hat{x}} = f(\hat{x}, u, \nu)$ with initial condition $\hat{x}(0)$. At time $t = \tau$, we consider the box $B^n_\infty(\hat{x}(\tau^-), \Lambda^2 E_0/(N^{1/n}))$. To define q_1 divide this box into N equal hypercubic boxes and let $q_1(x)$ be the number of the box that contains x, or $q_1(x) = 0$ if $x \notin B^n_\infty(\hat{x}(\tau^-), \Lambda^2 E_0/(N^{1/n}))$. If $q_1(x(\tau)) > 0$, then the encoded measurement singles out a box with edges at most $2\Lambda^2 E_0/(N^{1/n})^2$ which contains $x(\tau)$. Let $\hat{x}(\tau)$ be the center of this box to obtain

$$\|\hat{x}(\tau) - x(\tau)\| \leqslant \frac{\Lambda^2 E_0}{(N^{1/n})^2} = E_1$$

and continue. If $q_1(x(\tau)) = 0$, in this case it means we are in case (3.46) and we will act in the same way.

Repeating this procedure, we see that, as long as the measurements received by the controller are positive, the upper bounds on the norm of the estimation error $\|\hat{x} - x\|_\infty$ at the sampling time form a geometric progression with ratio $\Lambda/(N^{1/n})$. This leads to

***Assumption* 4** *We have*

$$\Lambda < N^{1/n}.$$

In view of the definition of Λ, this inequality characterizes the trade-off between the amount of information provided by the encoder at each sampling time and the required sampling frequency. This relationship depends explicitly on the Lipschitz constant L which, as we will see, can be interpreted as a measure of expansiveness of (3.11). We are now ready to prove the main result about quantized feedback.

Main Result

Before proceeding we need another assumption, which poses a lower bound on the prior knowledge we have of the system. Let $E_{\Delta\mu} = N^{1/n}\Delta\mu$, then

***Assumption* 5** $E_0 > E_{\Delta\mu}$.

Theorem 2 *Under Assumptions 2-5, the control law described in the previous section semi-globally contains (semiglobally asymptotically stabilizes) the system (3.11) with $\nu \neq \mu$ ($\nu = \mu$).*

Proof *We first show that $\|x(t)\|_\infty < D$ and $\|\hat{x}(t)\|_\infty < D$ for all $t \geq 0$, where D is defined by (3.41). Suppose that this is not true. Then, since x is continuous with $\|x(0)\| \leq \Lambda E_0 < D$ and $\hat{x}$ is continuous from the right with $\hat{x}(0) \leq \Lambda E_0 < D$, the time*

$$\bar{t} := \min\{t > 0 : \max\{\|x(t)\|_\infty, \|\hat{x}(t)\|_\infty\} \geqslant D\} \tag{3.44}$$

is well defined. So, for all time $t \in [0, \bar{t})$, formulas (3.43) and the property of k_ν imply that (x, u) and $(\hat{x}, u)$ stay inside region (3.40) . Let us define the estimation error

$$e := \hat{x} - x. \tag{3.45}$$

We know that $\|e(0)\|_\infty \leq E_0 < D$. Now we combine equation (valid between sampling times)

$$\dot{e} = f(\hat{x}, k_\nu(\hat{x}), \nu) - f(x, k_\nu(\hat{x}), \mu)$$

with the formula

$$\|f(\hat{x}, u, \nu) - f(x, u, \mu)\|_\infty \leqslant \mathrm{Lmax}\{\|e\|_\infty, \Delta\mu\}.$$

If $\|e(t)\|_\infty > \Delta\mu$ we can apply the Bellman-Gronwall lemma, and conclude that for every interval $(t_1, t_2) \subset [0, \bar{t})$ not containing any sampling time we have

$$\|e(t_2)\|_\infty \leqslant e^{L(t_2-t_1)}\|e(t_1)\|_\infty \leqslant \Lambda\|e(t_1)\|_\infty \quad (\ \|e^-(l)\|_\infty \leq \Lambda\|e(l-1)\|_\infty\ )$$

where the last inequality follows from (3.42). If $\|e(t)\|_\infty < \Delta\mu$

$$\|e(t_2)\|_\infty \leqslant \|e(t_1)\|_\infty + L\Delta\mu(t_2 - t_1) \leqslant \Lambda\|e(t_1)\|_\infty \quad \Leftrightarrow \quad \|e(t_1)\|_\infty \geqslant \frac{L(t_2 - t_1)}{(\Lambda - 1)}\Delta\mu$$

$$\Big(\ \|e^-(l)\|_\infty \leq \Lambda\Delta\mu \leq \Lambda\|e(l-1)\|_\infty \quad \Leftrightarrow \quad \|e(l-1)\|_\infty \geq \Delta\mu\Big)$$

This in turn guarantees that at each sampling time $k\tau \in [0, \bar{t})$*, we have*

$$\frac{\Lambda^k}{N^{\frac{k}{n}}} E_0 = E_{k-1} \geqslant \frac{\log(\Lambda)}{(\Lambda - 1)}\Delta\mu \quad \Rightarrow \quad q_k(x(k\tau)) > 0 \tag{3.46}$$

$$\left(\ \frac{\Lambda^l}{N^{\frac{l}{n}}} E_0 = E_{l-1} \geq \Delta\mu \quad \Rightarrow \quad q_l(x(l)) > 0\ \right) \tag{3.47}$$

If $E_{k-1} < \frac{\log(\Lambda)}{(\Lambda-1)}\Delta\mu$*, we may get* $q_k(x(k\tau)) = 0$*. In this case we just set* $\hat{x}(k\tau) = \hat{x}(k\tau^-)$ *with error* $E_k = E_{\Delta\mu} > \Lambda\frac{\log(\Lambda)}{(\Lambda-1)}\Delta\mu$*, where the last inequality comes from Assumptions 4-5. This way we satisfy condition (3.46), and we know that* $x(k\tau) \in B^n_\infty(\hat{x}(k\tau), E_k)$*. In any case, the upper bound on* $\|e\|_\infty$ *is divided by* $N^{1/n}$ *until condition (3.46) is satisfied. We thus can conclude*

$$\exists t_1 = \bar{k}\tau : \ \|e(t)\| < \Lambda E_{\Delta\mu}, \ \forall t \geqslant t_1$$

Invoking Assumptions 4-5, we arrive at the bound $\|e(t)\| < \Lambda E_0$ *for all* $t \in [0, \bar{t})$*. If* $\bar{t}$ *is not a sampling time, then* e *is continuous in* $\bar{t}$*; if* $\bar{t}$ *is a sampling time then* e *can only decrease in* $\bar{t}$*. In either case, we actually have*

$$\|e(t)\|_\infty < \Lambda E_0, \ \forall t \in [0, \bar{t}]. \tag{3.48}$$

Now, Assumption 2 expressed by (3.2.5) with $t_0 = 0$ *implies that* $\|x(t)\|_\infty \leqslant \beta_{\mu\nu}(E_0, 0) + \gamma_{\mu\nu}(\Lambda E_0) + r_{\mu\nu} < D$ *for all* $t \in [0, \bar{t}]$*, where the last inequality follows from Assumption 3. Using (3.45) and (3.48), we also obtain* $\|\hat{x}(t)\|_\infty \leqslant \beta_{\mu\nu}(E_0, 0) + \gamma_{\mu\nu}(\Lambda E_0) + \Lambda E_0 + r_{\mu\nu} < D$ *for all* $t \in [0, \bar{t}]$*. This yields a contradiction with the definition (3.44) of* $\bar{t}$*.*

We have thus established that all of the previous estimates are valid with $\bar{t} = \infty$*. In particular, by Assumption 4, we have that* $\|e(t)\|_\infty$ *decreases at each sampling time as long as* $\|e(t)\|_\infty \geqslant \frac{\log(\Lambda)}{(\Lambda-1)}\Delta\mu$*. Thus, remembering the procedure for* $q_k = 0$*, the system is governed by (3.37) with* $\|e\|_\infty < \Lambda E_{\Delta\mu}$*, which, for the ISC property of this system, implies* $\|x(t)\|_\infty \leqslant \beta_{\mu\nu}(E_0, t) + \gamma_{\mu\nu}(\Lambda E_{\Delta\mu}) + r_{\mu\nu}$*, and, for large times* $\|x(t)\|_\infty \leqslant \gamma_{\mu\nu}(\Lambda E_{\Delta\mu}) + r_{\mu\nu}$*, which ends the proof for* $\nu \neq \mu$*.*

If $\nu = \mu$*, then* $\|e(t)\|_\infty \longrightarrow 0$ *as* $t \longrightarrow \infty$*, and , for the ISS property of this system with respect to* e *we conclude that* x *converges to 0 as well. It remains to prove that the origin is a stable equilibrium. For this proof see ([2])* □

Thus, once we know everything about the system, we are able to build a digital controller, which uses as low information as permitted by the expansiveness of the system, and which stabilizes it.
In the case we have uncertainties about the parameters, we can take the system contained around the origin and have all the time to learn its inverse dynamics. In particular, with the notation of equation (3.33), we have

$$\|x(t)\| \leqslant \gamma_{\mu\nu}(\Lambda E_{\Delta\mu}) + r_{\mu\nu}$$

for both the discrete and the continuous time case.
To be sure that our approximation hold all the time, we should ask that the parameters of the system remain unchanged enough time to be learned, that is

***Assumption* 6** *The rate of change in the parameter R_p is much lower than minimum between the rate of the inverse dynamic learning R_ℓ and the inverse of the delay time T for the communication to the learning apparatus:*

$$R_p \ll \frac{1}{2}\min\{R_l, T^{-1}\},$$

that is, for each parameter change, the adaptive controller has the time to stabilize the system (i.e, learn the inverse dynamic).

It's worth to say that the hypothesis of ISS can be relaxed (see [2] and [11]), using a larger number of bits.

3.2.6 Practical estimation of containability borders

In the case we want to estimate the subset in which the system will be contained we can make use of the *Practical Stability theory.*
The notion of practical stability in dynamical system was discussed by Lasalle and Lefshetz ([7]) in the 1960s and then was treated by Liao ([8]). Recently, this concept has been renewed by Kapitaniak and Brindley ([10]) in dealing with stability of chaotic attractors and synchronization.
We recall here the classical practical stability of equilibrium points of dynamical systems. Consider a dynamical system described by

$$\dot{x} = g(x,t) \tag{3.49}$$

where $g(0,t) \equiv 0, g \in C^r[B \times \Re^+, \Re^n]$, and B is a region in $\Re^n$ containing $x = 0$. Let the system (3.49) be under influence of a permanently acting perturbation $p(x,t)$ with $\|p(x,t)\| \leqslant \delta$ so that the perturbed system is

$$\dot{x} = g(x,t) + p(x,t) \tag{3.50}$$

Definition 2 *Given a positive number δ and two sets Q and Q_0 such that $0 \in Q_0, Q_0 \subset Q \subseteq B$ and Q is bounded. If every solution $x(t_0, x_0, t)$ to Eq. (3.50) satisfies $x(x_0, t_0, t) \subset Q$ for $\forall x_0 \in Q_0$ and $\forall p(x,t)$ with $\|p(x,t)\| \leqslant \delta$, then the equilibrium solution $x = 0$ to Eq. (3.49) is called practically stable with respect to δ, Q, Q_0.*

The following theorem can be used for the estimation problem

Theorem 3 *Suppose that $Q_0 \subset \Re^n$ is a compact set containing $0 \in \Re^n$. If there exists a function $V(x,t) \in C^r[\Re^n \times \Re^+, \Re]$ such that*

$$D^+ V(x,t)|_{(3.50)} \leqslant 0 \quad \forall x \in Q_0^c$$

and

$$V(x_1, t_1) < V(x_2, t_2) \quad \forall x_1 \in Q_0, \quad \forall x_2 \in Q^c, \quad t_2 \geqslant t_1 \geqslant 0,$$

where D^+ denotes the Dini derivatives. Then every solution $x(x_0, t_0, t)$ to Eq. (3.12) is contained in Q for $t \geqslant t_0$ thus the equilibrium solution $x = 0$ to Eq. (3.49) is practically stable with respect to δ, Q, Q_0.

In fact, we can scalar multiply the vectorial control perturbation (3.31) due to measurement errors and parameter uncertainties and the general input perturbation p, with the vector function

$$\int_0^1 \left[\frac{\partial f(x,y,\mu)}{\partial y} \right]_{y = k_\mu(x) + \alpha p_i(x)} d\alpha$$

bounded for the Lipschitz property of f, where i run over the kind of perturbation, obtaining the general form

$$P_{e,\mu\nu}(x) + P_{\mu\nu,e}(x) + P(x) \tag{3.51}$$

which can be added explicitly in the system(3.11), and allows us to use the last theorem. The existence of a Lyapunov function is assured by the ISS assumption (see Section 3.1)

Containability border of the quantized digital controller

We want to find the largest set of parameters which can guarantee that, as long as the parameter vector changes remain in this set, the system remains containable. Thus, we are looking for the largest convex subset $\Omega_0 \subset \Re^p$, such that, for a fixed trial parameter vector, ν, the system is contained for all real parameter vectors μ. Ω_0 represents how sharp our knowledge of the system's parameters is.
Conditions of Theorem3 assure that, for any initial condition $x(0) \in \Re^n$, there

exists t_0 such that $x(t) \in Q$ for $\forall t > t_0$. We can apply this result to estimating the maximum size of Ω_0, such that the system is bounded in a determined region of the state space we want it to.
First, we define such a region as a ball of acceptable states

$$x(t), \hat{x}(t) \in B^n_\infty(0, R) \quad \forall t.$$

In this way the Lipschitz constant is well defined in the state and in the control spaces, and we can study its dependence on Ω_0, on the contrary to what we did in the last section, where the real and trial parameter vectors were fixed.
Then we write the correct form of the perturbation (3.51). In the case of the quantized digital controller, $P(x) = 0$ and $e = \hat{x} - x$, so, for a fixed μ, ν couple, the perturbation is bounded by

$$\delta^{\|e\|}_{\mu\nu}(x) = \max_{x' : \|x'-x\| = \|e\|} \|P_{e',\mu\nu}(x) + P_{\mu\nu,e'}(x)\| , \tag{3.52}$$

where $e' = x' - x$. Here, e is a dynamical variable, but we can put an upper bound on its norm. In fact, we can leave Assumption 5, and define $E_{d(\Omega_0)} = N^{1/n} d(\Omega_0)$. Assumption 5 relates the size of the initial condition and the distance between the real and the actual parameter vector and it was necessary to make sure that, when we get a zero from the encoder, the state and its estimate remain inside region (3.40), which was implicitly defined through comparison functions, and thus be able to well define the Lipschitz constant, while now it is evaluated over region (3.54). In any case, we will need a similar relation, but this time we will consider the diameter of the full parameter set, to be sure the state and its estimate remain inside this region (see Assumption 7).
Then, it results:

$$\|e(t)\|_\infty < \overline{E} = \max\left\{ \frac{\Lambda^2}{N^{1/n}} E_0, \Lambda E_{d(\Omega_0)} \right\}, \quad \Lambda = e^{\tau L(\Omega_{\Delta\mu})}, \tag{3.53}$$

where $\Omega_{\Delta\mu}$ is the smallest convex subset of Ω_0 which contains μ and ν, and $L(\Omega_{\Delta\mu})$ is the Lipschitz constant of the region

$$\{(x, u) : \|x\|_\infty \leqslant R, \|u\|_\infty \leqslant \kappa_\nu(R)\} \times \Omega_{\Delta\mu}, \tag{3.54}$$

κ_ν defined as in (3.40). Note that we have to consider $E_{d(\Omega_0)}$ instead of $E_{\Delta\mu}$ because, when we control the system we don't know the real difference in the parameters and we have to use its upper bound, to define the reset error when we get a zero from the encoder.
Now, we can compute $\delta^{\overline{E}}_{\mu\nu}(x)$ (actually the typical error to be considered, in the computation of delta only, is smaller as its norm has exponential decays to zero,

except the case in which we get a zero from the encoder, see Appendix A for details), for and find the relative $Q_{\mu\nu}$ which can satisfy all the assumptions of Theorem 2, with $Q = Q_0 = Q_{\mu\nu}$. If $Q_{\mu\nu}$ is such that $\|x\|_\infty \leq R$ and $\|\hat{x}\|_\infty \leq R$ for all time, then we can follow the control algorithm explained in section 3.2.5, until we have $e \leqslant \Lambda E_{d(\Omega_0)}$ and $Q_{\mu\nu}(\Lambda E_{d(\Omega_0)})$. Comparing with the proof of Th. 1, we conclude

$$d(Q_{\mu\nu}(\Lambda E_{d(\Omega_0)})) = \gamma_{\mu\nu}(\Lambda E_{d(\Omega_0)}) + r_{\mu\nu}, \quad r_{\mu\nu} = d(Q_{\mu\nu}(0)).$$

In the case $Q_{\mu\nu}$ is too big, for some $\mu \in \Omega_0$, it means we are dealing with too big a difference in the parameter and the system is not containable with the present knowledge of them, or that we are dealing with too big a set of initial condition. In either case we have too low a prior knowledge of the system.
In real cases, we generally can't control initial condition better than a certain range (due the low bandwidth), but we can do some failing trials with the system to sharpen our knowledge of its parameters. We thus have to fix, for example,

$$E_0 = \frac{R}{N},$$

where we decided to divide the set of possible state in N concentrically hypercubic boxes with diameter $2R/N, 4R/N, \ldots, 2R$, assign a symbol to each element of the associated partition, and start the system so that we get the symbol associated to the smallest of them, that is $x(0^-) \in B^n_\infty(0, E_0)$ and $x(0) \in B^n_\infty(0, \Lambda E_0)$. Once the set of initial conditions is fixed, we have to find the largest $\Omega_0 \subset \Re^p$, for which

$$d(Q_{\mu\nu}) < R - \overline{E}, \quad \forall \mu \in \Omega_0, \tag{3.55}$$

that assures that both $x(t)$ and $\hat{x}(t)$ remain inside $B^n_\infty(0, R)$ for all time, where ν is the initial trial parameter vector ($\nu \in \Omega_0$). Thus we have

$$\|x(t)\|_\infty < \max\{R - \overline{E}, \Lambda E_0\} \quad \forall t.$$

Note that, if $\Lambda E_0 > R - \overline{E}$, we have to impose that $\overline{E} < R - \Lambda E_0$ to be sure $\hat{x}$ remains in $B^n_\infty(0, R)$, which is in contradiction with $\Lambda E_0 > R - \overline{E}$. Thus we have to impose $R - \overline{E} > \Lambda E_0$, which leads to

Assumption 7 *a)* $N > 2\Lambda$, $\Lambda = e^{\tau L(\Omega_0)}$, *if* $\Lambda E_{d(\Omega_0)} < \Lambda^2 E_0 / N^{1/n}$

b) $N^{1/n} d(\Omega_0) < R(N - 2\Lambda)$, $\Lambda = e^{\tau L(\Omega_0)}$, *if* $\Lambda E_{d(\Omega_0)} > \Lambda^2 E_0 / N^{1/n}$

Assumption 7(a) *can* be more strict than Assumption 4 . Assumption 7(b) is a lower bound on the prior knowledge we have of the system. We are now ready for the following

Proposition 1 *Suppose $\mu \in \Omega_0$. Then $x(t)$ and $\hat{x}(t)$ belong to $B^n_\infty(0, R)$ for $\forall t$*

Proof The following relations hold for $\forall t$:

a) $\|x\|_\infty < R - \overline{E} < R$

b) $\|e\|_\infty \leqslant \overline{E}$

c) $\|\hat{x}\|_\infty \leqslant \|x\| + \|e\| < R$

□

We now invoke Assumption 6 . After the first parameter vector, μ has been learned, it can be possible to enlarge the allowed parameters set from Ω_0 to Ω, $\Omega_0 \subset \Omega$. In fact, once the inverse dynamic is learned, the system is stabilized and the error e=0. Then, when the parameter changes, we have $E_0 = 0$, so that

$$\|e\| \leqslant \Lambda E_{d(\Omega)}, \quad \Lambda = \Lambda(L(\Omega_{\Delta\mu})).$$

Thus we can leave Assumption 7 and we can substitute condition (3.55) with the new one (which imply the old one)

$$d(Q_{\nu'\mu}) < R - \Lambda E_{d(\Omega)}, \quad \forall \nu' \in \Omega. \tag{3.56}$$

In the next section we will show how to study the possible parameter set for a simple mechanical system, the inverted pendulum on a cart, controlled with minimal bit rate.

3.2.7 The inverted pendulum on a cart

We suppose that the only unknown parameter is the length of an inverted pendulum on a cart. Our aim is to find the largest $\Omega_0 \subset \Re$ which can contain the real length parameter $l \in \Re_+$, for which 3.55 holds for a fixed trial length parameter $\nu \in \Re_+$. Actually, Ω_0 will be the union of two sets, one that has ν as its min and one as its max, and for which 3.55 holds independently. This way the embedded digital controller with minimal bit rate suffices to contain the system, while the learning apparatus has all the time to learn the inverse dynamics and find the correct length parameter (as explained in figure 1.11).
First we have to find an ISS feedback law for the system. The solution is given in [12]. By choosing the closed-loop Lagrangian from a class of *controlled Lagrangian*, closed-loop dynamic is guaranteed to be Lagrangian. This has the advantage that the stabilization can be understood in terms of energy. In particular, we can make use of energy methods which provide a Lyapunov function which gives information on how to choose the control gains to achieve closed-loop stability. Moreover

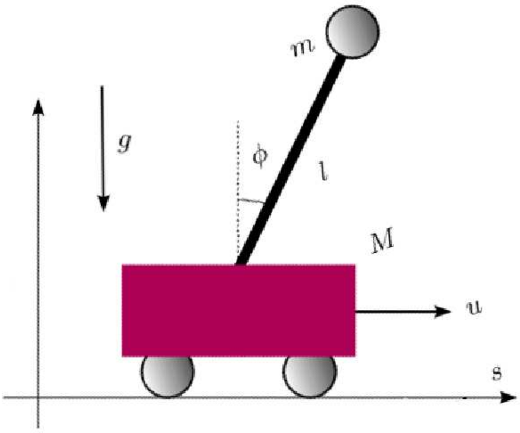


Figure 3.1:

we can calculate its time derivative along non perturbed and perturbed system trajectories to apply results of Section 3.2.6 and find the containability borders. First we analyze the system by choosing an arbitrary length, let's say $l = 1.5$m

The Lagrangian: Let s denote the position of the cart on the horizontal axis and let ϕ denote the angle of the pendulum with up-right vertical (see figure 3.1. Let $M = 3$kg be the cart mass, $m = 1.5$kg the pendulum bob mass, $g = 9.8\text{ms}^{-2}$ the intensity of the gravitational force field. Then

$$L(\phi, s, \dot{\phi}, \dot{s}) = \frac{1}{2}(\alpha\dot{\phi}^2 + 2\beta\cos(\phi)\dot{s}\dot{\phi} + \gamma\dot{s}^2) + \text{Dcos}(\phi), \tag{3.57}$$

where $\alpha = ml^2$, $\beta = ml$, $\gamma = M + m$, and $D = -mgl$ are constants. The relative equilibrium defined by $\phi = \dot{\phi} = \dot{s} = 0$ is unstable since $D < 0$.
Equations of motion: The equations of motion for the pendulum on a cart with control force u acting on the cart (and no direct force acting on the pendulum) are, since s is a cyclic variable

$$\frac{d}{dt}\frac{\partial L}{\partial\dot{\phi}} - \frac{\partial L}{\partial\phi} = 0,$$

$$\frac{d}{dt}\frac{\partial L}{\partial\dot{s}} = u,$$

which leads, in the cart reference system, setting $\phi = x$, to

$$\dot{x} = y$$

$$\dot{y} = \frac{1}{l}\frac{(m+M)}{(m\sin(x)^2+M)}\left(g\sin(x) - \frac{(ml)^2}{(m+M)}\cos(x)\sin(x)y^2 - \frac{\cos(x)}{(m+M)}u\right).$$

The relative nonlinear control law, which renders the closed-loop system still Lagrangian is

$$u(x,y) = k\frac{ml\sin(x)(ml^2y^2+\cos(x)D)}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)} +$$
$$-\frac{c\left(ml^2 - \frac{ml}{m+M}\cos^2(x)\right)((r(k+1)+1)ml\cos(x)y)}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)},$$

and the relative closed-loop Lyapunov function (figure 3.2) is

$$\mathcal{L}(x,y) = mgl - \left(\frac{1}{2}\left(ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)\right)y^2 + \right.$$
$$\left. + mgl\cos(x) - \frac{1}{2}(m+M)r(k+1)ml\cos(x)y^2\right),$$

where $k > M/m$, $c > 0$, $r > 0$. Note that the control tends to diverge for $x \to \text{acos}\left(\sqrt{\frac{m+M}{m(1+K)}}\right)$. We choose $R = 1$, thus the ball of admissible state is $B^2_\infty(0,1)$. If we want the control not to diverge when $|x|$ approach 1, we have to impose

$$1 + k > \frac{m+M}{m\cos^2(1)}.$$

Now we can start to look at the Lyapunov function time derivative. From here on we will write, in accordance to the last section, $l = \mu$. Being $\dot{x}$ independent of u, the perturbation will act only on $\dot{y}$. Keeping this in mind, we calculate the time derivative of the Lyapunov function along controlled system trajectories, the Dini derivative, in both unperturbed and perturbed case, by

$$\frac{d\mathcal{L}}{dt} = \nabla\mathcal{L}\cdot(f+p) = \nabla\mathcal{L}\cdot f + \frac{\partial\mathcal{L}}{\partial y}p \leq \nabla\mathcal{L}\cdot f + \left|\frac{\partial\mathcal{L}}{\partial y}\right||p|.$$

In figures 3.3-3.6, we plot this value for a constant divergent perturbation force field, $\mathcal{F}(x,y) = (0, p\cdot\text{sign}(y))$, and for various values of p. In the perturbed case, $d\mathcal{L}/dt$ will be negative outside the containability region, and positive inside it. Hence, its zeros define the containability borders. That is why we also plot its absolute value, for clarity of vision, inside the dark blue line it is positive, outside negative. Unfortunately, looking just at $d\mathcal{L}/dt$, it seems that the system is not containable along x. The relative unverified ISS assumption is condition 3.4 with ξ belonging to x-axis. In fact, $\frac{d\mathcal{L}}{dt}\Big|_{y=0} \equiv 0$, $\forall x$, as the trajectories of the system are orthogonal to the x-axis.

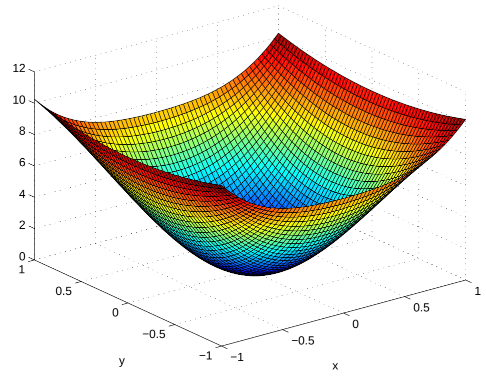


Figure 3.2: Lyapunov function for the closed loop System

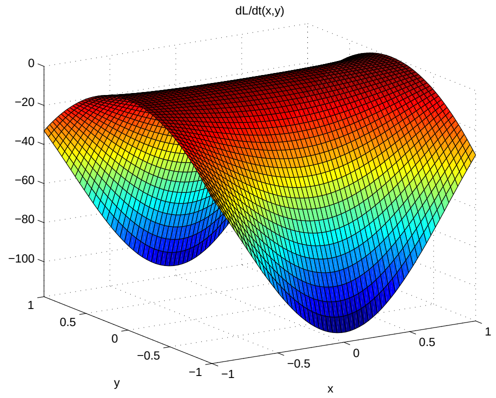


Figure 3.3: Dini derivative of the Lyapunov function in non perturbed case

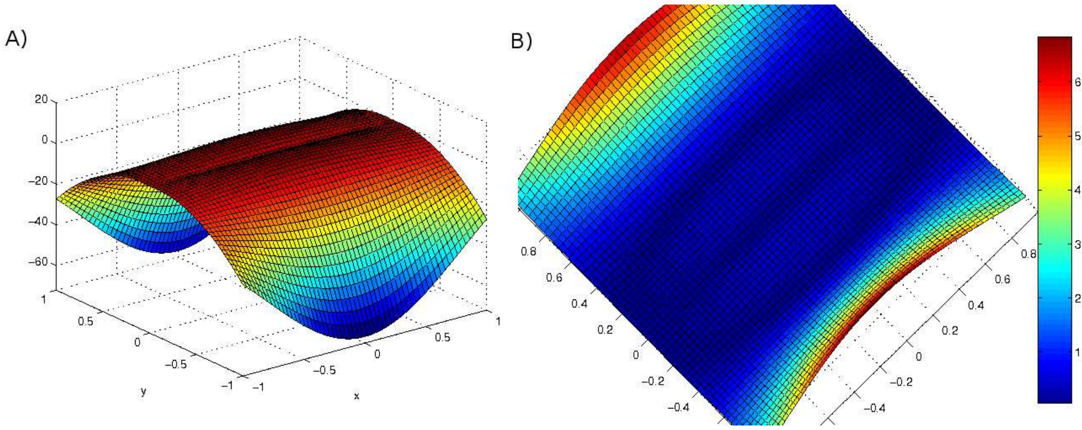


Figure 3.4: Dini derivative of the Lyapunov function, $p = 3$

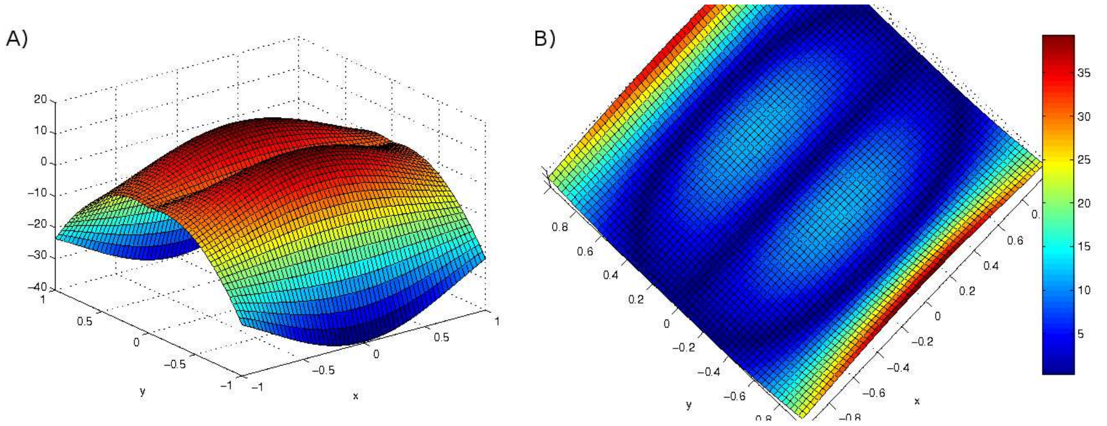


Figure 3.5: Dini derivative of the Lyapunov function, $p = 5$

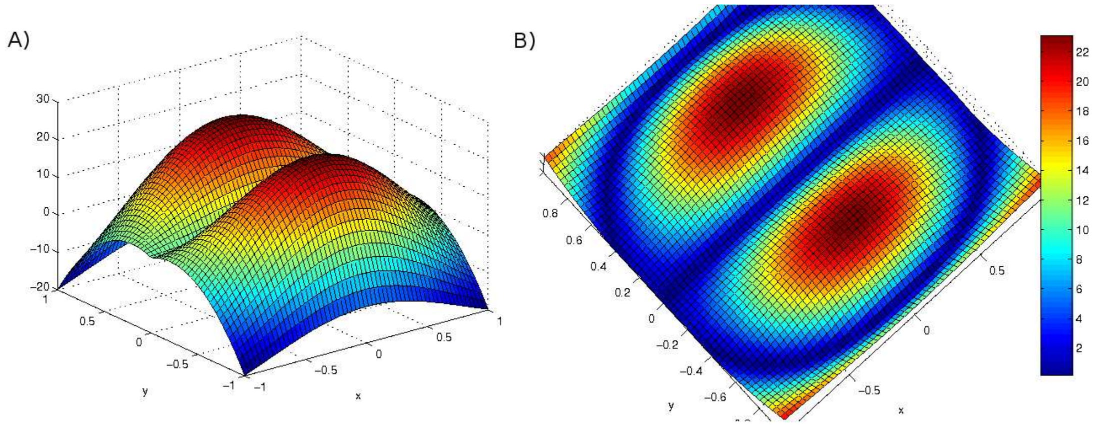


Figure 3.6: Dini derivative of the Lyapunov function, $p = 7$

Defining the perturbation as in (3.52), and plotting its absolute value in function of x and y for this system for $\mu = 1.5$, $\nu = 1.3$ (figure 3.7), where the error has been calculated as in Appendix A, we see that its values tend to grow very fast only near $|y| \sim 1$, where, fortunately, the Dini derivative of the Lyapunov function is very negative. In the rest of the admissible phase space its values remain small, it is almost constant along the x direction, monotonically decreasing along the negative y axis, and monotonically increasing along the positive y axis. Moreover, remember that the trajectories are orthogonal to the $y = 0$ axis, which means that the system cannot diverge along x directly but will always first explore the $|y| > 0$ region. Hence, we can try to apply the procedure explained in section 3.2.6, but limiting our attention to the y coordinate. Then, by adding a constant divergent field, as in figures 3.3-3.6, we see that the zeros of the Dini derivative of the Lyapunov function which tend to reach the admissible region borders along y, e.g $|y| = 1$, faster as the perturbation intensity increases are those along $x = 0$. We want to use these facts to write an algorithm which lets us estimate the containability region for a fixed couple of real and trial parameter vectors. The idea is to find the perturbation intensity p for a constant divergent force field along the y direction, $\mathcal{F}(x, y) = (0, p \cdot \text{sign}(y))$ for which condition (3.55) is satisfied along y. Then find the largest value, $\overline{y}$, for which the real perturbation field, $\delta^{||e||}_{\mu\nu}$ (3.52), intensity stays under this value in the region $x = 0$, $y \in (-\overline{y}, \overline{y})$. We want $\overline{y}$ to be large, so that the perturbation in the region explored by the system remains small. When the difference between the real and the trial parameter increases, $\overline{y}$ will tend to diminish until it will be smaller than $1 - \overline{E}$, which means that the perturbation is large in this region too and the system will tend to leave it.

Keeping this in mind, this is the proposed algorithm:

1. choose the set of fix parameter k, c, r (in our case $k = 15, c = 100, r = 0.00001$)

2. choose a trial length parameter ν (in our simulation $\nu = 1.25$m), a length parameter axis resolution $\delta\mu$ to choose successive possible real parameters, and set $\mu = \nu + \delta\mu$ (in our simulation $\delta\mu = 0.01$)

3. calculate $\Lambda = e^{L\tau}$, where $\tau = 0.005$, $L = L(B^2_\infty(0, 1), [\nu, \mu]) = L(\nu, \mu)$ (the Lipschitz constant of f over the region $B^2_\infty(0, 1) \times [\nu, \mu]$), and the relative N

4. calculate $\overline{E}$ as in (3.53)

5. solve $\left.\left(\frac{d\mathcal{L}}{dt} + h \cdot \left|\frac{\partial \mathcal{L}}{\partial y}\right|\right)\right|_{x=0, y=1-\overline{E}, \mu} = 0$ with respect to h, which gives us an estimate of the perturbation intensity of a constant divergent force field for which (3.55) is satisfied along y

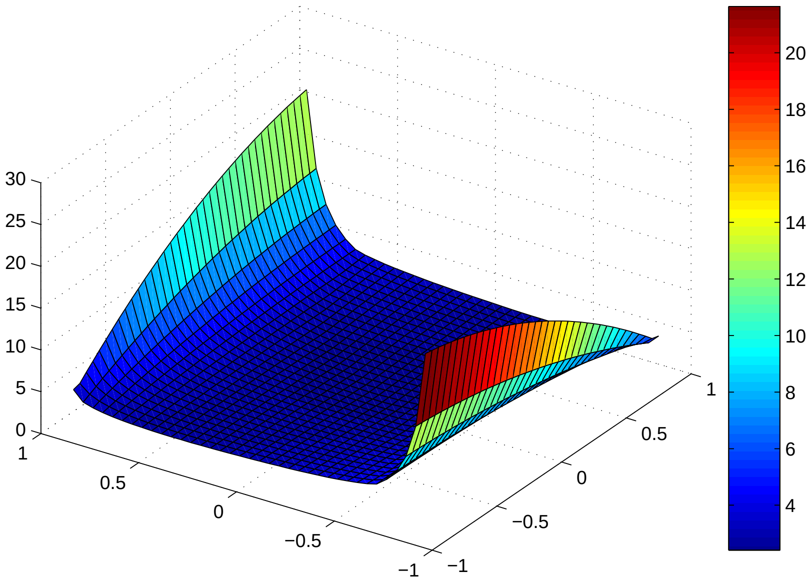


Figure 3.7: Absolute perturbation intensity

6. starting from $y = 1$, find the largest $\overline{y}$ for which $\delta_{\mu\nu}^{\|e\|}(0, \overline{y}) < h$, where $\delta_{\mu\nu}^{\|e\|}$ is defined by (3.52) and the error is approximated as the typical error in Appendix A.

7. if $(\overline{y} > 1 - \overline{E})$ and Assumption 7 is verified $\Rightarrow$ $\mu \to \mu + \delta\mu$ and go back to 3.; else set $\mu - \delta\mu = \mu_{\max}$ and $\Omega_0' = [\nu, \mu_{\max}]$ and proceed to 8.

8. set $\mu = \nu - \delta\mu$

9. calculate $\Lambda = e^{L\tau}$, where $\tau = 0.005$, $L = L(B_\infty^2(0,1), [\mu, \nu]) = L(\mu, \nu)$ (the Lipschitz constant of f over the region $B_\infty^2(0,1) \times [\nu, \mu]$), and the relative N

10. calculate $\overline{E}$ as in (3.53)

11. solve $\left(\frac{d\mathcal{L}}{dt} + h \cdot \left| \frac{\partial \mathcal{L}}{\partial y} \right| \right) \Big|_{x=0, y=1-\overline{E}, \mu} = 0$ with respect to h, which give us an estimate of the perturbation intensity of a constant divergent force field for which (3.55) is satisfied along y

12. starting from $y = 1$, find the largest $\overline{y}$ for which $\delta_{\mu\nu}^{\|e\|}(0, \overline{y}) < h$, where $\delta_{\mu\nu}^{\|e\|}$ is defined by (3.52) and the error is approximated as the typical error in Appendix A.

13. if ($\overline{y} > 1 - \overline{E}$) and Assumption 7 is verified $\Rightarrow$ $\mu \rightarrow \mu - \delta\mu$ and go back to 9., else set $\mu + \delta\mu = \mu_{\min}$ and $\Omega_0'' = [\mu_{\min}, \nu]$

14. to let 3.55 hold in Ω_0' and Ω_0'' at the same time they must have the same diameter, so that the encoder can us the same $d(\Omega_0') = d(\Omega_0'')$ when it resets the quantization error after a zero symbol, and it does not need to know in which of the two set is the real parameter. So, reduce the largest set between Ω_0' and Ω_0'', by diminishing $\mu_{\max}$ (if $d(\Omega_0') > d(\Omega_0'')$) or increasing $\mu_{\min}$ (if $d(\Omega_0') < d(\Omega_0'')$), until the two sets have the same diameter.

15. $\Omega_0 = \Omega_0' \cup \Omega_0''$

These are the results for this algorithm applied to our the inverted pendulum: trial parameter $\nu = 1.25$, $\mu_{\max} = 1.49$, $\mu_{\min} = 1.05$. We reduce $\mu_{\max}$ to 1.45, and get $\Omega_0 = [1.05, 1.45]$, with $d(\Omega_0') = d(\Omega_0'') = 0.2$ the allowed parameter set's diameter used by the encoder. Other values are: $L = 70$, $\Lambda = 1.42$, $N = 4$ ($N^{1/2}$ is not an odd integer, as required by Assumption 2, but it was only for light notation purposes, as a similar demonstration can be done in the case of generic integer $N^{1/n}$). Thus if the real system parameter is in Ω_0, we can contain the system by choosing ν as the trial parameter. Moreover we can contain it by using just five symbols (∼2.322 bits of information) for the codification of the state and of the feedback control. See the figures for the results of simulation. All simulations have initial conditions $(-1/4+0.001, -1/4+0.001)$. Note that the system actually remains inside the ball $B^2_\infty(0, R - \Lambda N^{1/2} d(\Omega_0)$.

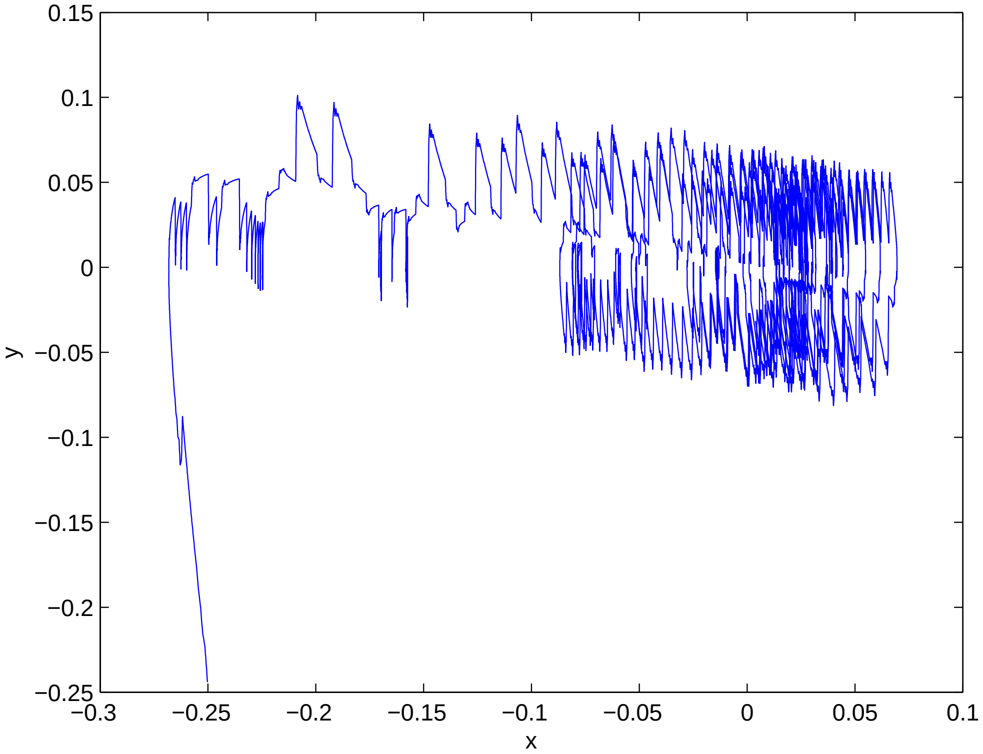


Figure 3.8: Phase space for the 1.05 m inverted pendulum, controlled with quantized feedback law (5 symbols), 5 ms sampling period, 1.25 m controller length parameter

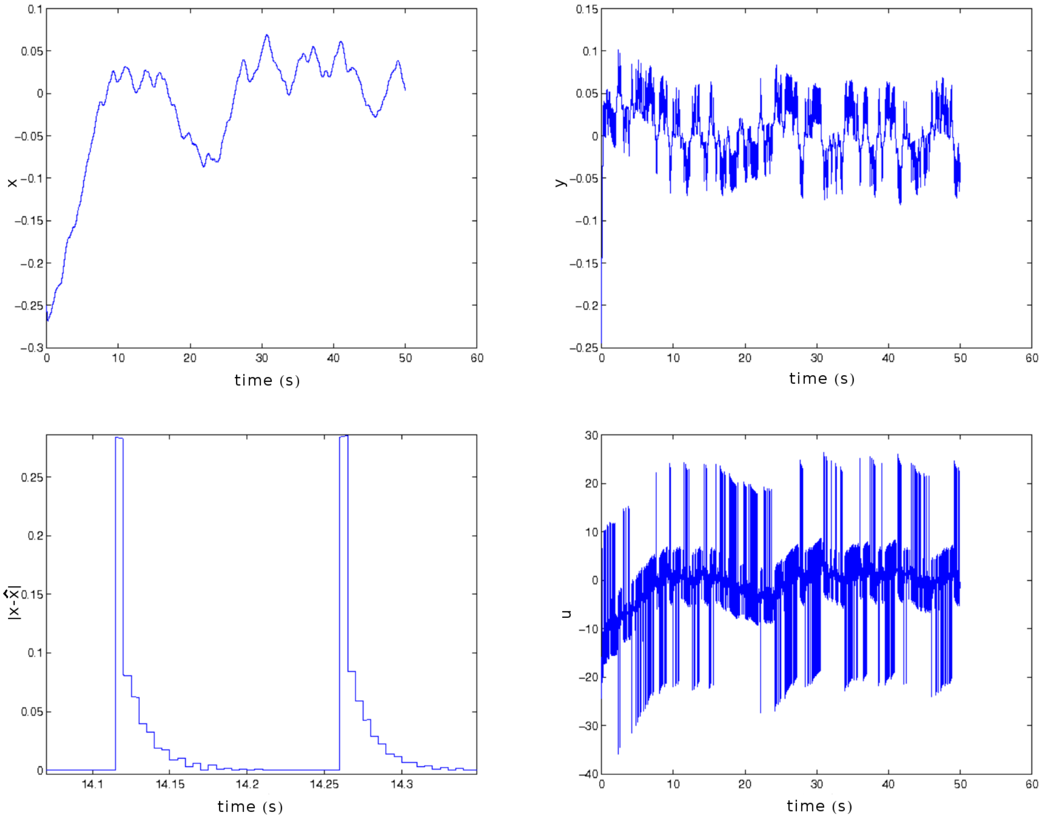


Figure 3.9: 1.05 m inverted pendulum, controlled with quantized feedback law (5 symbols), 5 ms sampling period, 1.25 m controller length parameter: Top left: Angle temporal evolution. Top right: angular speed temporal evolution. Bottom left: Absolute value of the difference between measured and actual angle temporal evolution; note the local inverse exponential behavior. Bottom right: applied control temporal evolution.

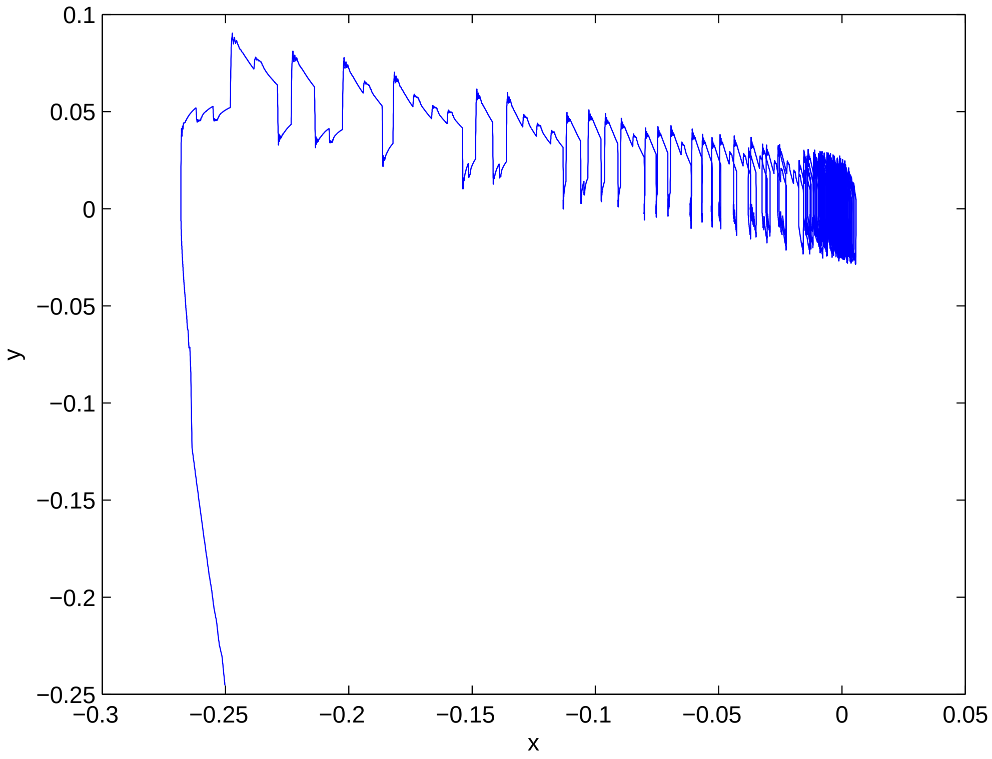


Figure 3.10: Phase space for the 1.45 m inverted pendulum, controlled with quantized feedback law (5 symbols), 5 ms sampling period, 1.25 m controller length parameter

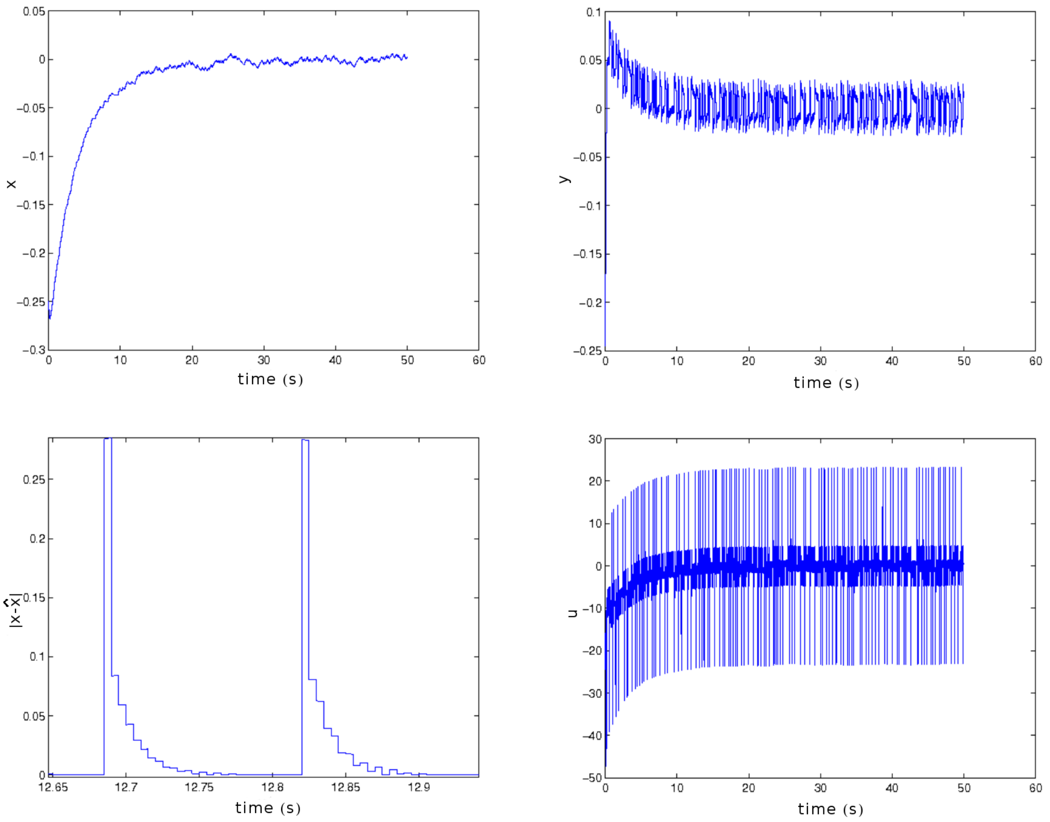


Figure 3.11: 1.45 m inverted pendulum, controlled with quantized feedback law (5 symbols), 5 ms sampling period, 1.25 m controller length parameter: Top left: Angle temporal evolution. Top right: angular speed temporal evolution. Bottom left: Absolute value of the difference between measured and actual angle temporal evolution; note the local inverse exponential behavior. Bottom right: applied control temporal evolution.

Chapter 4

Chaos skeleton

In this chapter we will introduce some general facts about deterministic chaotic systems, in particular that each chaotic dynamic can be decomposed into an infinity of periodic orbits, its *skeleton*. We will assume that the reader is familiar with the concepts of Autonomous Dynamical Systems, Autonomous Flows and Maps.
In what follows we will deal with a d-dimensional autonomous dynamical system

$$\dot{x} = f(x) \tag{4.1}$$

or, equivalently, with its flow $x(t) = F^t(x(0))$, where is F is a (locally) smooth function.
In the case of maps, $x_n = f^n(x_0)$, with f^n a (locally) smooth function for each n.

4.1 What is chaos?

The present section offers preliminary information about the dynamic of chaotic processes. For a more detailed presentation we refer the reader to ([71]). The chaotic systems represent a class of indeterminacy models differing from the stochastic models. Whereas with a knowledge of the current system state the deterministic model can predict the future trajectory for an arbitrarily long period and the stochastic model cannot make a precise forecast, generally speaking, even for an arbitrarily short time, the forecast error of the chaotic model grows exponentially and, consequently, a forecast can be made only for a limited time defined by the admissible forecast error.
In order to define chaotic systems, we have to introduce the concept of *attractor*.

Definition 3 *A closed set $W \subset \Re^n$ is called an attractor of the system (4.1) if (a) there exists an open set $W_0 \supset W$ such that all the trajectories $x(t)$ beginning* *in W_0 are definite for $t \geq 0$ and $\|x(t)\|_W \to 0$ for $t \to \infty$, where the point-to-set distance, $\| \cdot \|_W$, is defined in section 3.1.1, (b) no other eigensubset of W has this property.*

Now we can define the chaotic property for an attractor. There are many ways to do this, and we will just use the simpler, even if it is not the more precise (in particular, chaos implies this definition, but not vice-versa, see [71] for more details).

Definition 4 *An attractor is said to be chaotic if any two nearby trajectories starting in it separate exponentially. That is the norm of every infinitesimal displacement $\delta x(0)$, belonging to a chaotic attractor and transported by the flow F^t, grows exponentially:*

$$\|\delta x(t)\| = e^{\lambda t} \|\delta x(0)\|, \quad \lambda > 0$$

The dynamical system (4.1) is said to be chaotic if it has at least one chaotic attractor.

Thanks to this definition it is very easy to understand why the forecast error for a chaotic system grows exponentially: for any finite accuracy in the initial data δx the dynamics is predictable, up to an accuracy L, only for the finite time $T_{\text{Lyap}} \approx -\frac{1}{\lambda} \ln |\delta x/L|$.

Contrarily, chaotic systems have very nice properties, too: *recurrence of the trajectories* and *ergodicity on their chaotic attractors.*

Definition 5 *The function $x : \Re \to \Re^n$ is called recurrent if for any $\epsilon > 0$ there exists $T_\epsilon > 0$ such that for any $t \geq 0$ there exists $T(t, \epsilon)$, $0 < T(t, \epsilon) < T_\epsilon$, such that $\|x(t + T(t, \epsilon)) - x(t)\| < \epsilon$.*

That is with time any trajectory hits an arbitrarily small neighborhood of its position in the past.

We already defined the concept of invariant subset in section 3.1.2. We now define the restriction of the dynamical system (4.1) to an invariant closed set W as follows.

Definition 6 *The restriction of the dynamical system (4.1) to the invariant closed set W is the dynamical system*

$$\dot{x} = f_W(x) = f(x), \quad \forall x \in W, \quad x(0) \in W$$

.

Definition 7 *The dynamical system (4.1) is said to be ergodic on W if the only invariant set for its restriction to W is W, apart from sets of zero measure, with respect to some invariant measure m, where an invariant measure is such that given a set U, $m(U) = m(F^t(U))$, $\forall t$.*

Sets of zero measure can be unstable fixed points or manifold, for example. We refer the reader to ([71]) for more details. In any case, sets of zero measure are in practice never visited by the natural evolution of the system. What is important is that ergodicity implies *topological transitivity*:

Definition 8 *A continuous flow $x(t) = F^t(x(0))$ is said to be topological transitive on a closed set W if for any two subset U, V of W there exists $t \geq 0$ such that $F^t(U) \cap V \neq \varnothing$*

This means that for almost every initial condition, its image transported by the ergodic flow will visit any subset belonging to W.
Recurrence of the trajectories and ergodicity are at the base of the fact that the *skeleton* of every chaotic dynamic is made of an infinity of Unstable Periodic Orbits (UPOs), that these periodic orbits are dense in the chaotic attractor and the natural evolution of the system shadows them continuously, yet never falls on it, because they are all unstable, which corresponds to the *hyperbolicity assumption.* In what follows we will give just a "physical" derivation, based on the fact that the mean value of any observable of a chaotic dynamics takes contribution only from UPOs.

4.2 Poincaré sections

The following sections in this chapter is based from ([72])

Successive trajectory intersections with a *Poincaré section*, a $(d-1)$-dimensional hypersurface or a set of hypersurfaces $\mathcal{P}$ embedded in the d-dimensional phase space $\mathcal{M}$, define the *Poincaré map* $P(x)$, a $(d-1)$-dimensional map of the form

$$x' = P(x) = F^{\tau(x)}(x), \quad x', x \in \mathcal{P}. \tag{4.2}$$

Here the *first return function* $\tau(x)$ is the time of flight to the next section for a trajectory starting at x. The choice of a section hypersurface $\mathcal{P}$ is altogether arbitrary. The hypersurface can be specified implicitly through a function $U(x)$ that is zero whenever a point x is on the Poincaré section.
The gradient of $U(x)$ evaluated at $x \in \mathcal{P}$ serves a two-fold function. First, the flow should pierce the hypersurface $\mathcal{P}$, rather than being tangent to it. This leads to the *transversality condition*

$$(f \cdot U) \neq 0, \quad U_j = \frac{d}{dx_j}U(x), \quad x \in \mathcal{P}. \tag{4.3}$$

Second,the gradient U_j defines the orientation of the hypersurface $\mathcal{P}$. The flow is orientated as well, and even the shortest periodic orbit ($F^t(x) = F^{t+T_p}(x)$, for a given minimum period T_p) can pierce $\mathcal{P}$ twice, transversing in either direction. Hence a more natural definition of Poincaré return map is obtained by supplementing (4.2) with the condition

$$x_{n+1} = P(x_n), \quad U(x_{n+1}) = U(x_n) = 0, \quad n \in \mathbb{Z}^+ \\ f(x_n) \cdot U(x_n) > 0. \tag{4.4}$$

In this way the continuous time t flow $F^t(x)$ is reduced to a discrete time n sequence x_n of successive *oriented* trajectory traversal of $\mathcal{P}$.
In general, there are no explicit form for the Poincaré map. Anyway, it can be found numerically in different ways.

4.3 Local stability

4.3.1 Flow transport of neighborhood

We want to investigate the deformation of a neighborhood of a starting point $x_0 = x(0)$. This is best understood by considering a trajectory originating near x_0 with an initial infinitesimal displacement $\delta x(0)$, and letting the flow transport the displacement $\delta x(t)$ along the trajectory $x(x_0, t) = F^t(x_0)$. The system of linear *equations of variations* for the displacement of the infinitesimally close neighbor $x + \delta x$, from the vector field associated to the flow $\dot{x}_i = f_i$, by Taylor expanding to linear order read

$$\dot{x}_i + \delta\dot{x}_i = f_i(x + \delta x) \approx f_i(x) + \sum_j \frac{\partial f_i}{\partial x_j}\delta x_j.$$

The infinitesimal displacement δx is thus transported along the trajectory $x(x_0, t)$, with time variation given by

$$\frac{d}{dt}\delta x_i(x_0, t) = \sum_j \left.\frac{\partial f_i(x)}{\partial x_j}\right|_{x=x(x_0,t)} \delta x_j(x_0, t). \tag{4.5}$$

As both the displacement and the trajectory always depend on the initial condition point x_0 and the time t, we shall often abbreviate the notation $x(x_0, t) \rightarrow x(t) \rightarrow x$, $\delta x_i(x_0, t) \rightarrow \delta x_i(t) \rightarrow \delta x$ in what follows. Taken together, the set of equation

$$\dot{x}_i = f_i(x), \quad \delta x_i = \sum_i A_{ij}(x)\delta x_j \tag{4.6}$$

governs the dynamics in the tangent bundle $(x, \delta x) \in \mathrm{T}\mathcal{M}$ obtained by adjoining the d-dimensional tangent space $\delta x \in \mathrm{T}_x\mathcal{M}$ to every point $x \in \mathcal{M}$. The matrix of variations

$$A_{ij}(x) = \frac{\partial f_i(x)}{\partial x_j} \tag{4.7}$$

describes the instantaneous rate of shearing of the infinitesimal neighborhood of $x(t)$ by the flow.

Taylor expanding a *finite time* flow to linear order,

$$F_i^t(x_0 + \delta x) = F_i^t(x_0) + \sum_j \frac{\partial F_i^t(x_0)}{\partial x_{0j}} \delta x_j + \dots \tag{4.8}$$

one finds that the linearized neighborhood is transported by the *Jacobian* (or *fundamental*) *matrix*

$$\delta x(t) = \mathbf{J}^t(x_0)\delta x(0), \quad \mathbf{J}_{ij}^t(x_0) = \left.\frac{\partial x_i(t)}{\partial x_j}\right|_{x=x_0}, \tag{4.9}$$

which describes the deformation of an infinitesimal neighborhood at finite time t in the co-moving frame of $x(t)$. Looking at eigenvectors and eigenvalues of this matrix, one finds: *unstable directions* (eigenvalue magnitude larger than 1), along which nearby trajectories separates exponentially; *stable directions* (eigenvalue magnitude smaller than 1), along which nearby trajectories approach each other exponentially; *marginal directions* (eigenvalue magnitude equal to 1), along which nearby trajectories change their distance with a rate less than exponential.
If x_q is an equilibrium for the flow, than $\mathbf{J}^t(x_q) = e^{\mathbf{A}t}$, where $\mathbf{A}$ is the linearization of the velocity field in x_q.
As the eigenvalues of $\mathbf{J}^t$ have invariant meaning only for periodic orbits, we postpone their interpretation to Section 4.4.

4.3.2 Stability of maps

The transformation of an infinitesimal neighborhood of a trajectory under the iteration of a map follows from Taylor expanding the iterated mapping at *discrete* time n to linear order, as in (4.8). The linearized neighborhood is transported by the Jacobian matrix evaluated at a discrete set of times $n = 1, 2, \dots,$

$$\mathbf{J}_{ij}^n(x_0) = \left.\frac{\partial f_i^n(x)}{\partial x_j}\right|_{x=x_0}. \tag{4.10}$$

Stability of Poincaré return maps

We now relate the linear stability of the Poincaré return map (4.2,4.4) to the stability of the continuous time flow in the full phase space.
The hypersurface $\mathcal{P}$ can be specified implicitly through a function $U(x)$ that is zero whenever a point x is on the Poincaré section. A nearby point $x + \delta x$ is on $\mathcal{P}$

if $U(x+\delta x)=0$, and the same is true for variation around the first return point $x'=F(x_0,\tau(x_0))$ so expanding $U(x')$ to linear order in δx leads to the condition

$$U_i(x)\frac{dx'_i}{dx_j}\bigg|_{\mathcal{P}}=0. \tag{4.11}$$

In what follows U_i is the gradient of U defined in (4.3), unprimed quantities refer to the initial state, and the primed quantities to the first return. Both the first return x' and the time of flight to the next section $\tau(x)$ depend on the starting point, so the Jacobian matrix

$$\hat{\mathbf{J}}(x)_{ij}=\frac{dx'_i}{dx_j}\bigg|_{\mathcal{P}} \tag{4.12}$$

with both initial and the final variation constrained to the Poincaré section hypersurface $\mathcal{P}$ is related to the continuous flow Jacobian matrix by

$$\frac{dx'_i}{dx_j}\bigg|_{\mathcal{P}}=\frac{\partial x'_i}{\partial x_j}+\frac{dx'_i}{dt}\frac{dt}{dx_j}=\mathbf{J}_{ij}+v'_i\frac{dt}{dx_j}.$$

The return time variation dt/dx can be eliminated by substituting this expression into the constraint (4.11), yielding the projection of the full-space Jacobian matrix to the Poincaré map Jacobian matrix:

$$\hat{\mathbf{J}}_{ij}=\left(\delta_{ik}-\frac{f'_iU'_k}{f'\cdot U'}\right)\mathbf{J}_{ik}. \tag{4.13}$$

Given that $f(x(t))=\mathbf{J}^tf(x_0)$, it's easy to verify that $f(x)$ is a zero eigenvector of $\hat{\mathbf{J}}$

$$\hat{\mathbf{J}}\cdot f=0 \tag{4.14}$$

so the Poincaré section eliminates variations parallel to f, and $\hat{\mathbf{J}}$ is a rank $(d-1)$ matrix

4.4 Cycle stability

For chaotic systems almost all trajectories are aperiodic - nevertheless, stationary and periodic orbits will turn to be the key to unraveling chaotic dynamics.
An obvious virtue of periodic orbits is that they are topological invariants: a fixed point remains a fixed for any choice of coordinates, and similarly a periodic orbit remains periodic in any representation of the dynamics. Any re-parametrization of a dynamical system that preserves its topology has to preserve topological relations between periodic orbits, such as their relative inter-windings and knots. So, the

mere existence of periodic orbits suffices to partially organize the spatial layout of a non-wondering set. No less important, as we shall now show, is the fact that cycle stability eigenvalues are *metric* invariants: they determine the relative size of neighborhood in a non-wandering set.

To prove this, we start by noting that due to the multiplicative structure of the Jacobian Matrix,

$$\mathbf{J}^{t+t'}(x) = \mathbf{J}^{t'}(x')\mathbf{J}^{t}(x), \quad \text{where } x' = F^t(x),$$

the stability of the rth repeat of a prime cycle of period T_p is

$$\mathbf{J}^{rT_p}(x) = \mathbf{J}^{T_p}(F^{(r-1)T_p}(x))\ldots\mathbf{J}^{T_p}(F^{T_p}(x))\mathbf{J}^{T_p}(x) = (\mathbf{J}_p(x))^r, \tag{4.15}$$

where $\mathbf{J}_p(x) = \mathbf{J}^{T_p}(x)$ is the stability matrix for a single traversal of the prime cycle p, $x \in p$ is any point on the cycle. Hence, it suffices to restrict our consideration to stability of prime cycles.

4.4.1 Stability eigenvalues

We sort the *stability eigenvalues* $\Lambda_{p,1}, \Lambda_{p,2}, \ldots, \Lambda_{p,d}$ of the $[d \times d]$ Jacobian matrix $\mathbf{J}_p$ evaluated on the p-cycle into sets e, m, c

$$\begin{array}{rrcl} \text{expanding:} & \{\Lambda_p\}_e & = & \{\Lambda_{p,j} : |\Lambda_{p,j} > 1\} \\ \text{marginal:} & \{\Lambda_p\}_m & = & \{\Lambda_{p,j} : |\Lambda_{p,j} = 1\} \\ \text{contracting:} & \{\Lambda_p\}_c & = & \{\Lambda_{p,j} : |\Lambda_{p,j} < 1\} \end{array} \tag{4.16}$$

and denote by Λ_p (no jth eigenvalue index) the product of expanding eigenvalues

$$\Lambda_p = \prod_e \Lambda_{p,e}. \tag{4.17}$$

As $\mathbf{J}_p$ is a real matrix, complex eigenvalues always come in complex conjugate pairs, so the product of expanding eigenvalues is always real.

From stability eigenvalues we can define *stability exponents*, the stretching/contracting rates per unit time

$$\lambda_{p,i} = \frac{1}{T_p}\ln|\Lambda_{p,i}|. \tag{4.18}$$

If all stability exponents of *all* periodic orbits of a flow are strictly bounded away from zero, $|\lambda_i| \geq \lambda_{\min} > 0$, the flow is said to be *hyperbolic*. Otherwise the set is said to be *non-hyperbolic*.

Cycle *eigenvalues* are intrinsic properties of a cycle. Consider the ith eigenvalue, eigenvector pair $(\Lambda_{p,i}, \mathbf{e}_i)$ computed from $\mathbf{J}_p$ evaluated at a cycle point,

$$\mathbf{J}_p(x)\mathbf{e}_i(x) = \Lambda_{p,i}\mathbf{e}_i(x), \quad x \in p. \tag{4.19}$$

Consider another point on the cycle at time t later, $x' = F^t(x)$ whose Jacobian matrix is $\mathbf{J}_p(x')$. As $\mathbf{J}^{T_p+t} = \mathbf{J}^{t+T_p}$, the Jacobian matrix at x' can be rewritten either as

$$\mathbf{J}^{T_p+t} = \mathbf{J}^{T_p}(x')\mathbf{J}^t(x) = \mathbf{J}_p(x')\mathbf{J}^t(x), \quad \text{or} \;\; \mathbf{J}^{t+T_p} = \mathbf{J}^t(x)\mathbf{J}_p(x).$$

Multiplying (4.19) by $\mathbf{J}^t(x)$, we find that the Jacobian matrix evaluated at x' has the same eigenvalue $\Lambda_{p,i}$ of the Jacobian matrix evaluated at x, but with eigenvector $\mathbf{J}^t(x)\mathbf{e}_i(x)$

$$\mathbf{J}_p(x')\mathbf{e}_i(x') = \Lambda_{p,i}\mathbf{e}_i(x'), \quad \mathbf{e}_i(x') = \mathbf{J}^t(x)\mathbf{e}_i(x). \tag{4.20}$$

4.4.2 Stability of Poincaré map cycles

When a continuous flow periodic orbit p pierces the Poincaré section $\mathcal{P}$ once, the section point is a fixed point with stability (4.13), but with all primes dropped, being $x = x'$

$$\hat{\mathbf{J}}_{ij} = \left(\delta_{ik} - \frac{f_i U_k}{f \cdot U}\right)\mathbf{J}_{ik} \tag{4.21}$$

We have already established in (4.14) that the velocity $f(x)$ is a zero-eigenvector of the Poincaré section Jacobian matrix. Consider next $(\Lambda_{p,\alpha}, \mathbf{e}_\alpha)$, the full phase space αth (eigenvalue,eigenvector) pair (4.19), evaluated at a cycle point on a Poincaré section

$$\mathbf{J}(x)\mathbf{e}_\alpha(x) = \Lambda_\alpha \mathbf{e}_\alpha(x), \quad x \in \mathcal{P}. \tag{4.22}$$

If we multiply (4.21) by $\mathbf{e}_\alpha$ and inserting (4.22), we find that the Poincaré section Jacobian matrix $\hat{\mathbf{J}}$ has the same eigenvalue as the full phase space Jacobian matrix,

$$\hat{\mathbf{J}}(x)\hat{\mathbf{e}}_\alpha(x) = \Lambda_\alpha \hat{\mathbf{e}}_\alpha(x), \quad x \in \mathcal{P} \tag{4.23}$$

where $\hat{\mathbf{e}}_\alpha$ is a projection of the full phase space eigenvector onto the Poincaré section:

$$(\hat{\mathbf{e}})_i = \left(\delta_{ik} - \frac{f_i U_k}{f \cdot U}\right)(\mathbf{e}_\alpha)_k. \tag{4.24}$$

Hence, $\hat{\mathbf{J}}_p$ evaluated on any Poincaré section point along the cycle p has the same set of stability eigenvalues $\{\Lambda_{p,1}, \ldots, \Lambda_{p,d}\}$ as the full phase space Jacobian matrix $\mathbf{J}_p$.

4.5 Dynamical averaging, evolution operators, traces and determinants, dynamical zeta function: why cycles are so important

In chaotic dynamics detailed prediction is impossible, as any finitely specified initial condition, no matter how precise, will fill out the entire accessible phase space. Hence for chaotic dynamics one cannot follow individual trajectory for a long time; what is attainable is a description of the geometry of the set of possible outcomes, and evaluation of long time averages. Example of such averages are transport coefficient for chaotic dynamical flows, such as escape rate, mean drift and diffusion rate; power spectra; and a host of mathematical constructs such as generalized dimensions, entropies and Lyapunov exponents. Here we outline how such averages are evaluated within the *evolution operator* framework. The key idea is to replace expectation values of observable by the expectation values of generating functionals. This associates an evolution operator with a given observable, and relate the expectation value of the observable to the leading eigenvalue of the evolution operator. It will be then shown how these eigenvalues are dual to the *spectrum of periodic orbit*, which can be interpreted as the *skeleton* of a chaotic dynamics. For practical purpose an approximated formulation for long time leads to the *dynamical zeta function*, which vanishes on the leading eigenvalue of the evolution operator.

4.5.1 Time and space averages and evolution operator

Let $a = a(x)$ be any observable. We define the *time average* of a, along a trajectory starting in x_0, as

$$\overline{a(x_0)} = \lim_{t\to\infty} \frac{1}{t} A^t(x_0), \tag{4.25}$$

where

$$A^t(x_0) = \int_0^t d\tau a(F^\tau(x_0)) \tag{4.26}$$

is the *integrated observable.*

We remind that, being the system ergodic and mixing, this time average is equal to the spatial mean with respect to the natural measure of the system, ρ_0, and hence to the expectation value we are seeking. In most practical cases, this equality does not help, as it requires either the knowledge of the natural measure, or very long observation time.

There is another, more clever way to avoid this problem. We define the *space* *average* of a, $\langle a\rangle(t)$ as

$$\begin{aligned}\langle a\rangle(t) &= \frac{1}{|\mathcal{M}|}\int_{\mathcal{M}} dxa(x(t)) \\ |\mathcal{M}| &= \int_{\mathcal{M}} dx.\end{aligned} \tag{4.27}$$

We would have used any initial smooth density, perhaps concentrated on some small neighborhood, as any density will naturally tend to ρ_0. The worst case is the one just consider, that is, we weight every point the same. We then define the *expectation value* of a, $\langle a\rangle$, as

$$\langle a\rangle = \lim_{t\to\infty}\int_{\mathcal{M}} dx\frac{1}{t}\int_0^t d\tau a(F^\tau(x)). \tag{4.28}$$

The expectation value is a space average of time averages, with the advantage of averaging over space which smears over the starting point which were problematic for the time average, such as periodic points. While easy to define, the expectation value $\langle a\rangle$ turns out not to be particularly tractable in practice. Here come a simple idea which is the basis of all that follows: we investigate the space averages of form

$$\left\langle e^{\beta\cdot A^t}\right\rangle = \frac{1}{|\mathcal{M}|}\int_{\mathcal{M}} dx\; e^{\beta\cdot A^t}. \tag{4.29}$$

In the present context β is an auxiliary variable of no particular physical significance, and we will limit to consider scalar values of it.
As we already said, being the system ergodic, we expect the time average (4.25) to tend to same value $\langle a\rangle$ for almost all trajectories, and the integrated observable (4.26 to tend to $t\langle a\rangle$. So, as $t\to\infty$ we would expect the space average (4.29) to grow exponentially with time

$$\left\langle e^{\beta\cdot A^t}\right\rangle \propto e^{ts(\beta)},$$

and its rate of grow to go to a limit

$$s(\beta) = \lim_{t\to\infty}\frac{1}{t}\ln\left\langle e^{\beta\cdot A^t}\right\rangle. \tag{4.30}$$

Now we understand one reason for why it is smarter to compute $\left\langle e^{\beta\cdot A^t}\right\rangle$ rather than $\langle a\rangle$: the expectation value of the observable (4.28) and the moment of the integrated observable (4.26) can be computed by evaluating the derivatives of $s(\beta)$

$$\begin{aligned}\left.\frac{\partial s}{\partial\beta}\right|_{\beta=0} &= \lim_{t\to\infty}\frac{1}{t}\langle A^t\rangle = \langle a\rangle, \\ \left.\frac{\partial^2 s}{\partial\beta^2}\right|_{\beta=0} &= \lim_{t\to\infty}\frac{1}{t}\langle (A^t - t\langle a\rangle)^2\rangle,\end{aligned} \tag{4.31}$$

and so forth.
We can make the flow dependence of (4.29) explicit rewriting this quantity as

$$\left\langle e^{\beta\cdot A^t}\right\rangle = \frac{1}{|\mathcal{M}|}\int_{\mathcal{M}} dx \int_{\mathcal{M}} dy\delta(y - F^t(x))e^{\beta\cdot A^t}. \qquad (4.32)$$

Here $\delta(y - F^t(x))$ is the Dirac delta function: for a deterministic flow an initial point x maps into a unique point y at time t. Formally all we have done above is to insert the identity

$$1 = \int_{\mathcal{M}} dy\delta(y - F^t(x)), \qquad (4.33)$$

to make explicit the fact that we are averaging only over the trajectories that remain in in $\mathcal{M}$ for all times. However, having made this substitution we have replaced the study of individual trajectories $F^t(x)$ by the study of the evolution of density of the *totality* of initial conditions. Instead of trying to extract a temporal average from an arbitrarily long trajectory which explores the phase space ergodically, we can now probe the entire phase space with short and controllable finite time pieces of trajectories originating from every point in $\mathcal{M}$.
We will refer to

$$\mathcal{L}^t(y, x) = \delta(x - F^t(x))e^{\beta\cdot A^t(x)}. \qquad (4.34)$$

as the *evolution operator*. As a matter of fact *infinitesimally* short time evolution can suffice to determine its spectrum and eigenvalues.
In term of the evolution operator, the expectation value (4.29) is given by

$$\left\langle e^{\beta\cdot A^t}\right\rangle = \langle \mathcal{L}^t \mathbf{1}\rangle,$$

where $\mathbf{1}(x) = 1$ for all x. If $\mathcal{L}^t$ were a matrix we would be computing a weighted sum of its eigenvalues which is dominated by the leading one as $t \to \infty$. By analogy, as the trace of $\mathcal{L}^t$ is also dominated by the leading eigenvalue as $t \to \infty$, we might just want to look at the trace

$$\left\langle e^{\beta\cdot A^t}\right\rangle = \mathrm{tr}\mathcal{L}^t = \int dx\delta(x - F^t(x))e^{\beta\cdot A^t(x)}. \qquad (4.35)$$

Apart from practical advantages, inserting the identity (4.33) in equation (4.29) allows the following interpretation in term of *periodic orbits* of expectation values. We recall the property of the Dirac delta function

$$\int dx\delta(y - F^t(x)) = \sum_{(x - F^t(x))=0} \frac{1}{|det\frac{\partial F^t(x)}{\partial x}|}. \qquad (4.36)$$

According to this formula, *the trace (4.35) picks up contribution whenever $x - F^t(x) = 0$, that is, whenever x belongs to a periodic orbit.*
The explicit form of this trace can be derived, both for discrete and continuous time systems. In the first case we also apply the *zeta*-transform, and in the second the *Laplace*-transform (which is nothing but the zeta-transform with $z = e^{-s}$). Doing this will lead to an explicit duality between the spectrum of eigenvalues of the evolution operator and the spectrum of periodic orbit.

4.5.2 Trace and determinants

We have shown how global average relate to eigenvalues of appropriate evolution operators. Traces of evolution operators can be evaluated as integrals over Dirac delta functions, and this way spectra of evolution operators become related to periodic orbits. Here we want to show explicitly the global↔local duality which says

> the spectrum of eigenvalues (global) is dual to the spectrum of periodic orbits (local)

For dynamics on the circle, this is called Fourier analysis; for dynamics on well-tiled manifolds, Selsberg traces and zetas; and for generic nonlinear dynamical systems the duality is embodied in the trace formulas that we will introduce.
First of all, we recall the hyperbolicity assumption, that is we assume that $\mathbf{J}_p$ has no marginal eigenvalue. It's easy to understand why we have to do this in the case of discrete time maps (the continuous time is just more coumbersome, but conceptually identical).
The contribution to trace of an isolated prime cycle p of period n_p for a map f can be evaluated by restricting the integration to an infinitesimal open neighborhood $\mathcal{M}_p$ around the cycle,

$$\mathrm{tr}_p \mathcal{L}^{n_p} = \int_{\mathcal{M}_p} dx \delta(x - f^{n_p}(x)) = \frac{n_p}{|\det(\mathbf{1} - \mathbf{J}_p)|} = n_p \prod_{i=1}^{d} \frac{1}{|1 - \Lambda_{p,i}|}, \qquad (4.37)$$

which can be carried out only if the cycle Jacobian has no eigenvalues of unit magnitude.
In this case we can also factorize the trace (4.37) into a product over the contracting and expanding eigenvalues

$$|\det(\mathbf{1} - \mathbf{J}_p)|^{-1} = \frac{1}{|\Lambda_p|} \prod_e \frac{1}{1 - 1/\Lambda_{p,e}} \prod_c \frac{1}{1 - \Lambda_{p,c}}, \qquad (4.38)$$

where $\Lambda = \prod_e \Lambda_{p,e}$.

Trace formula for maps

With the above assumption the trace formula for map can be written as

$$\mathrm{tr}\mathcal{L}^n = \sum_p n_p \sum_{r=1}^{\infty} \frac{e^{r\beta\cdot A_p}}{|\det(\mathbf{1}-\mathbf{J}_p^r)|}\delta_{n,n_p r}, \tag{4.39}$$

with the Kronecker delta $\delta_{n,n_p r}$ projecting out the period contribution of total period n. This contribution is awkward, and will be more awkward in the continuous time case, where it yields a series of Dirac delta functions. Such sums are familiar from the density-of-states sums of statistical mechanics, where they dealt with in the same way as we shall do here: we smooth this distribution by taking a Laplace transform, or in the discrete time case, a zeta transform.
We *define* the trace formula for maps to be the zeta transform of (4.39), which, for discrete time mappings, is simply the generating function for the trace sums

$$\sum_{n=1}^{\infty} z^n \mathrm{tr}\mathcal{L}^n = \mathrm{tr}\frac{z\mathcal{L}}{1-z\mathcal{L}} = \sum_p n_p \sum_{r=1}^{\infty} \frac{z^{n_p r} e^{r\beta\cdot A_p}}{|\det(\mathbf{1}-\mathbf{J}_p^r)|}. \tag{4.40}$$

Expressing this formula in terms of the sum of the eigenvalues of $\mathcal{L}$, we obtain the *trace formula for maps*:

$$\sum_{\alpha=0}^{\infty} \frac{ze^{s_\alpha}}{1-ze^{s_\alpha}} = \sum_p n_p \sum_{r=1}^{\infty} \frac{z^{n_p r} e^{r\beta\cdot A_p}}{|\det(\mathbf{1}-\mathbf{J}_p^r)|}, \tag{4.41}$$

which expresses the duality between the spectrum of the evolution operator and the spectrum of periodic orbit.

Trace formula for flows

It can be derived that in continuous time case the trace (4.35) can be written as

$$\mathrm{tr}\mathcal{L}^t = \sum_p T_p \sum_{r=1}^{\infty} \frac{e^{r\beta\cdot A_p}}{|\det(\mathbf{1}-\mathbf{J}_p^r)|}\delta(t-rT_p). \tag{4.42}$$

A trace formula follows by taking a Laplace transform. This is a delicate step, since the transfer operator becomes the identity in the $t \to 0^+$ limit. In order to make sense of the trace we regularize the Laplace transform by a lower cut-off ϵ smaller than the period of any periodic orbit, and write

$$\begin{aligned}\int_\epsilon^\infty dt\; e^{-st}\mathrm{tr}\mathcal{L}^t &= \mathrm{tr}\frac{e^{-(s-\mathcal{A})\epsilon}}{s-\mathcal{A}} = \sum_{\alpha=0}^{\infty}\frac{e^{-(s-s_\alpha)\epsilon}}{(s-s_\alpha}) = \\ &= \sum_p T_p \sum_{r=1}^{\infty} \frac{e^{r(\beta\cdot A_p - sT_p)}}{|\det(\mathbf{1}-\mathbf{J}_p^r)|},\end{aligned} \tag{4.43}$$

where $\mathcal{A}$ is the generator of the semigroup of dynamical evolution

$$\mathcal{L}^t = e^{\mathcal{A}t}. \tag{4.44}$$

The *classical trace formula for flows* is the $\epsilon \to 0$ limit of the above expression:

$$\sum_{\alpha=0}^{\infty} \frac{1}{s - s_\alpha} = \sum_p T_p \sum_{r=1}^{\infty} \frac{e^{r(\beta\cdot A_p - sT_p)}}{|\det(\mathbf{1} - \mathbf{J}_p^r)|}. \tag{4.45}$$

This formula is still another example of the duality between the (local) cycles and (global) eigenvalues.

Asymptotic trace formula

The Laplace transform (4.41) and (4.45) are designed to capture the time→ ∞ asymptotic behavior of the trace sums. By the hyperbolicity assumption, for large $t = rT_p$, the cycle weight approaches

$$|\det(\mathbf{1} - \mathbf{J}_p^r)| \to |\Lambda_p|^r \tag{4.46}$$

where $\Lambda_p = \prod_e \Lambda_{p,e}$, the product of the expanding eigenvalue of $\mathbf{J}_p$. Substituting in (4.45, we get the *asymptotic trace formula*

$$\Gamma(s) = \sum_p \frac{n_p t_p}{1 - t_p}, \quad t_p = e^{-sT_p} e^{\beta\cdot A_p}/|A_p|. \tag{4.47}$$

To recover the maps case, $e^{-s} \to z$, $T_p \to n_p$.

From traces to determinant

The problem in dealing with traces (4.41) and (4.45) is that we have to determine their poles in order to get eigenvalues.
There is a simple way to get rid of this problem by considering the matrix identity

$$\ln\det(1 - M) = \mathrm{tr}\ln(1 - M) = -\sum_{n=1}^{\infty} \frac{1}{n}\mathrm{tr}M^n, \tag{4.48}$$

which leads to the relation between *spectral determinant* and trace for an evolution operator

$$\det(1 - z\mathcal{L}) = \exp\left(\sum_p \sum_{r=1}^{\infty} \frac{1}{r} \frac{z^{n_p r} er\beta \cdot A_p}{|\det(\mathbf{1} - \mathbf{J}_p^r)|}\right) \tag{4.49}$$

for the maps case, and

$$\det(s-\mathcal{A}) = \exp\left(\sum_p \sum_{r=1}^{\infty} \frac{1}{r} \frac{e^{r(\beta\cdot A_p - sT_p)}}{|\det(\mathbf{1}-\mathbf{J}_p^r)|}\right) \tag{4.50}$$

for flows. Now we can recover trace formula (4.41) and (4.45) from, respectively,

$$\mathrm{tr}\frac{z\mathcal{L}}{1-z\mathcal{L}} = -z\frac{d}{dz}\ln\det(1-z\mathcal{L}) \tag{4.51}$$

and

$$\mathrm{tr}\frac{1}{s-\mathcal{A}} = \frac{d}{ds}\ln\det(s-\mathcal{A}). \tag{4.52}$$

This way we avoid the problem of determine poles by finding zeros. Note that, with the substitution $z = e^{-s}$, spectral determinant has the same form both for maps and for flows.

4.5.3 Dynamical zeta function

If instead of the exact traces, we consider the large time behavior of the Jacobian's eigenvalues product (4.46) and the relative asymptotic trace formula (4.47), the spectral determinant (4.50) is replaced by the *dynamical zeta function*

$$1/\zeta = \exp\left(-\sum_p \sum_{r=1}^{\infty} \frac{1}{r} t_p^r\right), \tag{4.53}$$

which, resumming the logarithm by $\sum_r t_p^r/r = -\ln(1-t_p)$, can also be written as

$$1/\zeta = \prod_p (1-t_p), \tag{4.54}$$

with t_p defined by equation (4.47).
Also in the case of asymptotic behavior, there is a trace-determinant relation given by

$$\Gamma(s) = \frac{d}{ds}\ln\zeta^{-1} = \sum_p \frac{T_p t_p}{1-t_p}. \tag{4.55}$$

Hence, the dynamical zeta function is important because

$$1/\zeta(s) = 0$$

vanishes at $s = s_0$ the leading eigenvalue of $\mathcal{L}^t$.

4.6 Cycles are dense!

Our goal was to show that unstable periodic orbits form the skeleton of any chaotic dynamics. We show this by considering the dynamical averaging of some observable, concluding that it picks up contributions only from periodic orbits. However the average over all periodic orbits can accomplish the job only if the periodic orbits fully explore the asymptotically accessible phase space. Why should the unstable periodic points end up being dense?

The cycle are intuitively expected to be dense because on a connected chaotic set a typical trajectory is expected to behave ergodically, and pass infinitely many times arbitrarily close to any point on the set, including the initial point of the trajectory itself. The argument is more or less the following. Take a partition of the phase space $\mathcal{M}$ in arbitrarily small regions, and consider particles that start out in region $\mathcal{M}_i$, and return to it in time T after a peregrination in phase space. In particular, a particle might return a little to the left of its original position, while a close neighbor might return a little to the right of its original position. By assumption, the flow is continuous, so generically one expects to be able to gently move the initial point, that is, one expect a periodic point of period T in cell i. As we diminish the size of region $\mathcal{M}_i$, aiming to find a trajectory that returns to $\mathcal{M}_i$ becomes increasingly difficult. Therefore, we are guaranteed that unstable (because of the expansiveness of the flow) orbits of larger and larger period are densely interspread in the asymptotic non-wandering set.

Chapter 5

Chaos control

As we explained in the last chapter, chaotic systems are characterized by three main features:

- sensitivity to initial conditions
- there is an infinite number of unstable periodic orbits embedded in the underlying chaotic set. In other words, the skeleton of a chaotic attractor is a collection of an infinite number of periodic orbits, each one being unstable
- the dynamics in the chaotic attractor is ergodic, which implies that during its temporal evolution the system ergodically visits small neighborhoods of every point in each one of the unstable periodic orbits embedded within the chaotic attractor

A relevant consequence of these properties is that a chaotic dynamics can be seen as shadowing some periodic behavior at a given time, and erratically jumping from one periodic orbit to another. The idea of controlling chaos is then when a trajectory approaches ergodically a desired periodic orbit embedded in the attractor, one applies small perturbations to stabilize such an orbit. If one switches on the stabilizing perturbations, the trajectory moves to the neighborhood of the desired periodic orbit that can now be stabilized. This fact has suggested the idea that the critical sensitivity of a chaotic system to changes (perturbations) in its initial conditions may be, in fact, very desirable in practical experimental situations.

> The important point here is that, because of chaos, one is able to produce an infinite number of desired dynamical behaviors (either periodic and not periodic) using the same chaotic system, with the help of only tiny perturbations chosen properly. We stress that this is not the case for non chaotic dynamics, wherein the perturbations to be made for producing a desired behavior must, in general, be of the same order of magnitude as the unperturbed evolution of the dynamical variables.

There are many practical reasons to control chaos. Formally, the main reason to stabilize an unstable orbit, is that we can bring some observable to have a desired expectation value.
In fact, consider a chaotic flow and one of its unstable periodic orbit p of period T_p. If we manage to stabilize it, it will be the only surviving cycle and the determinant (4.50) will become

$$\det(s - \mathcal{A}) = \exp\left(\sum_{r=1}^{\infty} \frac{1}{r} \frac{e^{r(\beta A_p - sT_p)}}{1 - \Lambda'^r}\right), \tag{5.1}$$

where Λ' is the product of the eigenvalues of the new Jacobian's matrix evaluated on cycle p.
Another very fascinating point, is that there is a simple connection between chaos and communication theory. Chaotic systems can be viewed as information sources that naturally produce digital communication signals. The formal connection between chaotic dynamics and information theory began with the introduction of the concept of measure-theoretic entropy in ergodic theory. Chaotic systems are, indeed, characterized by having positive entropy and thus they are information sources. By assigning a discrete alphabet to the system state space using the formalism of symbolic dynamics, the chaotic system becomes a symbol source, and because it is a continuous-time waveform source, it is also a digital signal source. This fact can be used to synthesize ”meaning sources” through chaos control, such as in Associative Memories, obtained through chaotic neural networks (CNN) ([79],[83]). It is then possible to modulate each single chaotic orbit, as we would have done with standard sinusoidal wave, to carry different information on each of them ([82]). Moreover chaos synchronization through small perturbations allows for secure and robust communication ([75]).

5.1 Formulation of the problem and main result

There are several practical techniques fro obtaining stabilization of unstable periodic orbits (UPOs) (see [73] for an overview of them). In any case, for the moment we want to deal only with the existence problem of a state feedback law which possesses some particular stability properties. Given these properties we will be able to explain in term of local input-to-state stability, the robustness of chaos control methods against parameter uncertainties and measurement/control disturbances.
Consider the following dynamical system described by ordinary differential equations

$$\dot{x} = f(x) \tag{5.2}$$

and the controlled system

$$\dot{x} = f(x) + u(x) \tag{5.3}$$

where $f(x)$ is a continuously differentiable map $\Re^n$ to $\Re^n$, and $u(x)$ is a continuous feedback controller. Denote by $\phi(x_0, t)$ the orbit of (5.2) with initial condition x_0, i.e. $\phi(x_0, 0) = x_0$, and by $\phi_u(x_0, t)$ the orbit of the controlled system (5.3) with initial condition x_0.
Since we are interested in controlling chaotic systems, it is natural to make the following assumptions, due to ergodicity of chaotic systems on their attractors.

***Assumption* 8** *There exists a compact invariant set* $\mathbf{A}$ *for (5.2) on which the dynamic of (5.2) is chaotic.*

With this assumption the following property holds.

Property 1 *There exist non-wandering points in (5.2).*

Here a non-wandering point is defined as follows.

Definition 9 *A point* $x \in \Re^n$ *is said to be a non-wandering point of (5.2), if it satisfies the condition that for each neighborhood* U *of* x*, there exists a* $T > 0$*, such that*

$$\phi(U, T) \cap U \neq \varnothing$$

where $\varnothing$ *denotes the empty set.*

The problem concerned here is: whether *it possible to construct a small feedback control law for (5.3) to create a new asymptotically stable periodic orbit (fixed point)?*
Note that here, unlike the treatment of this problem in the literature ([73]), we do not demand that the new asymptotically stable orbit is an unstable periodic orbit of the uncontrolled system (5.2). Futhermore this latter case is included as a special case.
In the case of Property 1, it is not difficult to prove the existence of small perturbation (feedback control) to guarantee the creation of a new *asymptotically stable* periodic trajectory by virtue of closing lemma theory (see [74]).

Theorem 4 (X. Yang, [74]) *Suppose that* $x_p \in M \subset \Re^n$ *is a non-wandering point of (5.2), where* M *is an invariant set of (5.2). For each* $\epsilon > 0$*, there exists a small feedback control* $u_p(x(t))$ *satisfying*

$$\|u_p(x(t))\| < \epsilon,$$

which ensures that the orbit $\phi_{u_p}(x_p, t)$ *of the controlled system (5.3) is an asymptotically stable periodic orbit.*

5.2 Analysis of stability properties

Here we want to use characteristics of chaotic dynamics to derive stability properties of chaos control.
The general idea is that the system being chaotic, any typical trajectory is expected to behave ergodically, and pass infinitely many times near any points on the set. This peculiarity allows, under certain assumptions, to relate local stability properties to global stability properties. Once the system is on the attractor, this result is immediately proved. Problems can derive from two factors:

- the attractor is *strange*
- the rate of convergence of the points in the basin of attraction is too slow

The first problem requires careful attention, as it deals with the intrinsic geometrical properties of the attractor, as only strange or *hyperbolic* attractors can have fractional dimensionality (they are *fractals*, see [71]). By dimensionality $\mathcal{D}$ we mean the ratio between its Hausdorff dimension, D_H (see [81] for an introduction to this concept) and the geometrical dimension of the system n, which we want to be 1

$$\mathcal{D}(\mathbf{A}) = \frac{D_H(\mathbf{A})}{n}.$$

In the case of it being lower, the chaotic attractor has zero Lebesgue measure, that is, intuitively, it is a small set surrounded by holes, and its basin of attraction can have fractal boundaries too (see the literature at this page for details on this argument see [86]) such that infinitesimal displacements in initial conditions will lead to completely different dynamical behaviors.
Hence we have to exclude this case.

***Assumption* 9** **A** *is a chaotic attractor for the system (5.2), and it is not an hyperbolic attractor. Moreover initial conditions belong to its basin of attraction and fall on it in finite time.*

Note on hyperbolic attractor

A hyperbolic attractor is a hyperbolic set which is also an attractor. A hyperbolic set for a dynamical system is a set entirely made of unstable hyperbolic trajectory (i.e the Jacobian at every point has non-unitary eigenvalues, except for the direction parallel to the flow), but with a particularity: *each trajectory in the set has the same number with the same eigenvalues of unstable and stable manifolds.*. This characteristic leads to very strange behavior of trajectories belonging to an hyperbolic attractor which makes hyperbolic attractors potential set with fractional dimensionality. In fact, they belong to the class of *strange* attractors,

and only strange attractors can have fractional dimension ([71]). The idea is that trajectories keep on leaving and entering unstable and stable manifolds each time with same characteristics, which leads to fractal-like characteristics of the resulting support.
What may seem strange, then, is that we made the *hyperbolicity assumption* in order to define the trace of an evolution operator, and now we want to avoid such assumption in order to achieve $\mathcal{D}(\mathbf{A}) = 1$.
The fact is that, hyperbolicity assumption over single trajectories does not lead to the global hyperbolicity of the attractor, as trajectory hyperbolicity can be non-uniform. And it is exactly this non-uniformity (in particular in the number of stable and unstable directions) which leads to $\mathcal{D}(\mathbf{A}) = 1$ for the resulting attractor. The idea is that also in this case trajectories keep on leaving and entering unstable and stable manifolds, but each time with different characteristics and this leads to fill the entire geometrical space ergodically.

With the last assumption we are in a good situation, as our initial conditions belong to an invariant compact set with full dimensionality, and which is explored ergodically by almost every trajectory.
This means that, given a point $x_0 \in \mathbf{A}$ and a neighborhood $U \ni x_0$,

$$\lim_{T\to\infty} \phi(U, -T) \supseteq \mathbf{A}$$

Reversing this relation, this also means that

$$\forall\delta > 0, \forall x \in \mathbf{A}, \exists T < \infty, \text{such that } \|\phi(x,T) - x_0\| < \delta. \tag{5.4}$$

Comparing the definitions of global asymptotic stability and asymptotic stability, we conclude that, given Assumption 9,

> *asymptotic stability is equivalent to global asymptotic stability over the attractor* $\mathbf{A}$

We are now ready to state the main result of this section.

Theorem 5 *Let u_p be a state feedback as in Theorem 4, and p the relative asymptotically stabilized periodic orbit. Then, the system (5.3) is local input-to-state stable with respect to p.*

Proof *For Theorem 4 the system is asymptotically stable with respect to p. As we noted this is equivalent to saying that system (5.3) is global asymptotically stable with respect to p over* $\mathbf{A}$. *For equation(3.10) this implies that system (5.3) is local input-to-state stable with respect to p.* □

Now we recall the definition of local input-to-state stability (3.5). Locality just implies that, as long as initial conditions and perturbation are bounded under a certain norm, ρ, all the approximations that are valid for ISS systems are valid for LISS systems. In particular all the results given in Sections 3.2.1 to 3.2.4 are still valid. As we already pointed out, initial conditions can be taken arbitrarily small with respect to the fixed point, due to system ergodicity. The fact that there is an upper bound to the allowed perturbation can be understood by considering that the system has an infinity of unstable periodic orbits, thus, a too strong perturbation will bring the trajectory either into an other periodic orbit, or, in the worst case, into a chaotic regime again.
Thus, we can say that, as long as

$$\|\sigma\| < \rho,$$

where σ is a perturbation acting on the system (due to parameter uncertainties, measurement errors, input disturbances, ...), *the system (5.3) is input-to-state stable (ISS) with respect to p.*

5.2.1 Controlled Poincaré return map and its stability properties

As we explained in Chapter 4 a continuous time dynamic, as the system (5.2), induces discrete time map on the Poincaré section, the Poincaré return map. The same is true for the controlled system (5.3). In fact, given a Poincaré section for the uncontrolled system, we define the *Controlled Poincaré return map* $P(x, \tilde{u}(x))$ as

$$x' = P(x_0, \tilde{u}(x_0)) = x_0 + \int_0^{\tau(x_0)} [f(x(t)) + u(x(t))]\, dt, \quad x(0) = x_0, \tag{5.5}$$

where we also define the function $\tilde{u}(x_0)$ as

$$\tilde{u}(x_0) = \{u(x(t))\}_{t\in[0,\tau(x))}\,, \tag{5.6}$$

where $x(0) = x_0$ and $x(t)$ in a solution of the system (5.3).For the discrete time continuous map, the input variable belongs to the space of Lipschitz functions over the time interval $[0, \tau(x))$. We thus define its norm in the following way

$$\|\tilde{u}(x_0)\|_L = \sup_{t\in[0,\tau(x))} \|u(x(t))\|_\infty. \tag{5.7}$$

When continuous time control is of the type described in Theorem 4, that is $u = u_p$, the stabilization of an unstable periodic orbit p is equivalent to the stabilization of a fixed point x_p for the Poincaré map. Being the system (5.3) local input-to-state stable with respect to p,

> the controlled Poincaré map $P(x, \tilde{u}_p(x))$ will be local input-to-state stable with respect to x_p.

Now we recall the definition of local input-to-state stability (3.5). Locality just implies that, as long as initial conditions and perturbation are bounded under a certain norm ρ, all the approximations valid for ISS systems are valid for LISS systems. In particular all the results given in Sections 3.2.1 to 3.2.4 are still valid. Again locality with respect to initial conditions is not a problem, while locality with respect to perturbation intensity reflects the fact that the Poincaré return map has an infinity of fixed points, thus a too strong perturbation will bring the system near another fixed point, or, in the worst case into a chaotic regime again.
How does an input perturbation σ in continuous time reflect on the Poincaré map? By definition we have $(x(0) = x_0)$

$$P_\sigma(x_0, \tilde{u}(x_0)) = x_0 + \int_0^\tau (x) \left[f(x(t)) + u(x(t)) + \sigma(x(t)) \right] dt = P(x_0, \tilde{u}(x_0) + \tilde{\sigma}(x_0)) \tag{5.8}$$

where $\tilde{u}(x_0) + \tilde{\sigma}(x_0) = \{u(x(t)) + \sigma(x(t))\}_{t \in [0, \tau(x))}, \quad x(0) = x_0$. Hence, by subtracting $\tilde{u}(x_0)$ from both sides, it results

$$\tilde{\sigma}(x_0) = \{\sigma(x(t))\}_{t \in [0, \tau(x))}, \quad x(0) = x_0$$

Now we can also relate the norm of this perturbation to the norm of the perturbation in continuous time, by

$$\|\tilde{\sigma}(x_0)\|_L = \sup_{t \in [0, \tau(x))} \|\sigma(x(t))\|_\infty, \quad x(0) = x_0. \tag{5.9}$$

By taking the max over all possible initial point x_0 of both sides, we can say that the discrete time perturbation has the same norm as the continuous time perturbation.
Thus, we can say that as long as

$$\|\sigma\| < \rho,$$

where σ is some perturbation acting on the system (due to parameter uncertainties, measurement errors, input perturbations, ...), the *Controlled Poincaré return map is input-to-state stable (ISS) with respect to x_p*, where x_p is the nonwandering point of Theorem 4.

5.3 Practical control algorithm: the Delayed-Feedback-Control robustness to parameter uncertainties

Results from the last section and results of section 3.2 let us hope that chaos control will be robust against parameters uncertainties, measurement errors and input disturbances. We want to test robustness to parameter uncertainties on a practical algorithm for chaos control: the Pyragas method or Delayed-Feedback-Control (DFC).

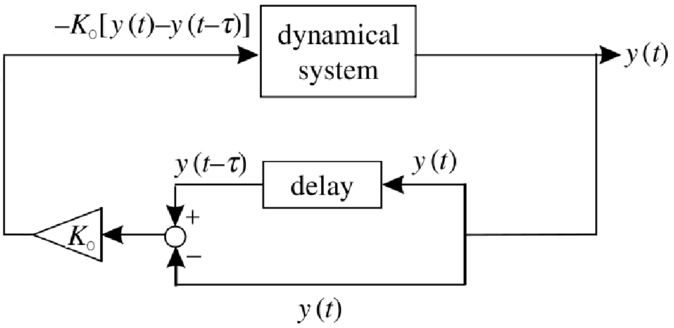


Figure 5.1: Block scheme of the Delay-Feedback-Control

Contrary to other chaos control methods, which require at least the partial knowledge of the uncontrolled dynamic and real-time computer processing to be achieved, the DFC method is reference-free and makes use of a control signal obtained from the difference between the current state of the system and the state of the system delayed by one period of the UPO which has to be stabilized. The block diagram of the method is presented in figure 5.1. Alternatively, the DFC method is referred to as a method of time-delay auto-synchronization, since the stabilization of the target orbit manifests itself as a synchronization of the current state of the system with its delayed state. The method allows us to treat the controlled system as a black box; no exact knowledge of either the form of the periodic orbit or the system of equations is needed. Taking into account only the period of the unstable orbit, the system under control automatically settles on the target periodic motion, and the stability of this motion is maintained with only small perturbations. The DFC algorithm is especially superior for fast dynamical systems, since it does not require any real-time computer processing.
This control is already known to be robust to the gain parameter K_0 (control-

lability range, see [75]). As we will see from one practical example, the exact knowledge of the period is not necessary as well, leading to an almost-blind chaos control method.

Moreover, the redundancy nature of the neural system makes the DFC a natural candidate to model global synchronous oscillations of groups of neurons as a period state of a controlled chaotic dynamics. In fact, in ([85]) it is shown how the emerging synchronization regimes, of a closed chain of uni-directionally coupled chaotic oscillators, show analogies with the experimental behavior of a single chaotic laser subjected to a delayed feedback.

5.3.1 Coupled FitzHugh-Nagumo equations chaos control

The uncontrolled system analysis is taken from [68].

In the following, we study a pair of excitable neurons, which are modeled by the FitzHugh-Nagumo FHN excitable elements ([63, 64, 68]), given by the following equations:

$$\begin{aligned}
\frac{du_1}{dt} &= u_1(u_1-\alpha)(1-u_1) - v_1 + \frac{K}{2}(u_2-u_1),\\
\frac{dv_1}{dt} &= (\tau(u_1-\gamma v_1)),\\
\frac{du_1}{dt} &= u_2(u_2-\alpha)(1-u_2) - v_2 + \frac{K}{2}(u_1-u_2),\\
\frac{dv_1}{dt} &= \tau(u_2-\gamma v_2),
\end{aligned} \tag{5.10}$$

where α, γ, τ, and K are parameters, $u_{1,2}(t)$ is the activator, and $v_{1,2}$ is the inhibitor. The reason that we use FHN neuron is that the equations are standard and minimal models for excitable systems in the following sense. The model contains no more than two variables, and does not describe a specific biochemical reaction: it can be used to describe both neural and cardiac dynamics.

Employing the following parameters: $\tau = 0.001$, $\gamma = 0$, each element is excitable for $\alpha > 0$, that is, a small but finite perturbation to the rest state $(u_1, v_1, u_2, v_2) = (0, 0, 0, 0)$ leads to a large excursion (an excitation). Because of the excitable nature of the units, the rest state is a globally stable solution when the coupling is excitable, i.e., $K > 0$. Indeed, starting from the rest state with a finite perturbation, these two elements excite and immediately synchronize. After synchronization, there are no input signals from the other element owing to diffusive coupling, and then, both elements immediately return to the rest states.

For a phase-repulsive coupling $K < 0$, however, periodic excitations are observed in some parameter regions. A rich variety of cyclic firing patterns is found numerically, as depicted in Fig. 5.2B. To characterize these periodic solutions, we introduce the following symbolic notation. In Fig. 5.2B-a , after one element excites, the other element excites soon. After this successive excitations, both elements stay at the quiescence state for a while. These excitations and quiescence are repeated, and we symbolize this firing pattern as AB-. Note that the code AB- and BA- correspond to the same periodic solution due to the invariance under the exchange of each element. In the same way, we characterize periodic solutions in Fig. 5.2B as AB-AB-BA-BA-, AB-BA-, and ABA-BAB-.

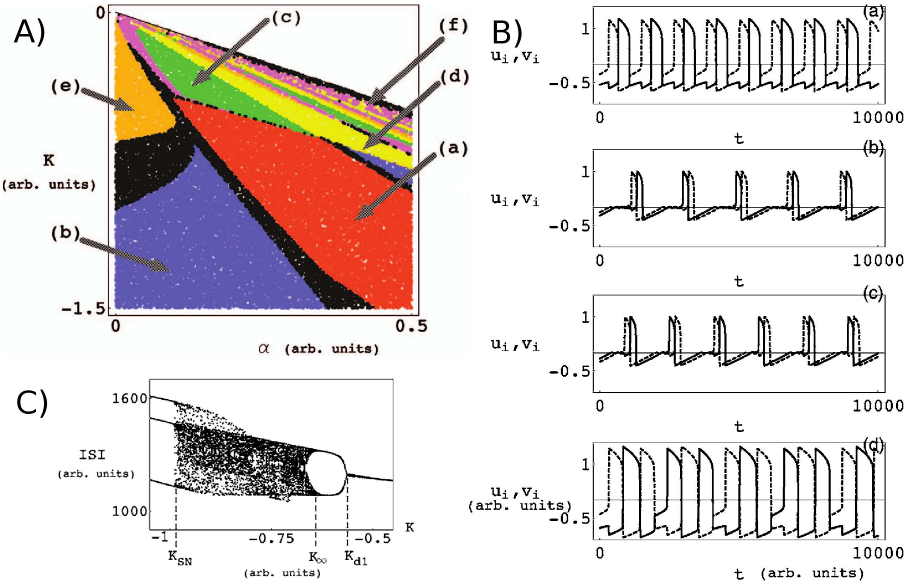


Figure 5.2: A) Firing pattern diagram in (K, α)-parameter space. The line represents the Hopf-bifurcation line: $K_{HP} = -\alpha - \gamma\tau$. Each colored dot corresponds to different firing pattern, (a) red: ABA-; (b) blue: ABA-BAB-; (c) green: AB-BA-; (d) orange: AB-; (e) yellow: AB-; (f) magenta: AB-BA-; and black: longer periodic or chaotic firing patterns. B) Time evolution of a pair of FHN elements with phase-repulsive coupling. Periodic oscillations appear below a critical coupling strength $K < -0.01$. Typical periodic solutions are shown: (a) $K = -0.5$: AB-firing pattern, (b) $K = -0.093 : AB - AB - BA - BA-$, (c) $K = -0.012 : AB - BA-$, (d) $K = -1.0 : ABA - BAB-$. The solid and broken lines represent $u_1(t)$ and $u_2(t)$, respectively. C) Interspike interval as a function of the coupling strength K (from [68])

To see dynamical behaviors, we calculate the interspike interval ISI , which is defined as follows. Integrating the Eq. (5.10) numerically we have used adaptive time step algorithm so called the fifth-order Runge-Kutta-Fehlberg formula because there exist slow and fast variables in the system , we have a sequence of time t_i at which excitation of element 1 occurs. The time sequence of the excitations can be obtained by $T = \{t_i | (u_1(t_i) = 0)\ ,\ (du_1(t_i)/dt > 0)\}$. The ISI defined as the sequence of $ISI_i = t_{i+1} - t_i$ is often used as a characterization of neural activ-

ities ([76]) . After an initial transient disappears, i.e., for $i \gg 1$, the superimposed ISI as a function of the parameter K is plotted in Fig. 5.2C. The ISI shows that the cyclic firing bifurcates to irregular one at $K = K_\infty$ by decreasing the coupling strength K. Further decreasing K, periodical firing appears again. The chaotic firing regions in (K, α)-parameter space are shown in Fig. 5.2A.
From stability analysis, it results that the rest state becomes unstable at critical values $K_{HP} = -\alpha - \gamma\tau$ via the Hopf bifurcation, i.e the fixed point loses its stability as a pair of complex conjugate eigenvalues of the linearization around the fixed point cross the imaginary axis of the complex plane. Under reasonably generic assumptions about the dynamical system, we can expect to see a small amplitude limit cycle branching from the fixed point. The limit cycle is orbitally stable if a certain quantity called the first Lyapunov coefficient is negative, and the bifurcation is supercritical. Otherwise it is unstable and the bifurcation is subcritical.

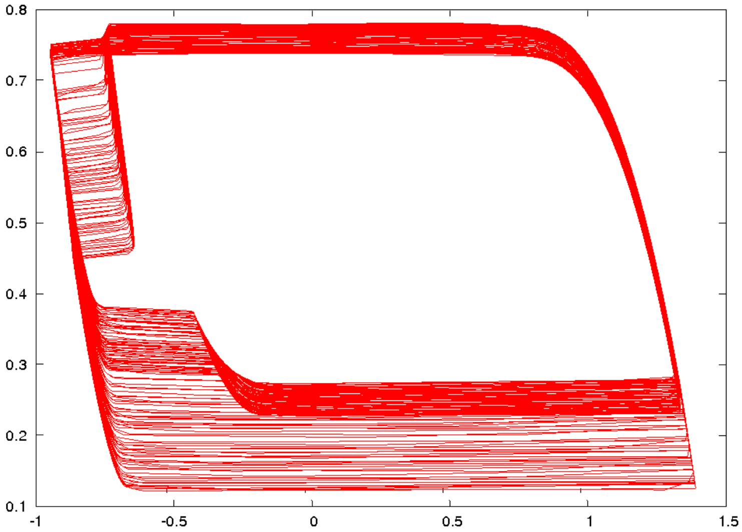


Figure 5.3: Projection of the chaotic attractor of the system (5.10)

In this case the bifurcation is subcritical, and the solution branch that stems from it is unstable. This means that for $K < K_{HP}$ the periodic pattern described in figure 5.2B) are not directly connected to the HP branch of unstable period solution. On the contrary they are induced by different kind of bifurcation, including period doubling bifurcations. In particular there is an accumulation point for the period doubling bifurcations at $K_\infty \sim -0.642$ under which the dynamic is chaotic.
For our chaos control simulation parameters are: $K = -0.8$, $\alpha = 0.01$, $\tau = 0.001$, $\gamma = 0.001$. In order to have time scales similar to those of neural dynamics we

have scaled the time by $2.0 \cdot 10^5$ (not shown in equations). Figure 5.3 shows the projection of the chaotic attractor on the (u_1,v_1) plane.

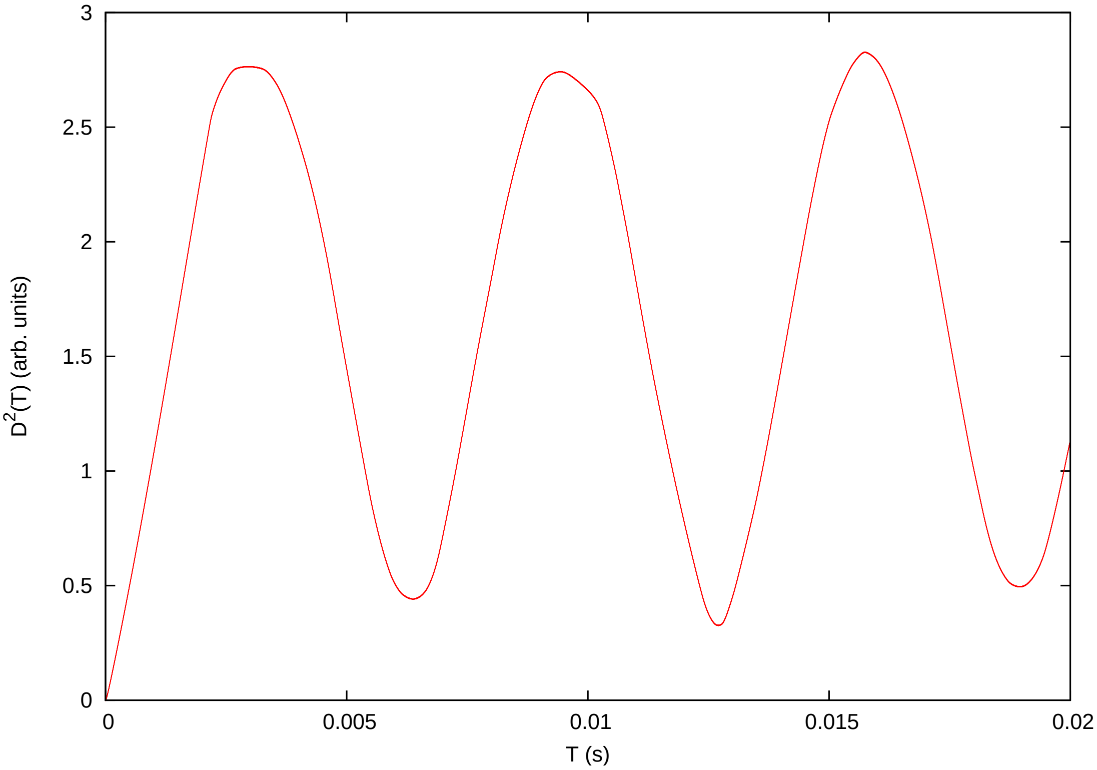


Figure 5.4: Behavior of $D^2(T) = \langle (u_1(t) - u_1(t-T))^2 \rangle$ as a function of T, during chaotic evolution. Minima of the plot approximate unstable periodic orbits periods, as the system shadows them continuously during its evolution.

To show how chaos control works and its robustness, we will stabilize period one and two cycles embedded in the system. The fact that the system shadows unstable periodic orbits allows for a very simple period recognition. We fix different trial values T, and for each of them we calculate the quantity $D^2(T) = \langle (u_1(t) - u_1(t-T))^2 \rangle$. The minima of this function will correspond to approximated values of unstable periodic orbits' period. Results for the current system are shown in figure 5.4. The first two minima are at $T_1 = 0.00638,\ T_2 = 0.0127$.

Equations for the controlled system are

$$\frac{du_1}{dt} = u_1(u_1 - \alpha)(1-u_1) - v_1 + \frac{K}{2}(u_2 - u_1) + u(t), \quad u(t) = K_0(u_1(t-T) - u_1(t)),$$

$$\frac{dv_1}{dt} = \tau(u_1 - \gamma v_1), \tag{5.11}$$

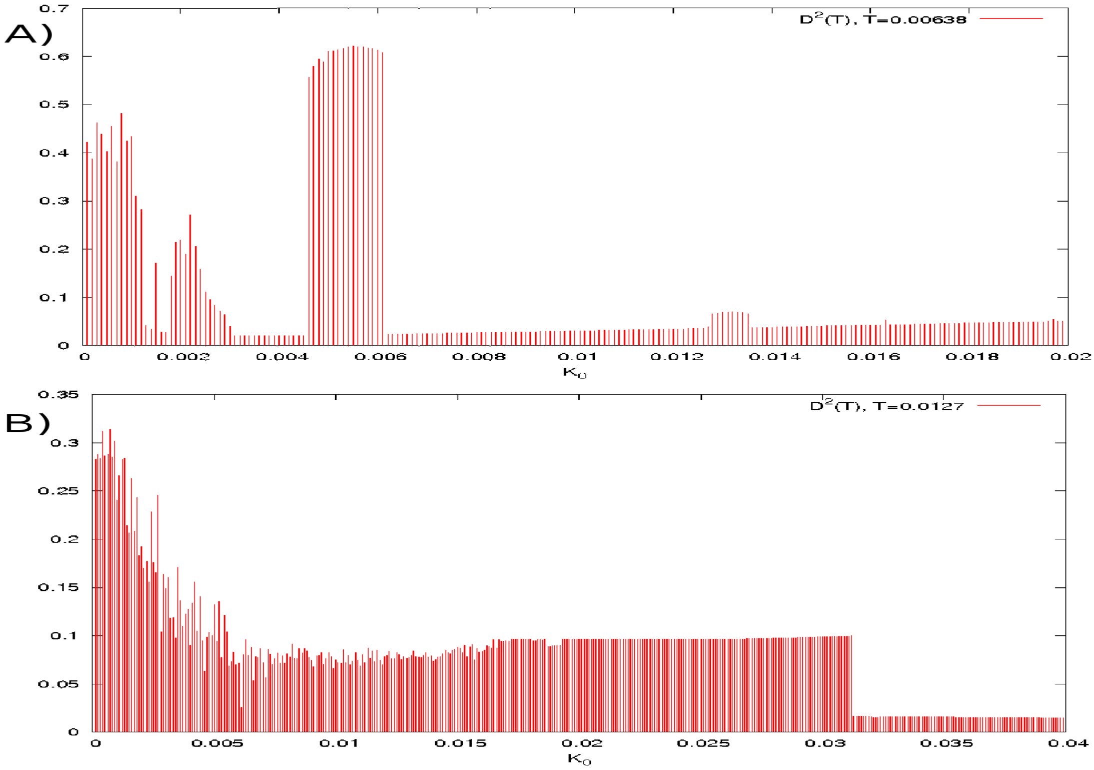


Figure 5.5: A) Behavior of $D^2(T)$, for $T = T_1$, as a function of the control gain K_0. There are two distinct regions in which the unstable periodic orbit of period T_1 turns to be stabilized: $K_0 \in [0.0031, 0.0045]$ and $K_0 \in [0.0063, 0.0126]$ (higher gains were not tested), which verifies the robustness of the controller against the gain parameter K_0. B) Behavior of $D^2(T)$, for $T = T_2$. The unstable periodic orbit of period 2 turns to be stabilized for $K_0 \in [0.0314, 0.0400]$ (higher gains were not tested), which verifies the robustness of the controller against the gain parameter K_0.

$$\frac{du_1}{dt} = u_2(u_2 - \alpha)(1 - u_2) - v_2 + \frac{K}{2}(u_1 - u_2),$$

$$\frac{dv_1}{dt} = \tau(u_2 - \gamma v_2).$$

First of all, we want to verify the robustness of the controller against the gain parameter, K_0. To do this we let the system evolve for 0.2 s without control and then we apply the control for growing values of K_0 and $T = T_1, T_2$. Then we let the system to stabilize for another 0.5 s and then calculate $D^2(T)$ for each value of K_0, as in figure 5.5. Both periods turn out to be stabilized for a wide range of gain parameter, confirming the robustness of the method. Figures 5.6-5.7 show the stabilized orbits for different values of K_0, and for $T = T_1$ and $T = T_2$ respectively.

To test the robustness against the control delay, T, for each cycle, we fix

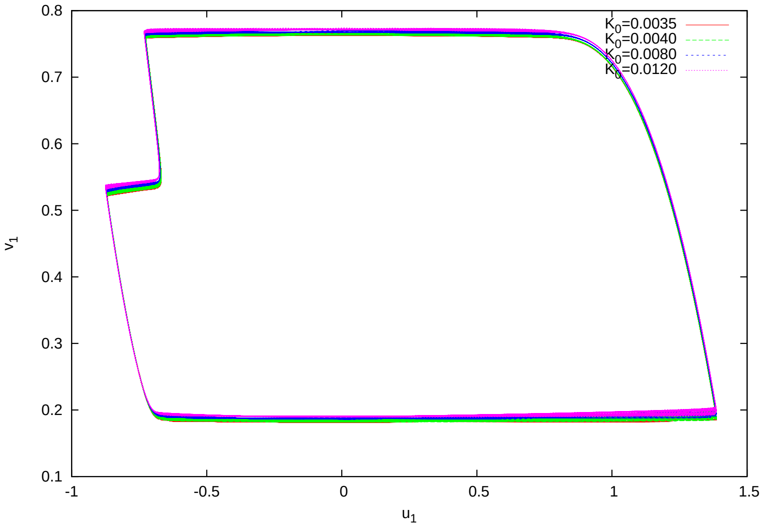


Figure 5.6: Projection of the unstable periodic orbit of period 0.00638, stabilized by different values of the gain parameter K_0

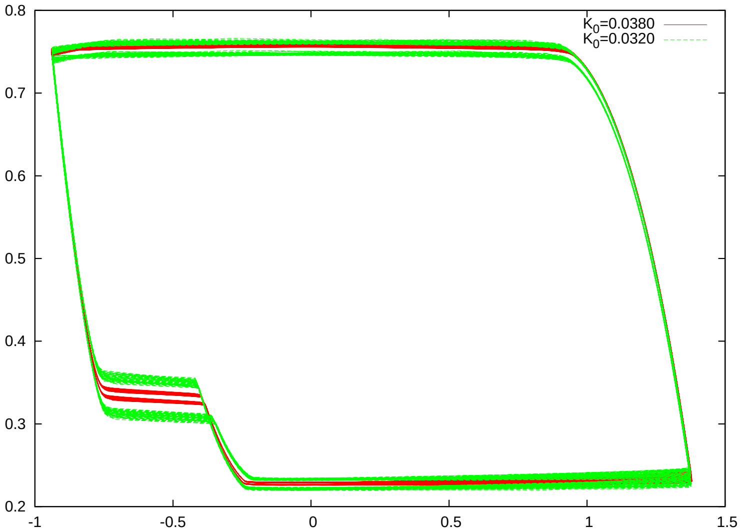


Figure 5.7: Projection of the unstable periodic orbit of period 0.0127, stabilized by different values of the gain parameter K_0

the control gain inside the stability region, as found in figure 5.5 (we test for $K_0 = 0.0038$ and $K_0 = 0.01$ for the cycle of period T_1, and $K_0 = 0.0350$ for the cycle of period T_2) and vary T. In this case, for each value of T, we calculate $D^2(\overline{T})$ where $\overline{T} = T_1, T_2$ is the period of the stabilized cycle. Looking at plot of $D^2(\overline{T})$ as a function of T (figure 5.8), gives an idea of which delays may succeed in stabilizing

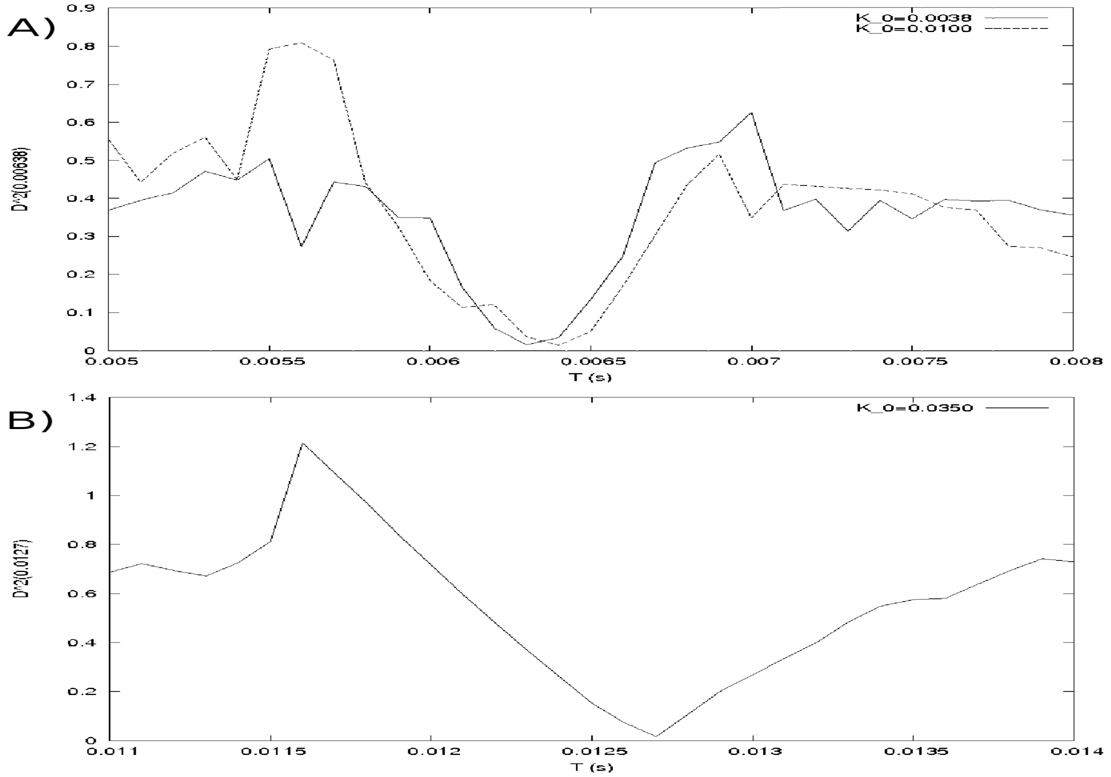


Figure 5.8: A) $D^2(\overline{T})$ for $\overline{T} = T_1$ calculated for the controlled system 5.11 for different values of the controller delay and two admissible control gains, as found in figure 5.5. The fact that the autocorrelation function remains quite small for a wide range of delays may indicate the robustness of the controller to this parameter. B) $D^2(\overline{T})$ for $\overline{T} = T_2$ calculated for the controlled system 5.11 for different values of the controller delay and two admissible control gain. The fact that the autocorrelation function remains quite small, compared to its value on the correct control period, for a wide range of delays may indicate the robustness of the controller to this parameter

the unstable periodic orbit. We fix $D^2(\overline{T}) < 0.15$ to consider a delay admissible. Figures 5.9-5.10-5.11 show some of the resulting orbits for different admissible control delays. The control of period T_1 with control gain $K_0 = 0.01$ turns to be not robust against delay uncertainties (figure 5.10), as the stabilized orbits falls in the neighborhood of the period T_2 cycle. The reasons of this phenomenon are still to be investigated. One possibility, is that, looking at the shape of the chaotic attractor (which looks like to pass more often near the T_2-cycle) and at the height of the minima of figure 5.4 (the T_2-minimum is lower then the T_1 one), the period T_1 orbit is more unstable than the period T_2 orbit. In all other cases the control is robust to delay time uncertainties (figures 5.9-5.11), as predicted by theoretical results of section 5.2.

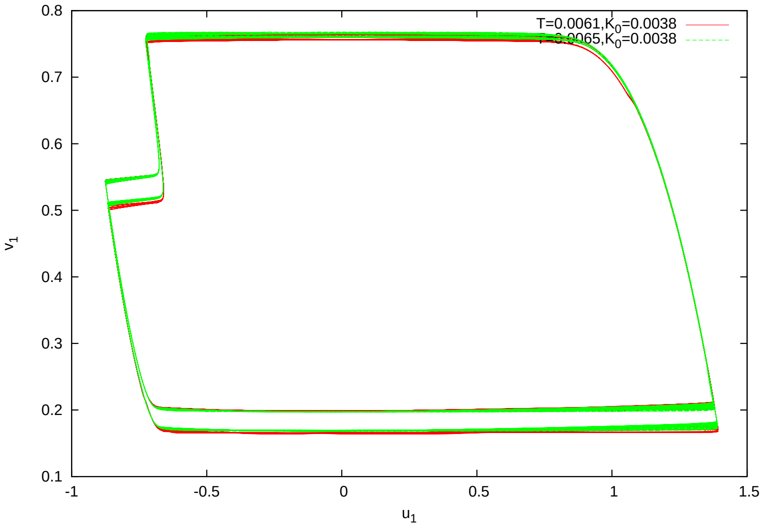


Figure 5.9: Control of the UPO of period 0.00638 s with delay times in the acceptable region found in figure 5.8 and $K_0 = 0.0038$. The control is robust.

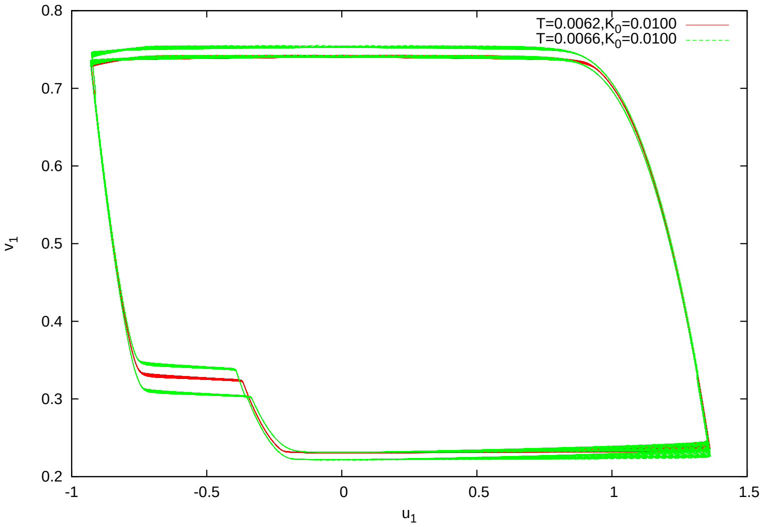


Figure 5.10: Control of the UPO of period 0.00638 s with delay times in the acceptable region found in figure 5.8 and $K_0 = 0.01$. The control is not robust, as the stabilized orbit falls near the UPO of period T_2.

5.4 Chaos synchronization

Synchronizing two identical chaotic systems with different initial conditions means linking the trajectory of one system to the same values of the other so that they remain in step with each other, through the transmission of a signal. Synchroniza-

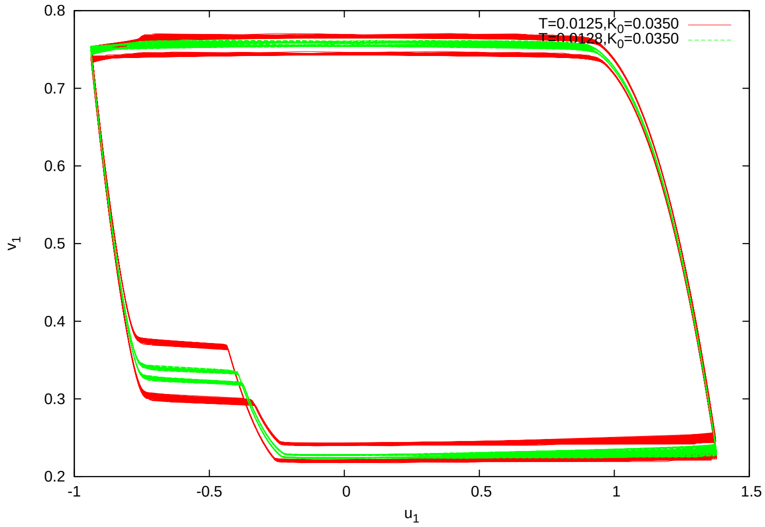


Figure 5.11: Control of the UPO of period 0.0127 s with delay times in the acceptable region found in figure 5.8 and $K_0 = 0.0350$. The control is robust.

tion can be used to transmit information ([77]), or in a set of neurons to create coherent states as a form of representation ([65]).

Chaos synchronization has practical application in robust and secure communication (see [75] and the references therein).

5.4.1 Coupled FitzHugh-Nagumo equations chaos synchronization through small perturbations

In what follows we will study the synchronization between two pairs of chaotic FHN equations. There will be a "guiding" and a "guided" pair, in the sense that the signal will be transmitted from a pair to the other and not vice-versa.

Equations of the system are

$$
\begin{aligned}
\frac{du_1}{dt} &= u_1(u_1-\alpha)(1-u_1) - v_1 + \frac{K}{2}(u_2-u_1), \\
\frac{dv_1}{dt} &= \tau(u_1 - \gamma v_1), \\
\frac{du_2}{dt} &= u_2(u_2-\alpha)(1-u_2) - v_2 + \frac{K}{2}(u_1-u_2), \\
\frac{dv_2}{dt} &= \tau(u_2 - \gamma v_2),
\end{aligned}
\tag{5.12}
$$

$$
\frac{du_3}{dt} = u_3(u_3-\alpha)(1-u_3) - v_3 + \frac{K}{2}(u_4-u_3) + U
$$

$$\frac{dv_3}{dt} = \tau(u_3 - \gamma v_3),$$

$$\frac{du_4}{dt} = u_4(u_4 - \alpha)(1 - u_4) - v_4 + \frac{K}{2}(u_3 - u_4),$$

$$\frac{dv_4}{dt} = \tau(u_4 - \gamma v_4),$$

where $U = \sigma(u_1 - u_3)$ is the control signal and σ is the adjustable signal intensity (coupling strength). All the other parameters are as above.

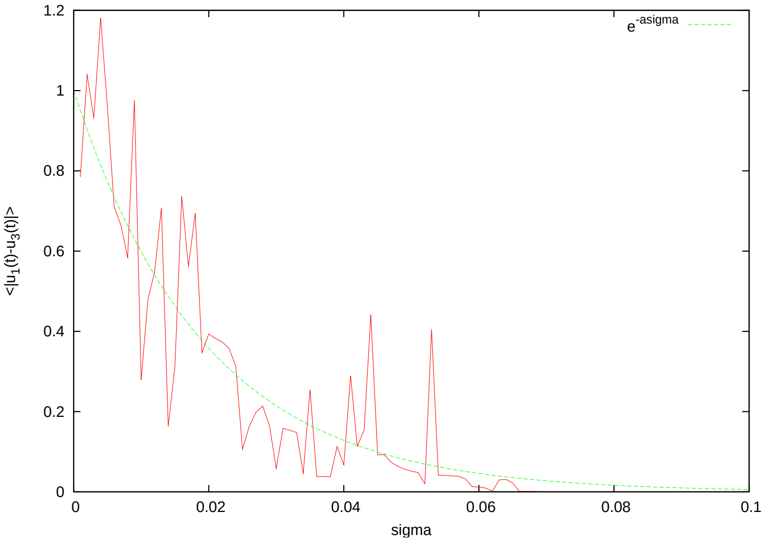


Figure 5.12: Synchronization as a function of the signal strength σ for two pairs of coupled chaotic FHN equations.

We study the synchronization in term of the mean absolute difference between u_1 and u_3 for various signal intensity ($\sigma \in [0, 1]$ with steps of 0.001) and after the system has stabilized. To do this we let the system evolve with $\sigma = 0$ for 0.2 s, then switch the signal on and let the system evolve for 0.8 s, then compute the mean of $u_1(t) - u_2(t)$, $\langle |u_1(t) - u_2(t)| \rangle$, for 0.2 s. Results are given in figure 5.12. We try to fit the plot with an inverse exponential function, $e^{-a\sigma}$, obtaining $a = 51 \pm 3$, (5%) (fig. 5.12). Again, small perturbations, compared to the unperturbed dynamics, suffice to obtain synchronization.

Chapter 6

The proposed model revisited and conclusions

In chapter 1 the general problem of adaptive control was faced without any technical instrument. In chapter 2, inspired by movement learning, we propose a vertebrate-like adaptive controller, but there were still open problems. In chapter 3-5, theoretical instruments to face these problems were presented, including some practical examples.
Based on these results, we want to synthesize a possible ISS vertebrate like adaptive control system. We will use both an analogical control scheme, which can be seen as a "biological" model, and a digital one, more useful for practical implementations, and, as we will see, even more efficient for fine parameter tuning.
We will focus only on the "feedback loop" and on the "adaptive loop" of figure 2.8. That is regarding possible ways of implementing a synergy ISS feedback controller and on open problems and possible solutions regarding adaptation through synergy activation coefficient tuning.

6.1 The ISS synergy feedback controller and adaptation problems

For both an analogical and a digital controller, we choose the synergy summation field (2.5) to be ISS, that is, if $\nu_0 \in \Re^p$ is our present knowledge of system's parameters we have to find a function, $k_{\nu_0}(x)$, such that the closed loop system $\dot{x} = f(x, k_{\nu_0}(x), \nu_0)$ is ISS at the origin. This way we are sure that, even if the real parameter vector, μ, differs from ν_0, the system can be contained. Depending on hardware capabilities, we can split this force field in one or more synergies. Our aim is to modify their activation coefficients to obtain adaptation through a suitable algorithm.

In general, the set of parameter vectors we can approximate by modifying synergy activation coefficients will depend on the number and on the specific form of available synergies. In fact, depending on these two factors it will be possible to approximate more or less precisely the full parameter phase space.
Let us make it clear with an example. Suppose that, for a fixed initial parameter vector, ν_0, we can split the feedback force field, $k_{\nu_0}(x)$, in K synergies, $\{\phi_i(x)\}_{i=1,\dots,K}$, of arbitrary form (i.e arbitrary x dependence), with the constraints that: (i) they add vectorially and their vectorial summation with the initial activation coefficient set $\{c_i^{\nu_0}\}_{i=1,\dots,K}$ gives $k_{\nu_0}(x)$ (ii) K is fixed, (iii) we cannot further modify their form, apart from their activation coefficients, $\{c_i\}_{i=1,\dots,K}$. These constraints are equivalent to the fact that we can set the synergy feedback controller as we want, depending on hardware capabilities, before starting to control the system, but we cannot further modify it after the control starts.
The easiest case we can imagine is when there is only one unknown parameter. Indicating with $[\nu]_j$ the j-th component of a vector $\nu \in \Re^p$, this means $[\nu_0]_j = [\mu]_j$ for $j = 1, \dots, p-1$, $[\nu_0]_p \neq [\mu]_p$, and $[\mu]_p$ is unknown. Then one synergy, given by $k_{\nu_0}(x)$, is sufficient only if the ISS force field depends linearly on $[\nu_0]_p$, that is if $k_{\nu_0}(x) = [\nu_0]_p \cdot k'_{[\nu_0]_1,\dots,[\nu_0]_{(p-1)}}(x)$. In fact it is sufficient to set its coefficient to $[\mu]_p/[\nu_0]_p$, through a suitable tuning algorithm. But, slightly complicating things, if $k_{\nu_0}(x) = k'_{[\nu_0]_1,\dots,[\nu_0]_{(p-1)}}(x) + [\nu_0]_p \cdot k''(x)$, affine dependence, then, in order to not modify the other parameter dependence, we would need two synergies, $k'_{[\nu_0]_1,\dots,[\nu_0]_{(p-1)}}(x)$ and $[\nu_0]_p \cdot k''(x)$, and then set the coefficient of the first to 1 and the coefficient of the second to $[\mu]_p/[\nu_0]_p$. If the hardware does not permit it and we can use only one synergy, necessarily given by $k_{\nu_0}(x)$, then we can not blindly set its activation coefficient to $[\mu]_p/[\nu_0]_p$, because in this way we will also modify the dependence of the feedback law on the other parameters.
To formalize this problem, we can describe the synergy activation pattern $\{c_i\}_{i=1,\dots,K}$ as a vector $c \in \Re^K$ in an obvious manner. Depending on the particular form of the ISS feedback force field, and on the number and forms of synergies we are using, there will exist a map

$$M : \Re^K \to \Re^p,$$

which describes the effect of a change in synergy activation on the parameter vector components. M will in general depend on x, too. In the case of linear dependence of only one unknown parameter explained above we will simply have, $M(c) = ([\nu_0]_1, \dots, [\nu_0]_{(p-1)}, c \cdot [\nu_0]_p)$. But, even for the affine case with only one synergy, the exact form of M will change from case to case, and we can say nothing in general.
It would be important that M does not depend on x, otherwise a change in synergy

activation will induce a position dependent change in the parameter vector [1]. In what follows, we then make the following assumption.

***Assumption* 10** *M is independent of x.*

Going back to the affine dependence case with only one synergy explained above, or to any other general case, with the last assumption we can define the vector $\nu = M(c)$. The tuning will then consist in finding the vector, $c \in \Re^K$, that minimizes $\|\nu - \mu\|_\infty$ through a suitable tuning algorithm, even if in general we will not be able to obtain $\nu = \mu$.

6.1.1 Analogical synergy feedback controller

This subsection is an attempt to synthesize an analogical adaptive controller as a model of vertebrate movement stabilization and adaptation through spinal force fields and reflexes.

In this case, the feedback force field is given by the co-contraction of different muscles, and is therefore analogical, but we suppose that its output is affected by strong disturbances, possibly of the same order of intensity of the force field itself. Moreover, as we do not need state quantization, there will not be an allowed parameter set Ω_0 estimation procedure, as in section 3.2.6. In fact, we don't even need to specify an acceptable state region, as, for the ISS nature of the closed loop dynamics, the system will be contained for any couple of real and trial parameter vectors, and for any input perturbations (in the case of the digital controller it was necessary to define a ball of acceptable states, $B^n_\infty(0, R)$, in order to unambiguously define the Lipschitz constant of $f(x, u, \mu)$ and correctly synthesize the adaptive controller). Once we have split the ISS feedback law, $k_\nu(x)$ in different synergies, $\{\phi_i(x)\}_{i=1,\dots,K}$, we have to define how to tune their activation, in a similar way to vertebrate reflex adaptation. First, we have to define a measured variable $\tilde{x} = m(x) \in \Re^s$, in continuous time or sampled, which should describe divergences from the origin (see section 6.1.1 for a practical example) and an admissible region, $\mathcal{A} \subset \Re^s$, for it. This variable mimics the function of muscle

[1] If M depends on x and $k_\nu(x) = k(x, \nu)$ has smooth dependence on ν, we can use a trick, that is Taylor expansion in the space of the $p' \le p$ unknown parameters. We can fix a precision, ϵ, for the unknown parameter set we want to explore, and then find the relative Taylor expansion order and expansion terms around ν_0. Then choose as fixed coefficient synergy $k_{\nu_0}(x)$ and as tunable coefficient synergies the expansion terms with null initial coefficients. M will be determined by the coefficients of the first order terms $c_i = [\delta\mu]_i$, $i = 1, \dots, p'$, as $M(c_1, \dots, c_{p'}) = ([\nu_0]_1 + c_1 \pm \epsilon, \dots, [\nu_0]_{p'} + c_{p'} \pm \epsilon, [\nu_0]_{(p'+1)}, \dots, [\nu_0]_p)^T$. In real cases we want K to be small, and so ϵ will be large. What is important is that the parameter approximation, which will depend on the particular system under consideration, due to this fact does not lead to instabilities for the ISS nature of the feedback force field.

spindles. In the case it leaves this region, an error signal (sensory inflow) is sent to the synergy activation tuner, which will change their coefficient, after a certain delay, on the basis of a suitable algorithm (sensory feedback).

We may want to implement the representation of synergy activation coefficient through the state of a chaotic system. In this case the error signal will change the control parameter of a chaos control scheme, inducing the transition from one periodic orbit to another. Alternatively, it can change the synchronization, for example by changing the coupling strength, between groups of neurons, to obtain spatial-temporal coherence as a form of representation. A simple, but effective idea is to use UPOs stabilization to represent the *coarse* feedforward activation coefficient set, due to chaos control robustness. While synchronization could be used for fine reflex tuning, as it is achieved faster, is easy to implement and leads to smooth responses, for example to changes in the coupling strength.

As happens for vertebrate movement learning ([62]), adaptation through reflexes works only if the present knowledge of system parameters does not suffice to stabilize the origin, or, for movement learning, the instantaneous virtual trajectory point. In the case the system turns out to be already stabilized, this scheme can do nothing, and different adaptation algorithms, such as metabolic optimization of power consumption, should be implemented to achieve finer parameter learning.

The inverted pendulum on a cart

We want to test our analogical synergy adaptive controller on the same mechanical system of section 3.2.7. We recall the equations of motion

$$\dot{x} = y$$

$$\dot{y} = \frac{1}{l}\frac{(m+M)}{(m\sin(x)^2+M)}\left(g\sin(x) - \frac{(ml)^2}{(m+M)}\cos(x)\sin(x)y^2 - \frac{\cos(x)}{(m+M)}u\right).$$

and relative nonlinear control law, which renders the closed-loop system still Lagrangian is

$$u(x,y) = k\frac{ml\sin(x)(ml^2y^2+\cos(x)D)}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)} + \\ -\frac{c\left(ml^2 - \frac{(ml)^2}{m+M}\cos^2(x)\right)((r(k+1)+1)ml\cos(x)y)}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)}, \tag{6.1}$$

where $D = -mgl$.

With no state quantization the origin is an asymptotically stable fixed point for any couple of real and trial parameter vectors, but the case in which the real acceleration due to gravity, g, is larger than the one used by the controller, g'.

Thus, we will assume that it is the only unknown parameter.
We can devise the feedback law in two terms, which add vectorially, one, $\phi_s(x)$ that depends on g' and one, $\phi_d(x)$ that does not. Their explicit form is

$$\phi_s(x) = g' \frac{k(ml)^2 \sin(x) \cos(x)}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)}, \tag{6.2}$$

$$\phi_d(x) = k \frac{ml \sin(x) ml^2 y^2}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)} + \\ - \frac{c\left(ml^2 - \frac{(ml)^2}{m+M}\cos^2(x)\right)((r(k+1)+1)ml\cos(x)y)}{ml^2 - \frac{(ml)^2}{m+M}(1+k)\cos^2(x)}. \tag{6.3}$$

From the force field plot of these two terms, we see that the first is a *stabilizing* force field , that is negative force for positive angle, and positive force for negative angle, and it serves to stabilize the system (fig. 6.1 top right), while the second is a *dissipating* force field, that is negative acceleration for positive speed, and positive acceleration for negative speed, and it serves to obtain asymptotic stability (fig. 6.1 bottom).
When the controller uses $g' \geq g$, the origin is the only asymptotically stable fixed point (fig. 6.1 center left and fig. 6.2), while, when it uses a $g' < g$ and $\left(\frac{m+M}{m}\frac{g}{(k+1)g-kg'}\right)^{1/2} < 1$, the origin becomes unstable and two new fixed points appears for $x = \pm\overline{x}(g', g)$, $\overline{x} = cos^{-1}\left(\left(\frac{m+M}{m}\frac{g}{(k+1)g-kg'}\right)^{1/2}\right)$ (see figure 6.1 center right and figure 6.3).
Now we have to define a suitable measured variable which should describe divergences from the origin. We decide to use a sampled variable with memory, with sampling time τ. In this case divergences from the origin occur in two cases: (i) the system is leaving the origin ($x \cdot y > 0$) and the absolute value of the speed, $|y|$, is not decreasing; (ii) the speed is asymptotically going to zero for $|x| > 0$ (that is the origin is not the only asymptotically stable fixed point). Hence we define the measured variable $\tilde{x}(j\tau) \in \Re^4$ as follows

$$\tilde{x}(j\tau) = (\ x(j\tau)y(j\tau),\ |y(j\tau)| - |y((j-1)\tau)|,\ |x(j\tau)|,\ |y(j\tau)|)\,.$$

The admissible region, $\mathcal{A} \subset \Re^4$ for it is given by

$$\mathcal{A} = (\ (\tilde{x}(j\tau)_1 \leq \tilde{s}_1)\ \cup\ (\ (\tilde{x}(j\tau)_1 > \tilde{s}_1)\ \cap\ (\tilde{x}(j\tau)_2 < \tilde{s}_2)\ )\ ) \cap$$

$$(\ (\tilde{x}(j\tau)_3 \leq \tilde{s}_3)\ \cup\ (\ (\tilde{x}(j\tau)_3 > \tilde{s}_3)\ \cap\ (\tilde{x}(j\tau)_4 \geq \tilde{s}_4)\ )\ ) \cap \overline{\mathcal{A}}, \tag{6.4}$$

where $\overline{\mathcal{A}} = [-1,1] \times [-1,1] \times [0,1] \times [0,1]$ is the admissible phase space for $\tilde{x}$ induced by the fact that $(x,y) \in B^2_\infty(0,1)$, $\tilde{s}_i \geq 0$, $i = 1,2$, are fixed tolerance levels to avoid disturbance effects and in the noise free case they are set to zero. While $\tilde{s}_3 \ll 1$, $\tilde{s}_4 \ll 1$ determine when we are close to the origin and under which value the angular speed should be consider as zero, respectively. Let us explain in the disturbance free case, $\tilde{s}_i \equiv 0$, $i = 1,2$, the meaning of $\mathcal{A}$.
The first line tells us that the system is stable, but not necessarily at the origin. In fact $(\tilde{x}(j\tau)_1 = \dot{x}^2 \leq 0)$ means that the system is not leaving the origin, and $(\ (\tilde{x}(j\tau)_1 > 0) \cap (\tilde{x}(j\tau)_2 < 0\ )$ means that, even if the system is leaving the origin, the absolute value of the angular velocity is decreasing, and, by the ISS nature of the force field, which implies boundness of trajectories, it will change sign in finite time.

The second line tells us the origin is the only stable fixed point. In fact $(\tilde{x}(j\tau)_3 \leq \tilde{s}_3)$ means we are close to the origin, while, $(\tilde{x}(j\tau)_4 \geq \tilde{s}_4)$) means that the system is allowed to stay far from it only with non-zero angular velocity, except when it changes sign. The tolerance level $\tilde{s}_4 \ll 1$ serves to discriminate whether the system is just passing through $y = 0$ to change the sign of the angular velocity, or if it is asymptotically stabilizing on a fixed point different from the origin. By choosing an arbitrary small $\tilde{s}_4 \ll 1$, in the first case the probability of sampling $\tilde{x}$ outside $\mathcal{A}$ goes to zero, while in the second case, $\tilde{x}(t)$ will remain outside $\mathcal{A}$ until it is sampled and an error signal is correctly sent, regardless how small $\tilde{s}_4 \ll 1$ is, as long as it is finite. In the perturbed case, we will have to choose $\tilde{s}_i \geq 0$, $i = 1,2$ and, possibly, change $\tilde{s}_3$, $\tilde{s}_4$ in accordance to the perturbation intensity. In particular it will be necessary to set a positive $\tilde{s}_1$, as there will be positive value of $\tilde{x}_1$ induced by the perturbation; and it will be necessary to increase the value of $\tilde{s}_4$, as the perturbation will tend to keep $\tilde{x}_4$ outside the error region.
Now we have to define the error signal. Let $\mathcal{A}^C = \overline{\mathcal{A}}/\mathcal{A}$ be the complementary set of $\mathcal{A}$ with respect to $\overline{\mathcal{A}}$. Then $\mathcal{A}^C = \mathcal{A}^C_1 \cup \mathcal{A}^C_2$, where $\mathcal{A}^C_1$ is the complementary to the set described in the first line of 6.4 and $\mathcal{A}^C_2$ the complementary to the set described in the second. Thus, at each sampling time, we can independently check first if $\tilde{x} \in \mathcal{A}^C_1$, which tells us wether the system is leaving the origin, and then if $\tilde{x} \in \mathcal{A}^C_2$, which tells us that the origin is not the only stable fixed point. In the former case we will send an error signal r_1, in the latter r_2. Obviously we can have both signals.
As we said, in this case modifying the dependence of the feedback law on other parameters than g' will not lead to instabilities. That is why we can use only one synergy, which is given by the full ISS feedback force field 6.1. As we said, $\phi_d(x)$ is independent of g', while $\phi_s(x)$ depends linearly on it, and in this case, by setting $\nu_0 = (l, g', \text{other parameter})^T$, it results $M(c) = (c{\cdot}l, c{\cdot}g', \text{other parameters unchanged})^T$,

independent of x. So in the case of error signal we have to increase the synergy activation.

We want to include results of section 5.4 in the synergy tuning procedure. Without specifying an exact combination of periodic orbits to represent their feedforward activation, we want to use chaos synchronization to modify it through a reflex gain $r \in [0,1]$. If c is the feed forward activation, $c(r) = c(1 + \alpha \cdot r)$ is the reflex tuned activation, where α is the maximum percentual reflex boost. At the beginning $r = 0$, while in general we choose it to be proportional to the synchronization of two coupled pairs of chaotic FitzHugh-Nagumo equations. If σ is the coupling strength, from results of section 5.4 it results $r(\sigma) \sim 1 - e^{-a\sigma}$. The role of the error signals r_1 and r_2 is to increase σ by $\delta\sigma_1$ and $\delta\sigma_2$, respectively. At the end this is the proposed algorithm:

1. set the synergy force field to be $\phi_d(x) + \phi_s(x)$ with parameter g' and activate it with initial coefficient 1 ($K = 1$)

2. let the system start to evolve

3. sample $\tilde{x}$ each τs

4. check if $\tilde{x}(j\tau) \in \mathcal{A}_1^C$, in which case send an r_1 signal to the synergy activation tuner

5. check if $\tilde{x}(j\tau) \in \mathcal{A}_2^C$, in which case send an r_2 signal to the synergy activation tuner

6. after a fixed delay T, the coupling σ is increased by $\delta\sigma_1$, in the case of error signal r_1 and by $\delta\sigma_2$, in the case of error signal r_2, inducing a reflex boosting of the synergy activation

7. repeat step 1-6 until one of the following two

   (a) the origin is asymptotically stabilized

   (b) the reflex gain has gone to 1 without asymptotically stabilizing the origin

8. in the first case the synergy is correctly tuned

9. in the second start again from step 1, with initial feedforward coefficient $(1 + \alpha)$ and $r = 0$

We use the following parameters in simulations (see figures 6.4-6.6): $g' = 9\text{ms}^{-2}$, $g = 14\text{ms}^{-2}$, $\tau = 0.005\text{s}$, $\alpha = 1$, $a = 50$, $\delta\sigma_1 = 0.001$, $\delta\sigma_1 = 0.003$, $T = 0.2$. In the disturbance free case $\tilde{s}_1 = \tilde{s}_2 = 0$, $\tilde{s}_3 = 0.01$ and $\tilde{s}_4 = 0.00001$.

In the perturbed case the output of the feedback controller is perturbed by two kind of disturbances: (i) a high frequency, ω_1, periodic disturbance of constant intensity, D; (ii) a lower frequency, ω_2, periodic disturbance of intensity proportional to the unperturbed control intensity, with constant proportionality coefficient α. The first kind of disturbance models fine movement control noise (a shaking hand). The second tends to be larger whenever the unperturbed control is large, and models the fact the muscles may control large forces worst than small ones.
In the simulation we choose $\omega_1 = 7$Hz, $\omega_2 = 5$Hz, $\tilde{s}_1 = 0.05$, $\tilde{s}_2 = 0$, $\tilde{s}_3 = 0.01$ and $\tilde{s}_4 = 0.002$ and for different values of $D = 0.5, 1, 2$ and $\alpha = 0.05, 0.1, 0.2$. The frequencies of the disturbances are similar to those of a real shaking hand ([84]).
In both cases, non perturbed and perturbed, the algorithm works to correctly tune the synergy. In the disturbance free simulation, it is possible to clearly distinguish the effect of the two different kinds of error signal. In the perturbed one, unpredictable effects of disturbances lead to a small synergy over boosting. As we said, finer and higher level tuning procedures, similar to animal metabolic optimization of power consumption, can be used to correct this problem.

6.1.2 The digital controller with minimal bit rate

The use of an analogical controller can be very unpractical in real situations, and, as we explained, its tuning algorithm only suffices to stabilize the origin, but is very poor for fine parameter approximation. Here we want to use properties of both the synergy controller and the digital controller with minimal bit-rate described in section 3.2.5.
After we have found the unknown parameter region which assure containability of the system, $\Omega_0 \subset \Re^p$, we choose an initial trial parameter $\nu \in \Omega_0$, and define the ISS feedback law (in this case we have to impose slightly different conditions on vectors in Ω_0, than what we imposed in section 3.2.6, we will come back on this point later).
What we said about the approximation of the real parameter vector through the combination of tuned synergies is still valid. Nevertheless, in this case we are not allowed to modify the dependence of the feedback law on parameter we have not included in the unknown ones. Thus, if we can choose the synergy set $\{\phi_i(x)\}_{i=1,\ldots,K}$ such that we can tune them without modifying dependence on known parameters, there are no problems and we can start the tuning procedure. In the opposite case we have to include those known parameters, whose dependence is changed by synergy tuning, with the unknown ones. In both cases, there will be an allowed parameter region, Ω_0, which induces an allowed activation coefficient region, $C_0 \subset \Re^K$, where K is the number of synergies we are using, $\Omega_0 = M(C_0)$, and M is defined in section 6.1.
Once we have done it, the digital controller output allows for a very powerful tun-

ing procedure. From what explained in Appendix A, the mean number of positive symbols we get from the encoder, $\overline{h}$, is a decreasing function of the difference between actual and trial parameters, $\Delta\mu = \|\nu - \mu\|_\infty$.
We can use this fact to explore the admissible parameter region and find for which parameter vector $\overline{h}$ is maximized
Thus, in this case, the output of the sensors will just be an empirical estimation of $\overline{h}$, $\tilde{h}(c)$, where $c \in C_0$ and $\nu = M(c)$ is the present trial parameter, and there will not be an allowed region. The tuning procedure will then consist in moving the trial parameter, by changing synergy activation in C_0 and stop on the activation pattern, $\overline{c} \in C_0$, which maximizes $\tilde{h}$. The computation of $\tilde{h}$ will be done by fixing a large number of sampling times, $\tilde{S} \gg 1$, and by counting the number of zeros from the encoder $N_0(c)$ in these samples. Then $\tilde{h}(c)$ can be found by solving the following equation

$$\frac{N_0(c)}{\tilde{S}} = \frac{1}{\tilde{h}(c) + 1}.$$

If c^ν is the initial activation pattern, first we compute $\tilde{h}(c^\nu)$, then we start to explore C_0 with a certain resolution δc, that is we fix a constant grid $\mathcal{G} \ni c^\nu$ over C_0 of diameter δc and start to explore all of its node until we find the activation pattern $\overline{c} \in C_0 \cap \mathcal{G}$, which maximizes $\tilde{h}$.
As we said, in computing the allowed parameter region, Ω_0, we have to impose different condition than section 3.2.6. In fact, in this case the trial parameter is not fixed, so relation 3.55 has to be verified for each real parameter vector $\mu \in \Omega_0$, but now not just for a *fixed* trial parameter vector $\nu \in \Omega_0$ but *for all* possible trial parameter in Ω_0, that is condition 3.55 becomes

$$d(Q_{\mu\nu}) < R - \overline{E}, \quad \forall \mu, \nu \in \Omega_0. \tag{6.5}$$

In the old learning scheme the central controller received the codified variable, and the relative control, at each sampling time, and from this could learn in some way (neural network, Kalman filter, ...) the inverse dynamics of the system, so that the trial parameter could remain fixed, until the correct one has been learned and the feedback law is consequently tuned, while now we want to implement a faster adaptation algorithm directly in the (synergy) feedback controller with minimal bit-rate.

The inverted pendulum on a cart

As in section 3.2.7, we assume that the only unknown parameter is the pendulum length, l expressed in meters. Without complicating things too much, this time we will use two synergies, $\phi_s(x)$ independent of l, and $\phi_d(x)$ which depends linearly on l. Hence if ν is the initial length parameter and $c_s \equiv 1$, then $M(c_d) = (c_d \cdot$

$\nu,$other parameters unchanged$)^T$, independent of x.
As we mentioned, to find the allowed parameter set, Ω_0, we cannot follow the algorithm of section 3.2.7. In that case we wanted to find the largest possible set which could contain the real length parameter μ given a *fixed* trial parameter ν, while now the trial parameter has to move in order to maximize $\overline{h}$, that is why the system must be containable *for each* trial and real parameter in Ω_0. To make this possible we use a similar algorithm. We define step 3 to 7 of the old algorithm followed by

- if $(\overline{y} > 1 - \overline{E})$ and Assumption 7 is verified, then give 1 as result; else 0

as the function $\mathcal{T}(\mu, \nu)$. This function gives 1 if the system with real length parameter μ can be contained with trial parameter ν, and 0 otherwise.
For a fix length-axis resolution δ_μ, suppose we have found a set of r value, $\Omega_{\text{temp}} = \{\Omega_m, \Omega_m + \delta_\mu, \ldots, \Omega_m + (r-1)\delta_\mu\}$, for which $\mathcal{T}(\mu, \nu)$ gives 1 for each $\mu, \nu \in \Omega_{\text{temp}}$. Then to enlarge this set for larger values (right side), we use the following algorithm:

1. set $\nu = \Omega_m + r\delta_\mu$
2. for $(i = 1, \ldots, r)$ do ($\mathcal{T}(\Omega_m + (i-1)\delta_\mu, \nu)$, $\mathcal{T}(\nu, \Omega_m + (i-1)\delta_\mu)$)
3. if in the last step we get a zero, we cannot enlarge the admissible parameter set for a larger value; else we set $r \longrightarrow r + 1$

while to enlarge it for a smaller value (left side), we use this

1. set $\nu = \Omega_m - r\delta_\mu$
2. for $(i = 1, \ldots, r)$ do ($\mathcal{T}(\Omega_m + (i-1)\delta_\mu, \nu)$, $\mathcal{T}(\nu, \Omega_m + (i-1)\delta_\mu)$)
3. if in the last step we get a zero, we cannot enlarge the admissible parameter set for a smaller value; else we set $r \longrightarrow r + 1$, $\Omega_m = \nu$

At the beginning we have $\Omega_{\text{temp}} = \nu_0$, $r = 1$, and we start to try to enlarge it alternatively for a larger and a smaller value, until we get an error for one of the two sides and we continue with the other until an error occur. For our system, we set initial trial parameter $\nu_0 = 1.25$ and $\delta_\mu = 0.01$. At the end we got $\Omega_m = 1.15$, $r = 22$, thus $\Omega_0 = [1.15, 1.36]$ with resolution 0.01.
This set corresponds to an allowed $\phi_d(x)$ synergy activation region $C_0 = [1.15/1.25, 1.36/1.25]$ with resolution 0.01/1.25.

The following simulation (figure 6.7) shows the effectiveness of this method. We measured $N_0(\nu)/\tilde{S}$, for $\tilde{S} = 5000$ and trial length parameter $\nu \in \Omega_0$, and different real length parameter $\mu \in \{1.15, 1.25, 1.35\}$. In each case the minimum corresponds exactly to the real parameter. For comparison, we also plot the same value as computed from Appendix A. In this case we get higher values, as in Appendix A we computed an upper bound for the typical error, thus a lower bound on $\overline{h}$ and an upper bound on $N_0/\tilde{S}$, but the similarity in the shape of the two plot is striking. Again, it confirms the correctness of our predictions.

6.2 Conclusions

The main problem of this work was to study stability properties of non-linear adaptive control systems with communication constraints. To do this we used two approaches: (i) the first one was an heuristic approach, based on the way vertebrates learn to control complex limb movements, which gave us the idea of the synergy controller activated by feedforward commands, along with its adaptation through reflexes, and the idea of the neural representation based on chaos control ; (ii) the second one was a technical approach, based on the Input-to-State Stability theory, which let us safely face problems such as control of non-linear dynamics also in the case of bandwidth limitation and parameter uncertainties. Nevertheless, the first approach was also the source of a new problem, that is chaos control stability.

At the end we built a formal framework, based on ISS, to study the stability of closed loop dynamics (sections 3.2.1-3.2.4) which let us solve the containability problem for adaptive control system with bandwidth limitation (Theorem 2 and section 3.2.6), and test these results in a practical situation (section 3.2.7). This framework also let us study stability properties of the proposed vertebrate-like adaptive control of section 2.4. In particular we were able to prove the (local) Input-to-State Stability of chaos control (Theorem 5), which justifies the robustness of neural representations based on it, and to test this result in a practical situation (section 5.3.1). Finally, in this last chapter we studied consequences of synergy activation tuning on ISS closed loop dynamics for both an analogical (biological model) and a digital (practical implementation model) controller, and the relative tuning algorithm. While for the analogical controller the tuning algorithm is just a rude emulation of vertebrate sensory feedback, for the digital controller statistical properties of its measurements allow for fine parameter approximation.

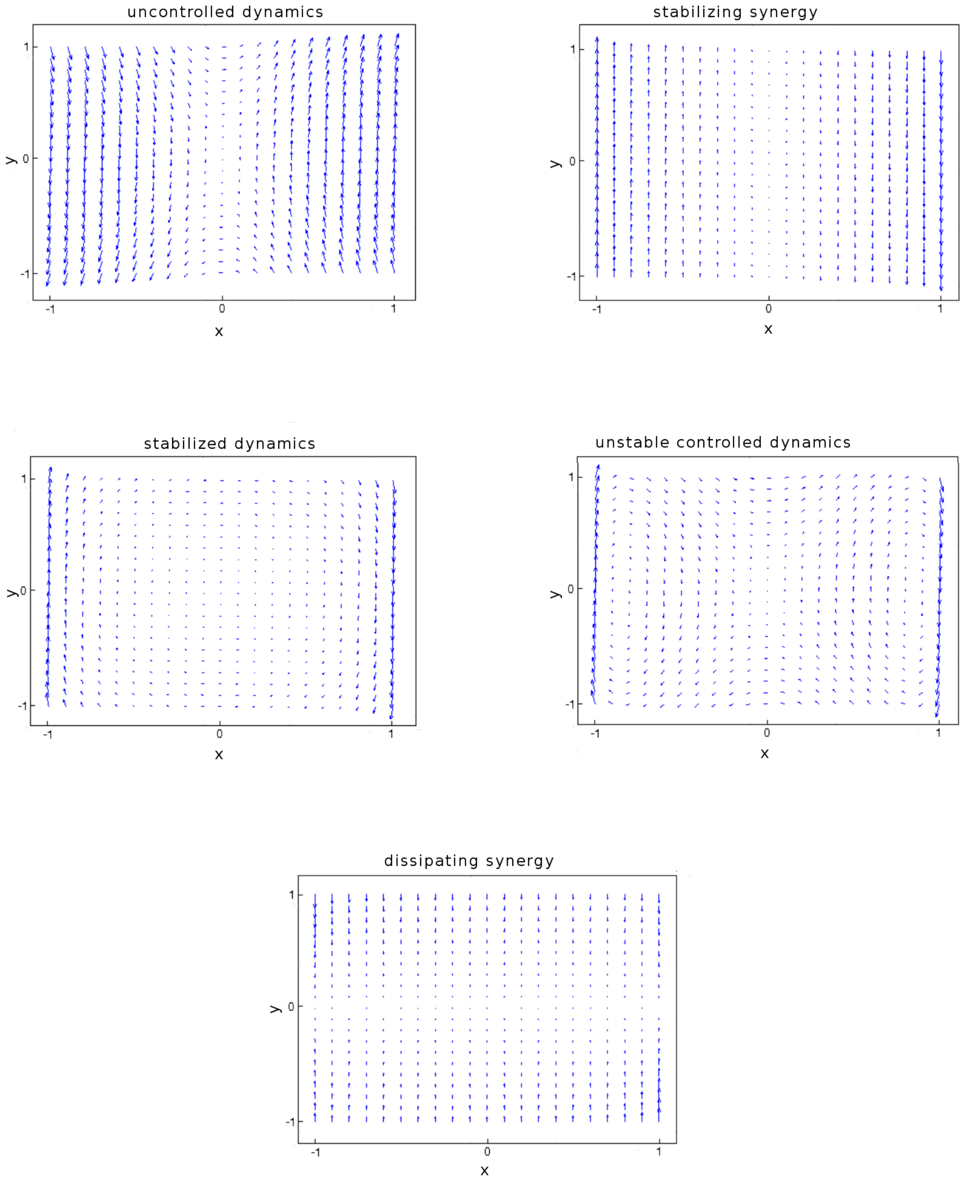


Figure 6.1: Top left: the force fields of the uncontrolled dynamic; Top right: force field of the stabilizing synergy; Center left: closed-loop system force field controlled only with the stabilizing synergy and $g' = g$; Center right: force field of the dynamic controlled only with the stabilizing synergy and $g' < g$. When the stabilizing synergy uses the correct value of the gravity acceleration, the origin is the only stable fixed point, which will then be rendered asymptotically stable through the dissipating synergy. When gravity acceleration is increased, without increasing controller parameter, the origin becomes unstable, and two new fixed points appear.

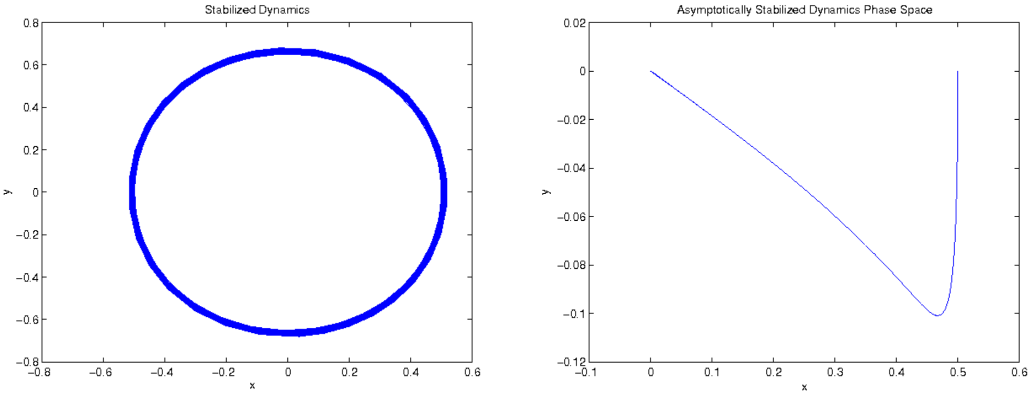


Figure 6.2: Stable dynamics phase space ($g = g'$). Left: only the stabilizing synergy is used. Right: both the dissipating and the stabilizing synergies are used.

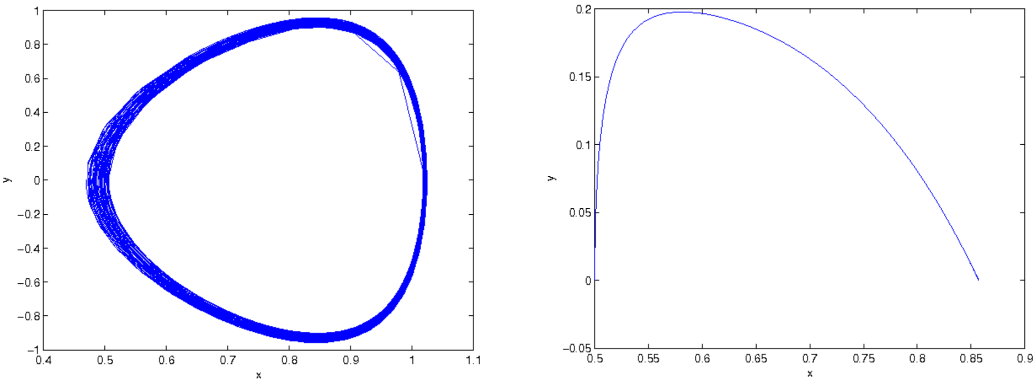


Figure 6.3: Unstable dynamics phase space ($g' < g$, $\left(\frac{m+M}{m}\frac{g}{(k+1)g-kg'}\right)^{1/2} < 1$). Left: only the stabilizing synergy is used. Right: both the dissipating and the stabilizing synergies are used.

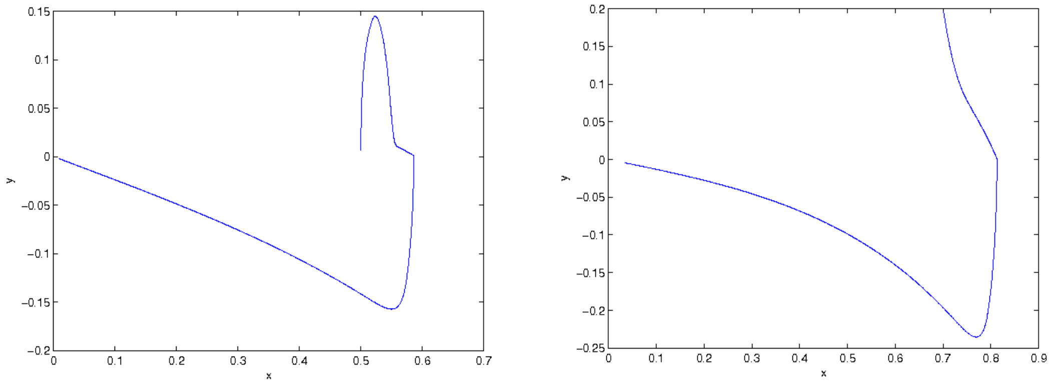


Figure 6.4: Phase space of the inverted pendulum controlled by the analogical synergy controller and initial conditions $x = 0.5$, $y = 0$ (left) and $x = 0.7$, $y = 0.2$ (right) in the unperturbed case. See figure 6.5 for the time evolution. Right: At the beginning of the evolution $\tilde{x} \in \mathcal{A}_1^C$ and the synergy activation is boosted consequently. After about 8 seconds from the onset of the evolution $\tilde{x} \in \mathcal{A}_2^C$ and the synergy activation is boosted again. At this point we have $g' \sim 9 * 1.59 = 14.31 > g$ and the origin is an asymptotically stable fixed point. Left: After about 4 seconds from the onset of the evolution $\tilde{x} \in \mathcal{A}_2^C$ and the synergy activation is boosted. At this point we have $g' \sim 9 * 1.47 = 13.23 < g$, which makes the origin the only asymptotically stable fixed point as $\left(\frac{m+M}{m}\frac{g}{(k+1)g-kg'}\right)^{1/2} > 1$.

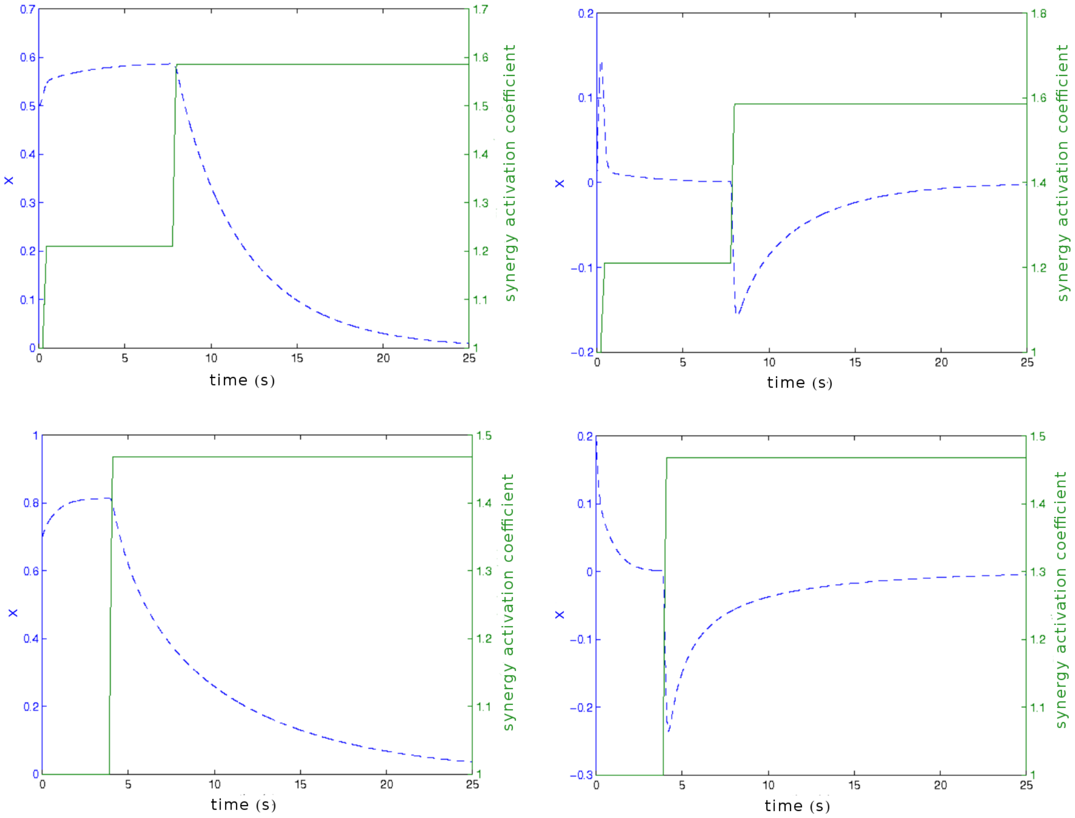


Figure 6.5: Top left: angle (dashed line) and synergy reflex coefficient (solid line) temporal evolution with initial conditions $x = 0.5$, $y = 0$, unperturbed case. Top right: angular speed (dashed line) and synergy reflex gain (solid line) temporal evolution with initial conditions $x = 0.5$, $y = 0$, unperturbed case. Bottom left: angle (dashed line) and synergy reflex gain (solid line) temporal evolution with initial conditions $x = 0.7$, $y = 0.2$, unperturbed case. Bottom right: angular speed (dashed line) and synergy reflex gain (solid line) temporal evolution with initial conditions $x = 0.7$, $y = 0.2$, unperturbed case.

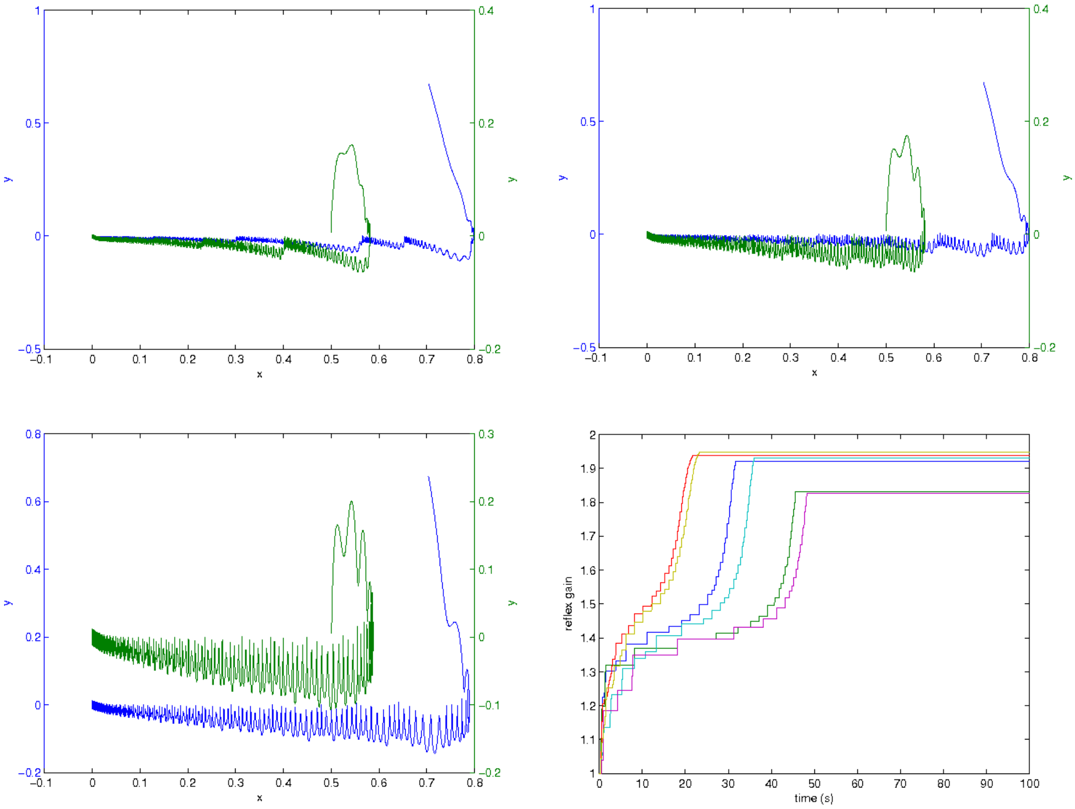


Figure 6.6: Top left: phase space for the inverted pendulum controlled by the analogical synergy feedback controller in the perturbed case and $D = 0.5$, $\alpha = 0.05$ and different initial conditions. Top right: phase space for the inverted pendulum controlled by the analogical synergy feedback controller in the perturbed case and $D = 1.0$, $\alpha = 0.1$ and different initial conditions. Bottom left: phase space for the inverted pendulum controlled by the analogical synergy feedback controller in the perturbed case and $D = 2.0$, $\alpha = 0.2$ and different initial conditions. Bottom right: synergy reflex gain temporal evolution for the inverted pendulum controlled by the analogical synergy feedback controller in the perturbed case and different initial conditions and perturbation intensities. The smallest boosts correspond to $D = 0.5$, $\alpha = 0.05$ perturbation intensities, the mid boosts to $D = 1.0$, $\alpha = 0.1$ perturbation intensities and the biggest boosts to $D = 2.0$, $\alpha = 0.2$ perturbation intensities. In each case there is over-boosting, compared to the unperturbed case, but with no saturation, as the maximum reflex gain is never reached.

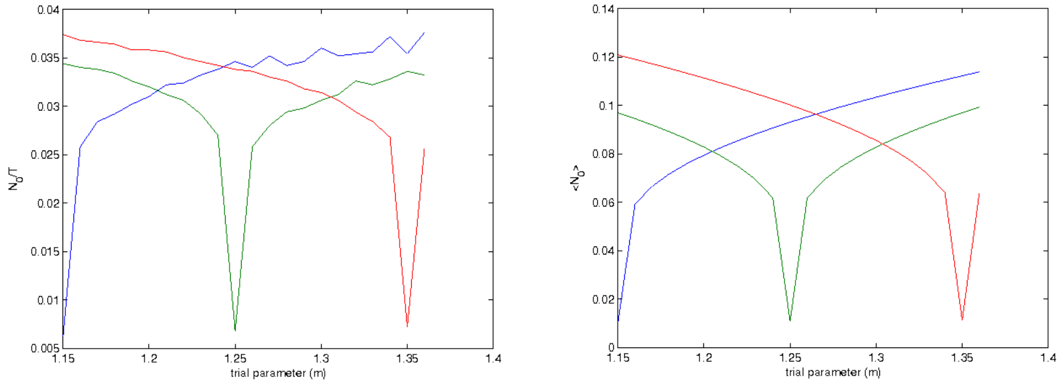


Figure 6.7: Left: simulation result; Right: results of computation based on Appendix A. In both cases, we perturb the real parameter by adding $1.0 \cdot 10^{-15}$ to it. That is why we do not get zero, but the minimum is still very evident.

Appendix A

Typical error computation for the quantized controller

When we consider the typical error made by the encoder in measuring the state of the system, it is important, in order not to overestimate it, to consider its exponential decay with rate $(\Lambda/N^{1/2})^{t/\tau}$. In the demonstration of Theorem 2, it was explained how it can happen that we get the zero symbol, due to a wrong estimate of the parameters in the encoder. The main point was that when the error is lower than $\frac{\log(\Lambda)}{(\Lambda-1)}\Delta\mu$, its rate of growth is no more given, between sampling intervals, by $\|e(t+t_0)\| \leq \|e(t)\|e^{Lt_0}$, but by $\|e(t+t_0)\| \leq \|e(t)\| + Lt_0\Delta\mu$. Thus, it can not be sufficient to enlarge the encoding region by a factor Λ at each sampling time to ensure a positive encoder symbol. On the other hand, as soon as we get positive symbols from the encoder, the error's norm at each sampling time, $k\tau$, is bounded by the quantization error

$$E_k = E_{d(\Omega_0)} \left(\frac{\Lambda}{N^{1/n}} \right)^{k-k_0},$$

where $k_0\tau$ is the last sampling time we get a zero from the encoder, Ω_0 is the possible set of unknown real parameters, μ, for which we know the trial parameter set, ν, suffices to contain the system, and $\Lambda = e^{L(\Omega_0)\tau}$

We thus can define an empirical probability of getting zero from the encoder at time $k\tau$, valid for $E_{k-1} < \log(\Lambda)\Delta\mu/(\Lambda-1)$, as

$$P_0(k\tau) = 1 - \frac{2\Lambda - 1}{\Lambda} \frac{\Lambda E_{k-1}}{E_{k-1} + \log(\Lambda)\Delta\mu}, \quad E_{k-1} = E_{d(\Omega_0)} \left(\frac{\Lambda}{N^{1/n}} \right)^{k-1-k_0}$$

We now want to compute the mean value of positive symbols, after a zero, before the next zero, and as a function of $\Delta\mu$.

There will be a "deterministic" number of positive symbols, needed to let the quantization error be smaller than $\frac{\log(\Lambda)}{(\Lambda-1)}\Delta\mu$, as after a zero it is reset to $E_{d(\Omega_0)} > \log(\Lambda)\Delta\mu/(\Lambda-1)$. This number is given by the smaller $\overline{k}(\Delta\mu)$ for which

$$N^{1/n}d(\Omega_0)\left(\frac{\Lambda}{N^{1/n}}\right)^{\overline{k}} < \frac{\log(\Lambda)}{(\Lambda-1)}\Delta\mu$$

Then, the probability of getting h additional positive symbols is given by

$$\tilde{P}(h) = (1 - P_0((k_0 + \overline{k} + 1)\tau)(1 - P_0((k_0 + \overline{k} + 2)\tau) \dots (1 - P_0((k_0 + \overline{k} + h)\tau)$$

and its mean value, $< h >=< h(\Delta\mu) > (\Delta\mu)$, by

$$< h >= \sum_i i\tilde{P}(i).$$

Finally we can compute the upper bound of the typical error done by the encoder $\tilde{e}(\Delta\mu)$ as the mean value of $N^{1/n}d(\Omega_0)\left(\frac{\Lambda}{N^{1/n}}\right)^l$ for $l \in \aleph$, $l \in [0, \overline{k} + h')$. Note that both $\overline{k}(\Delta\mu)+ < h(\Delta\mu) > (\Delta\mu) = \overline{h}(\Delta\mu)$ and $\frac{1}{\tilde{e}(\Delta\mu)}$ are monotonically increasing with $\Delta\mu$.

Bibliography

[1] Alberto Bressan and Benedetto Piccoli, *Introduction to the Mathematical Theory of Control*, AIMS Series on Applied Mathematics

[2] J.P.Hespanha, D.Liberzon *Stabilization of nonlinear systems with limited information feedback*, IEEE TR. ON AUT. CONTROL, VOL. 50, NO. 6, JUNE 2005

[3] C. De Persis *n-Bit Stabilization of n-Dimensional Nonlinear System in Feedforward Form*, IEEE TR. ON AUT. CONTROL, VOL. 50, NO. 3, MARCH 2005

[4] R.S.Johansson *Sensory Input and The Control of the Grip*, Symposium on sensory guidance of movement Novartis Foundation, 1998

[5] R.Shedmer, F.A. Mussa-Ivaldi *Computational Elements of the Adaptive Controller of the Human Arm*, Advances in Neural Information Processing, Vol. 6, 1994, pp. 1077-1084

[6] H.Gomi, M.Kawato *Learning Control for a Closed Loop System using Feedback-Error-Learnign*, Proc. 29th Conference on Decision and Control, 1990, pp. 3289-3294

[7] J.Lasalle,S.Lefschetz *Stability by Lyapunov, Direct Method and Application* Accademic Press, New York, 1961

[8] X.Liao *Stability Theory and its Application* Huazhong Normal University Press, 1988

[9] X.Yang *Practical stability in dynamical systems* Chaos, Solitons and Fractals, vol. 11, 2000, pp. 1087-1092

[10] T.Kapitaniak,J.Brindley *Practical stability of chaotic attractors* Chaos, Solitons and Fractals, vol. 9, 1998, pp. 43-50

[11] C.De Persis, A.Isidori *Stabilization by state feedback implies stabilizability by encoded state feedback*, Systems & Control Letters, Vol. 53, Issues 3-4, November 2004, pp. 249-258

[12] A.M.Bloch, N.E.Leonard *Controlled Lagrangians and the Stabilization of Mechanical Systems I: The First Matching Theorem*, IEEE Tr. Aut. Cont., VOL. 45, NO. 12, December 2000, pp. 2253-2269

[13] Slotine J-JE., Li W. *Applied Nonlinear Control*, Prentice Hall, 1991

[14] strm KJ., Wittenman B. *Adaptive Control*, Addison Wesley, 1995

[15] Ioannou PA., Sun J. *Robust Adaptive Control*, Upper Saddle River, NJ, Prentice Hall PTR, 1996

[16] Mareels I. *A simple selftuning controller for stably invertible systems*, Systems & Control Letters, 1984

[17] Polderman JW, Mareels I. *Two Scale High Gain Adaptive Control*, International Journal of Adaptive Control and Signal Processing, 18:393402, 2004

[18] Ogata, K. *Modern Control Engineering*, Prentice-Hall Inc., 1997

[19] Slotine, J-J.; Li, W. *Adaptive manipulator control: a case study* IEEE International Conference on Robotics and Automation, 1987

[20] Kawato M, Furukawa K, et al. *A hierarchical neural-network model for control and learning of voluntary movement*, Biological Cybernetics, 57:169185, 1987

[21] Sherrington C. 1910 *Flexion-reflex of the limb, crossed extension reflex and reflex tepping and standing*, J. Physiol., vol 40,1910, pp. 28-121

[22] Merton P. *How we contro teh contraction of muscles* Sci. Am., vol. 226, 1972, pp. 30-37

[23] Taub E. & Berman *Movement learning in the absence of sensory feedback*, Neurophysiology of spatial oriented behavior (ed S. Freeman), 1968, pp. 173-192

[24] Vallbo A. *Slowly adapting receptors in man*, Acta Physiol. Scand, vol. 78, 1970, pp. 315-333

[25] Brady M., Hollerbach J., Johnson T., Lonzano-Perez T. & Mason M. *Robot motion: planning and control*, Cambridge MA: MIT Press, 1982

[26] Hollerbach J., Flash T. *Dynamic interactions between limb segments during planar arm movements*, Biol. Cybernet., vol. 44, 1982, pp. 67-77

[27] Sainhurg R., Poizner H. *Loss of proprioceptions produces deficits in interjoint cordination*, J. Neurophysiol, vol. 70, 1993, pp. 2136-2147

[28] Bastian A.J., Martin T.A., Keating J.G. & Thach W.T. *Cerebellar ataxia: abnormal control of interaction torques across multiple joints*, J. Neurophysiol., vol. 76, 1996, pp. 492.509

[29] Raibert M. *A model for sensorimotr control and learning*, Biol. Cybernet., vol. 29, 1978, pp. 29-36

[30] Albus J. *Theory of the cerebellar function*, Math. Biosci., vol. 10, 1971, pp. 25-61

[31] Marr D. *A theory of cerebellar cortex*, J. Physiol., vol. 202, 1969, pp. 437-470

[32] Raibert M., Horn B. *Manipulator control using the configuration space method*, Industr. Robot., vol. 5, 1978, pp. 69-73

[33] Hogan N. *Impedance control: an approach to manipulation*, ASME J Dynamic Syst. Measurment Control, vol. 107, 1985, pp. 1-24

[34] Feldmann A.G. *Functional tuning of the nervous system with control of movement or maintenace of steady posture. II. Controllable parameters of muscles*, Byophysics, vol. 11, 1966, pp. 565-578

[35] Polit A., Bizzi E. *Characteristics of motor programs underlying arm movements in monkeys*, J. Neurophysiol., vol. 42, 1979, pp. 183-194

[36] Bizzi E., Accornero N, Chapple W., Hogan N., *Posture control and trajectory formation during arm movement*, J. Neuroscience, vol. 4, 1984, pp. 2738-2744

[37] Hogan N. *The mechanics of posture and movement*, Biol. Cybernet, vol. 52, 1985, pp. 315-331

[38] Flash T. *The control of hand equilibrium trajectories in multi joint arm movements*, Biol. Cybernet., vol. 57, 1987, pp. 257-274

[39] Morasso *Spatial control of arm movements*, Exp. Brain Res, vol. 42, 1981, pp. 223-227

[40] Mussas-Ivaldi F., Hogan N., Bizzi E., *Neural, mechanical and geometrical factors subserving arm posture in humans*, J. Neurosci., vol. 5, 1985, pp. 2732-2743

[41] Gomi H., Kawato M.*Human arm stiffness and equilibrium point trajectory during multi joint movements*, Biol. Cybern., vol. 76, 1997, pp. 163-171

[42] Won J., Hogan N. *Stability properties of human reaching movements*, Exp. Brain Res., vol. 107, 1995, pp. 125-136

[43] Shadmehr R., Mussa-Ivaldi F., Bizzi E. *Postural force fields of the human arm and their role in generating multi-joint movements*, J. Neurosci., vol. 13, 1993, pp. 45-62

[44] Lackner J.R., Dizio P. *Rapid adaptation to Coriolis force perturbation of arm trajectory*, J. Neurophysiol., vol. 72, 1994, pp. 299-313

[45] Shadmehr R., Mussa-Ivaldi F. *Adaptive representation of dynamics during learning of a motor task*, J. Neurosci, vol. 13, 1994, pp. 45-62

[46] Bizzi E., Mussa-Ivaldi F., Giszter S. *Computaions underlying the execution of movement: a biological perspective*, Science, vol. 253, 1991, pp. 287-291

[47] Giszter S., Mussa-Ivaldi F., Bizzi E. *Convergent force fields organized in the frog's spinal cord*, J. Neurosci, vol. 13, 1993, pp. 467-491

[48] Tresch M., Bizzi E. *Respones to spinal miscrustimulation in the chronically spinalized rat and their relationship to spinal systems activated by low threshold cutaneous stimulation*, Exp. Brain Res., vol. 129, 1999, pp. 401-416

[49] Mussa-Ivaldi F., Giszter S., Bizzi E. *Motor-space coding in the central nervous system*, Cold Spring Harb. Symp. Quant. Biol., vol. 55, 1990, pp. 827-835

[50] Mussa-Ivaldi F., Giszter S., Bizzi E., *Linear combinations of primitives in vertebrate motor control*, Proc. Natl Acad. Sci. USA, vol. 91, 1994, pp. 7534-7538

[51] Kargo W.J., Giszter S.F. *Rapid correction of aimed movements by summation fo force fields primitives*, J. Neurosci., vol. 20, 2000, pp. 409-426

[52] Mussa-Ivaldi F., Giszter S. *Vector field approximation: a computational paradigm for motor control and learning*, Biol. Cybernet., vol. 67, 1992, pp. 491-500

[53] Mussa-Ivaldi F., *Nonlinear force fields: a distributed system of control primitives for representing and learning movements*, Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and automation pp 84-90 Los Alamitos, CA: IEEE Computer Society Press, 1997

[54] Jordan M., Rumelhart D. *Forward models supervised learning with a distal teacher*, Cogn. Sci 16, 1992, pp. 307-354

[55] Kawato M., Wolpert D. *Internal models for motor control*, Novartis Found. Symp, vol. 218, 1998, pp. 291-304

[56] McIntyre J., Berthoz A., Lacquaniti F. *Reference frame and internal models*, Brain Res. Rev., vol. 28, 1988, pp. 143-154

[57] Wolpert D., Miall R., Kawato M. *Internal model in the cerebellum*, Trends Cogn. Sci, vol. 2, 1998, pp. 338-347

[58] Mussa-Ivaldi F.A., Bizzi E. *Motor learning through the combination of primitives*, Phil. Trans. R. Soc. Lond., vol. 355, 2000, pp. 1755-1769

[59] Padoa-Schioppa C., Li C.R., Bizzi E., *Neural activity in the supplementary motor area of monkeys adapting to a new dynamical environment*, J. Neurophysiol, vol. 91, 2004, pp. 449-473

[60] Cheung VCK, d'Avella A, Tresch MC, Bizzi E *Central and sensory contributions to the activation and organization of muscle synergies during natural motor behaviors*, J.Neurosci, vol. 25(27), 2005, pp. 6419-6434

[61] Lambert J.D.C, *Lecture notes in neurophysiology: Control of motor function*, http://www.fi.au.dk/jl/lect.html

[62] R. Osu, E. Burdet, D. W. Franklin, T. E. Milner, M. Kawato *Different Mechanisms Involved in Adaptation to Stable and Unstable Dynamics*, J. Neurophysiol, vol. 90, 2003, pp. 3255-3269

[63] R. FitzHugh, Biophys. J 1, 445, 1961

[64] J. Nagumo, S. Arimoto, S. Yoshizawa, Proc. IRE 50, 2061, 1962

[65] W.J Freeman *How and why brains create meaning from sensory information*, Int. J. Bifurc. Chaos 14, vol. 2, 2004

[66] W.J. Freeman *Characteristics of the synchronization of brain activity imposed by finite conduction velocities of axons*, Int. J. Bifurc. Chaos, vol. 10, 1999

[67] R. Kozma, W.J. Freeman *Chaotic resonance - Method and applications for robust classifcation of noisy and variable patterns*, International Journal of Bifurcation and Chaos, 2001

[68] T. Yanagita, T. Ichinomiya, Y. Oyama *Pair of excitable FitzHugh-Nagumo elements: Synchronization, multistability and chaos*, Phys. Rev E 72, 2005

[69] E.D Sontag *Smooth stabilization implies coprime factorization*, IEEE Trans. Automatic Control, vol. AC-34, 1989, pp. 435-443

[70] E.D Sontag, Y. Wang *New Characterizations of Input to State Stability*, IEEE Transactions on Automatic Control, 1996

[71] L. A. Bunimovich, S. G. Dani, R. L. Dobrushin, M. V. Jakobson, I. P. Kornfeld, N. B. Maslova, Ya. B. Pesin, Ya. G. Sinai, J. Smillie, Yu. M. Sukhov, A. M. Vershik *Dynamical Systems, Ergodic Theory and Applications*, Springer, 2000

[72] P. Cvitanovic, R. Artuso, R. Mainieri, G. Tanner, G. Vattay, N. Whelan, A. Wirzba *Classical and quantum chaos Part I: Deterministic Chaos*, 2004, ChaosBook.org

[73] B. R. Andrievskii, A. L. Fradkov *Control of Chaos: Methods and Applications I: Methods*, Automation Remote Control, vol. 64, 2003, pp. 673-713

[74] X. Yang, S. Zhang *On the possibility of vreating new asymptotically stable periodic orbits in continuos time dynamical systems by small feedback control*, Nonlinearity, vol. 16, 2003, pp. 1853-1859

[75] S. BOCCALETTI, C. GREBOGI, Y.-C. LAI, H. MANCINI, D. MAZA *THE CONTROL OF CHAOS: THEORY AND APPLICATIONS*, Physics Reports, vol. 329, 2000, pp. 103-197

[76] Clay J.R, J. Comput. Neurosci. 15, vol. 43, 2003

[77] Arecchi F.T., Boccaletti S. *Adaptive strategies for recognition, noise filtering, control, synchronization and targeting of chaos*, Chaos, vol. 7, 1997, pp. 621-

[78] V.P. Zhigulin, M.I Rabinovich, R. Huerta, H.D.I Abarbanel *Robustness and Enhancement of Neural Synchronization by Activity Dependent Coupling*, arXiv:physics/0212070v1, 2002

[79] A.J. Jones et al *Neural model of arbitrary chaotic systems: construction and the role of time delayed feedback in control and synchronization*, Complexity Int, 2002

[80] N.J. Corron et al. *Limiter control of a Chaotic Transistor Oscillator*, Int. J. Bifurcation Chaos, vol. 13, 2003

[81] D. Harte *Multifractals theory and applications*, Chapman & Hall CRC, 2001

[82] H.D.I Abarbanel, P.S. Linsy *Secure Communications and Unstable Periodic Orbits of Strange Attractors* IEEE Transaction on circuits and systems-II: Analog and Digital Signal Processing, vol. 40, 1993

[83] P. Thiran, M. Hasler *Information Processing using stable and unstable oscillation: a tutorial* Third IEEE International Workshop on Cellular Neural Networks and their Applications, Rome, Italy, December 18-21, 1994

[84] R. N. Stiles *Frequency and displacement amplitude relations for normal hand tremor*, J Appl Physiol, vol. 40, 44-54, 1976

[85] I. Leyva et al. *In phase and anti-phase synchronization of coupled homoclinic chaotic oscillators* arXiv:nlin/0210042v1

[86] `http://www-chaos.umd.edu/publications/papers.html#fractalBasin),`