Neural Networks in press
URL ftp://iserv.iki.kfki.hu/pub/papers/new/szepes.cc.ps.Z
WWW http://iserv.iki.kfki.hu/New/index.html

Neurocontroller using Dynamic State Feedback for Compensatory Control

Csaba Szepesvári [b,a] Szabolcs Cimmer [b,a] András Lőrincz [c,a,1]

[a] *Department of Photophysics, Institute of Isotopes of the Hungarian Academy of Sciences, Budapest, P.O. Box 77, Hungary, H-1525*

[b] *Bolyai Institute of Mathematics, University of Szeged, Szeged, Hungary, H-6720*

[c] *Department of Adaptive Systems, University of Szeged, Szeged, Hungary, H-6720*

Abstract

A common technique in neurocontrol is that of controlling a plant by static state feedback using the plant's inverse dynamics, which is approximated through a learning process. It is well known that in this control mode even small approximation errors or, which is the same, small perturbations of the plant may lead to instability. Here, a novel approach is proposed to overcome the problem of instability by using the inverse dynamics both for the Static and for the error compensating Dynamic State feedback control. This scheme is termed SDS Feedback Control. It is shown that as long as the error of the inverse dynamics model is "signproper" the SDS Feedback Control is stable, i.e., the error of tracking may be kept small. The proof is based on a modification of Liapunov's second method. The problem of on-line learning of the inverse dynamics when using the controller simultaneously for both forward control and for dynamic feedback is dealt with, as are questions related to noise sensitivity and robust control of robotic manipulators. Simulations of a simplified sensorimotor loop serve to illustrate the approach.

Keywords: Neural network control, compensating perturbations, stability, feedback control, feedforward control, inverse dynamics, on-line learning, Liapunov's second

method.

1 Introduction

A vast amount of work has dealt with neural networks for controlling a plant with known, partially known, or unknown dynamics. Techniques, such as inverse system identification (see e.g. (Miller, 1987; Kawato et al., 1987; Widrow et al., 1978)), or identification-based ("indirect") methods (see e.g. (Jordan, 1990; Werbos, 1988; Widrow, 1986)), have been proposed for learning the inverse dynamics. (For an overview see (Dean and Wellman, 1991; Miller et al., 1990; Narendra and Parthasarathy, 1990) or (Vemuri, 1993)). Some of the proposed techniques separate the learning and a subsequent working phase. During the working phase the controller is no longer adapting. In real world problems it is quite common, however, that the plant's dynamics changes over time, i.e., it might be necessary to retain adaptivity. Adopting now a different approach, the retention of adaptivity during the working phase cannot solve all problems. To give an example, if the dynamics has to be relearned whenever the load changes it may involve a considerable time until the controller can get accustomed to working with the required precision in a new task. There are at least two options when dealing with this problem: (i) add new dimensions to the dynamics and learn the control policy for every possible task (e.g., weights) and estimate the load on-line (Anderson and Miller, III, 1992), or (ii) use a feedback controller in order to extend the region in which the feedforward controller can work (Miyamoto et al., 1988; Lewis et al., 1995). This latter option has several advantages but may also be disadvantageous. In order to elaborate this point the terminology will be defined first, since the

[1] This work was partially founded by OTKA Grants T017110, T014330, T014566, and US-Hungarian Joint Fund Grants 168/91-A, 519/95-A.

meaning of some of the concepts may be different depending on the field, viz. Control, Artificial Intelligence, Neural Networks, etc.

If planning and control are interleaved, i.e., at each time t the upgraded *instantaneous* (state) information [2] is used to generate a new control signal then the system will be called a *closed-loop system*. If the value of the control at time t depends only on the state of the plant at the same time, the control is said to be in a *static state feedback* control mode and the controller is called a *feedforward controller* (FFC). Assume that the planned motion and the actual motion are different. Then the difference, i.e., the error, can be used to generate an error-compensating signal. Generation of the error-compensating signal is the task of the *feedback controller* (FBC). (Note the ambiguous use of the term feedback.) The output of the feedback controller should be integrated in order to recall previous errors and thus to develop a *preventive compensatory control signal*. This means that a feedback controller applies *dynamic state feedback*, i.e., it is precisely the *dynamics* of the (compensatory) control signal that depends on the state of the plant as opposed to the case of static state feedback when the control signal itself depends on the state of the plant. In other words, in the case of dynamic state feedback the control signal is the output of another dynamical system. If, however, one views the problem from the aspect of the feedback controller, its output may depend only on the error, i.e., the feedback controller may itself be a feedforward control system working on the error as the state input. From this viewpoint the task of the feedback and that of the feedforward controller are similar: both should map state values to *control values*. In the following we use the term *feedback control* to refer to dynamic state feedback control.

Now, we again consider feedback control: the advantage of the extra FBC is

[2] It is assumed that the state information contains all the information needed to describe the dynamics of the plant. In case of sensorimotor control the state information should be developed from the sensory input. In this case observability, i.e., whether enough information can be recovered or not, is also a question.

that it allows the FFC to work with a broader range of problems since the FBC can compensate for errors. However, since feedback is working on the basis of a possible error, such an error first has to develop before any compensatory action can be made, i.e., feedback control is somewhat delayed.

Neurocontrollers typically realise static state feedback control where the neural network is used to approximate the inverse dynamics of the controlled plant (Miller et al., 1990). In practice it is often unknown *a priori* how precise such an approximation can be. On the other hand, it is well known that in this control mode even small approximation errors can lead to instability (Ortega and Yu, 1987). The same instability happens if one is given a precise model of the inverse dynamics, but the plant's dynamics changes. There are well-known ways of neutralising the effects of unmodelled dynamics in adaptive control, such as the σ-modification, signal normalisation, (relative) dead zone, and projection methods, being widely used and discussed in the control literature (see for example, (Ortega and Yu, 1987)). Here we propose a new method where the FFC is used in a parallel feedback operation mode, i.e., the same controller provides both the feedforward and the compensatory feedback signals. The rate of change of the compensatory control signal is the difference of the optimal control signal of the unperturbed plant and the control signal that would move the unperturbed plant in the direction along which the plant has moved. Here we shall prove that the resulting control scheme is robust: the error of tracking may be kept as small as desired by adjusting the gain of the feedback signal, provided that the error of the inverse dynamics model is "signproper". This method may be considered as an alternative to stabilzing control-loops by means of PD/PID controllers.

The SDS control scheme might be an attractive answer to the dilemma "when to switch between feedforward and feedback control methods". The dilemma arises when the learning issue is considered, since errors should result in learning. However, if both feedback and feedforward methods are taking place, then an interesting instability may arise, namely: Which system is to be blamed for the error? In other words: Which system should be trained? This is one type of the credit assignment problem (Minsky, 1961). This problem seems easier if the feedforward and the feedback systems are the same. We shall return to this point later. It will be shown by theoretical considerations as well as by computer experiments that the compound controller is capable of compensating perturbations, i.e., perturbations that do not reverse the effect of any component of the control signal.

The article is organized as follows: First, in Section 2, we give the background required for considering the perturbed dynamics: Some assumptions considering the plant's dynamics are given; we then examine the so called speed field tracking task that forms the bases of our analysis; finally, we consider the perturbed dynamics and define the static state compensatory signal. The next section (Section 3) deals with error-compensating dynamic state feedback control. First, the proposed dynamics of control is given. Then we analyze this control scheme and prove that under reasonable conditions the error signal is uniformly ultimately bounded inside a neighborhood of the origin. The size of this neighborhood depends on the gain of the dynamic state feedback control. Larger gain leads to a larger upper bound on the initial error that can be compensated and a smaller ultimate bound on the error. In Section 4 we consider the effects of using the same controller for dynamic and static state feedback *during learning*. It appears that direct learning methods fit the proposed scheme, whereas indirect learning methods are more suitable if used together with a fixed, independent stabilizing feedback controller. Illustrative simulations of a sensorimotor loop are given in Section 5. Both the sensorimotor loop and the parallel architecture of the neurocontroller, which can be tuned by strictly associative learning techniques, are described. The simulations illustrate how the scheme can compensate for structural errors as well as for external (unmodelled) perturbations. Finally, in Section 6 we discuss some questions related to nonstationary perturbations, noise sensitivity, and the controlling of higher order plants. Our conclusions are drawn in Section 7. The Appendix contains the proof of the required modification of Liapunov's second method, a detailed proof of the ultimate boundedness of the error in the error-compensating control scheme as well as the proof that the results are relevant to the controlling of robotic manipulators that should work with variable loads.

2 Background

2.1 Assumptions concerning the controlled plant

Let $\mathbf{R}^{m\times n}$ denote real $m \times n$ matrices. We say that a matrix $\mathbf{A}$ admits a generalized inverse[3] if there is a matrix $\mathbf{X}$ for which $\mathbf{AXA} = \mathbf{A}$ holds.[4] For convenience, the generalized inverse of a non-singular matrix $\mathbf{A}$ will be denoted by $\mathbf{A}^{-1}$.

Assume that the plant's equation is given in the following form (Isidori, 1989):

$$\dot{\mathbf{q}} = \mathbf{b}(\mathbf{q}) + \mathbf{A}(\mathbf{q})\,\mathbf{u} \tag{1}$$

where $\mathbf{q} \in \mathbf{R}^n$ is the state vector of the plant, $\dot{\mathbf{q}}$ is the time derivative of $\mathbf{q}$, $\mathbf{u} \in \mathbf{R}^m$ is the control signal, $\mathbf{b}(\mathbf{q}) \in \mathbf{R}^n$, and $\mathbf{A}(\mathbf{q}) \in \mathbf{R}^{n\times m}$. We assume that the domain (denoted by D) of the state variable $\mathbf{q}$ is compact and is simply connected; that $n \leq m$, and for each $\mathbf{q} \in D$ the rank of matrix $\mathbf{A}(\mathbf{q})$ is equal

[3] Sometimes it is called the pseudo-inverse, or simply the inverse of matrix $\mathbf{A}$.

[4] It is well known that (i) $\mathbf{A}$ is nonsingular if and only if it has a unique generalized inverse and (ii) all the solutions of the linear equation $\mathbf{Ax} = \mathbf{b}$ have the form $\mathbf{x} = \mathbf{Xb} + (\mathbf{E} - \mathbf{XA})\mathbf{y}$ provided that the considered linear equation does in fact have a solution (Ben-Israel and Greville, 1974). Here $\mathbf{y}$ denoted an arbitrary vector of the appropriate dimensions.

to n; that is, the matrix is nonsingular. As a consequence the plant is strongly controllable. In this case the inequality $n < m$ means that there are more independent actuators than state vector components, i.e., the control problem is redundant. Another kind of redundancy, or ill-posedness occurs when $n > m$ in which case even $\mathbf{A}^{-1}$ is non-unique.

Further, we assume that both of the matrix fields, $\mathbf{A}(\mathbf{q})$ and $\mathbf{A}^{-1}(\mathbf{q})$ are differentiable w.r.t. $\mathbf{q}$ (differentiation is assumed to be extended to matrix fields in the usual way (Lovelock and Rund, 1975)).

2.2 *Speed field tracking*

One way to obtain a closed-loop control task is to consider the speed field tracking problem. This is defined as follows: Let $\mathbf{v} = \mathbf{v}(\mathbf{q})$ be a fixed n dimensional vector field over D. The *speed field tracking task* is to find the static state feedback control $\mathbf{u} = \mathbf{u}(\mathbf{q})$ that solves the equation $\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q})$, that is

$$\mathbf{v}(\mathbf{q}) = \mathbf{b}(\mathbf{q}) + \mathbf{A}(\mathbf{q})\mathbf{u}(\mathbf{q}). \tag{2}$$

Speed field tracking is non-typical in the control literature, but arises naturally if we consider path planning tasks (Connolly and Grupen, 1993; Fomin et al., 1994; Lei, 1990). More conventional tasks, such as the *point to point control* and the *trajectory tracking* tasks cannot be exactly rewritten in the form of speed field tracking.

In the case of point to point control the task is to find a control that moves the plant from a given initial state $(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ into a prespecified final state given by $\mathbf{q}_f$ and $\dot{\mathbf{q}}_f = 0$, the control signal being a function of time. Point to point control is ill-posed as there are an infinite number of paths to $(\mathbf{q}_f, \dot{\mathbf{q}}_f = 0)$. The requirement of "collision free" motion when the plant should not enter a so called (stationary) obstacle region restricts the variety of solutions but can not solve the problem of ill-posedness in the general case. If one can design a collision free path as a function of time, $\mathbf{q}_d(t)$, then collision free motion may be ensured if this path is tracked as closely as possible. In this way one arrives at the trajectory tracking task when a trajectory is given and the aim of the control is to find a feedback control law which is able to impose on the error $\mathbf{q}(t)-\mathbf{q}_d(t)$ a behavior which asymptotically decays to zero as time tends to infinity. For convenience, it is usually assumed that the desired reference trajectory is not just a fixed function of time but, rather, coincides with the output of some autonomous dynamical system.

Speed field tracking may be formulated as a special case of trajectory tracking provided that the distinguished autonomous system is the plant itself controlled by an optimally designed state feedback control law. It then follows that the major difference between speed field tracking and trajectory tracking is caused by the fact that a speed field is given as a function of state while a trajectory is given as a function of time. Consequently speed field tracking is more robust against state perturbations. This can be important if it is critical to ensure collision free motion. Note that trajectory tracking may result in collision if the actual and the desired states of the plant differ sufficiently. Often it is hard to exclude the possibility of such differences because of unforeseen disturbances. With speed field tracking there is no such problem since the speed field determines the motion as a function of state rather than as a function of time. Speed field design for collision free control is the subject of current research (Hwang and Ahuja, 1992). It is important to note, that collision free speed fields may be constructed efficiently by computing the stationary flow of a well designed diffusion over the state space (Lei, 1990; Tarassenko and Blake, 1991; Keymeulen and Decuyper, 1992; Connolly and Grupen, 1993; Morasso et al., 1993; Glasius et al., 1995). The neural architecture that is capable of designing a discretized speed field is briefly described in Section 5.1. Another method for constructing the speed field is the potential field method (Locano-Pèrez and Wesley, 1979). Artificial potential fields, however, may have deceptive local minima.

2.3 Inverse dynamics

Given the plant's dynamics by Equation (1) the inverse dynamics of the plant is given as follows:

$$\mathbf{p}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{A}^{-1}(\mathbf{q})\big(\dot{\mathbf{q}} - \mathbf{b}(\mathbf{q})\big) + \big(\mathbf{E} - \mathbf{A}^{-1}(\mathbf{q})\mathbf{A}(\mathbf{q})\big)\mathbf{y}(\mathbf{q}, t), \tag{3}$$

where $\mathbf{y} = \mathbf{y}(\mathbf{q}, t)$ is an arbitrary function. Of course, the control signal

$$\mathbf{u}(\mathbf{q}) = \mathbf{p}(\mathbf{q}, \mathbf{v}(\mathbf{q})) \tag{4}$$

solves the speed field tracking control task given by Equation (2). *In the following we will look at the main value of the inverse dynamics, i.e., we assume that* $\mathbf{y}(\mathbf{q}, t) = 0$ *and thus*

$$\mathbf{p}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{A}^{-1}(\mathbf{q})\big(\dot{\mathbf{q}} - \mathbf{b}(\mathbf{q})\big). \tag{5}$$

This assumption simplifies the calculations and is justified by the learning method described in detail in (Fomin et al., 1994; Szepesvári and Lőrincz, 1995; Szepesvári and Lőrincz, 1996c).

3 Compensatory control by dynamic state feedback

The topic of the present article is to propose dynamic state feedback control to compensate *permanent* perturbations or, which is equivalent, unmodelled dynamics. The problem is the following. Assume that at state $\mathbf{q}$ the speed field tracking problem prescribes the speed $\mathbf{v}(\mathbf{q})$ for the plant. Assume, further, that the equation of motion of the plant changes and the new equation system reads as follows:

$$\dot{\mathbf{q}} = \hat{\mathbf{b}}(\mathbf{q}) + \hat{\mathbf{A}}(\mathbf{q})\mathbf{u}, \tag{6}$$

where $\hat{\mathbf{A}}(\mathbf{q})$ is a nonsingular matrix field. Let us first assume that we seek a static state feedback compensatory control signal, $\mathbf{w} = \mathbf{w}(\mathbf{q})$, such that the control signal $\mathbf{u}(\mathbf{q}) + \mathbf{w}(\mathbf{q})$ solves the original speed field tracking problem for the perturbed plant. One can check that for the compensatory control signal [5]

$$\mathbf{w}(\mathbf{q}) = \hat{\mathbf{A}}^{-1}(\mathbf{q})\big(\mathbf{v}(\mathbf{q}) - \hat{\mathbf{v}}(\mathbf{q})\big) \tag{7}$$

it holds that $\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q})$. Here $\hat{\mathbf{v}}(\mathbf{q})$ is the speed vector field followed by the perturbed plant provided that the control signal is $\mathbf{u}(\mathbf{q})$:

$$\hat{\mathbf{v}}(\mathbf{q}) = \hat{\mathbf{b}}(\mathbf{q}) + \hat{\mathbf{A}}(\mathbf{q})\mathbf{u}(\mathbf{q}).$$

Unfortunately, the learning of $\mathbf{w}(\mathbf{q})$ is as complex as it is to estimate $\hat{\mathbf{A}}^{-1}(\mathbf{q})$ and $\hat{\mathbf{b}}(\mathbf{q})$ and thus it is the same as retaining the adaptivity of the feedforward controller. We should like to alleviate this problem by introducing dynamic state feedback for estimating the compensatory control signal.

First, observe that $\mathbf{w}(\mathbf{q})$ satisfies the equality

$$\mathbf{u}(\mathbf{q}) = \mathbf{p}\left(\mathbf{q}, \dot{\mathbf{q}}\right), \tag{8}$$

where $\dot{\mathbf{q}}$ is the speed of the plant controlled by $\mathbf{u} = \mathbf{u}(\mathbf{q}) + \mathbf{w}(\mathbf{q})$:

$$\dot{\mathbf{q}} = \hat{\mathbf{b}}(\mathbf{q}) + \hat{\mathbf{A}}(\mathbf{q})\big(\mathbf{u}(\mathbf{q}) + \mathbf{w}(\mathbf{q})\big). \tag{9}$$

[5] Note that the compensatory control signal $\mathbf{w}(\mathbf{q}) + (\mathbf{E} - \mathbf{A}^{-1}(\mathbf{q})\mathbf{A}(\mathbf{q}))\mathbf{y}(\mathbf{q}, t)$, where $\mathbf{y} = \mathbf{y}(\mathbf{q}, t)$ is arbitrary, results in $\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q})$, too. Thus $\mathbf{w}(\mathbf{q})$ can be viewed as the mean or main part of perfect compensatory signals.

The simplest error-feedback law is to let $\mathbf{w}$ change until Equation (8) is satisfied. Using Equation (4) we get the following equations:

$$\begin{aligned} \dot{\mathbf{w}} &= \Lambda\Big(\mathbf{p}(\mathbf{q}, \mathbf{v}(\mathbf{q})) - \mathbf{p}(\mathbf{q}, \dot{\mathbf{q}})\Big) \\ \dot{\mathbf{q}} &= \hat{\mathbf{b}}(\mathbf{q}) + \hat{\mathbf{A}}(\mathbf{q})\Big(\mathbf{u}(\mathbf{q}) + \mathbf{w}\Big) \end{aligned} \tag{10}$$

where Λ is a fixed positive number.[6] Fortunately, Equation (10) can be realized by applying a compound control algorithm provided that the speed of the plant is measurable. The block diagram of the compound controller is given in Fig. 1.

As is depicted in the figure and is suggested by the equations the controller that realizes the inverse dynamics plays a dual role: it computes the feedforward control signal that would move the unperturbed plant into the desired direction and, in case of error, the very same controller also computes the (feedback) compensatory signal. The compound controller will be called **S**tatic and **D**ynamic **S**tate Feedback Controller (SDS Feedback Controller). The computation of the control signal is as follows: We assume that the state and the speed of the plant are available. The inverse dynamics controller first computes the feedforward control signal by using the speed field to be tracked at point $\mathbf{q}$. Then an identical copy of the same controller computes a control signal by using the actual speed ($\dot{\mathbf{q}}$) of the plant. This control signal is then subtracted from the feedforward control signal, the result is integrated through time, and is added to the feedforward control signal. The sum is used as the control input to the plant.

[6] Equivalently one might consider the feedback equation

$$\dot{\mathbf{w}} = \Lambda\Big(\mathbf{p}(\mathbf{q}, \mathbf{v}(\mathbf{q})) + \mathbf{p}(\mathbf{q}, -\dot{\mathbf{q}})\Big).$$

Again, in the case of equilibrium $\mathbf{w} = \mathbf{w}(\mathbf{q})$. These two equations are not the same if $\mathbf{b}(\mathbf{q})$ is nonzero.

It is clear that in feedback equilibrium, i.e., when $\dot{\mathbf{w}} = 0$, it must hold that $\mathbf{v}(\mathbf{q}) = \mathbf{v}(\mathbf{q}, \mathbf{w})$. On the other hand, if at any time $\dot{\mathbf{w}}(t) = 0$ but $\mathbf{w}(\mathbf{q})$ is non-constant in the neighborhood of $\mathbf{q}$ then $\mathbf{w}(t+s)$ must differ from $\mathbf{w}(\mathbf{q}(t+s))$ provided that s is sufficiently small. This means that $\mathbf{w}$ cannot be kept ideal unless $\mathbf{w}(\mathbf{q})$ is constant. Below it will be shown that under some well defined conditions $\mathbf{w}$ can be kept as close to the ideal control signal as desired by choosing large enough Λ.

To proceed in this direction let us rewrite Equations (10) based on the variable

$$\mathbf{z} = \hat{\mathbf{A}}(\mathbf{q})\mathbf{w} - \Big(\mathbf{v}(\mathbf{q}) - \hat{\mathbf{v}}(\mathbf{q})\Big). \tag{11}$$

Note that if and only if $\mathbf{w} = \mathbf{w}(\mathbf{q})$ then $\mathbf{z} = 0$. Thus $\mathbf{z}$ may be viewed as an error variable. Equations (10) now take the form

$$\begin{aligned} \dot{\mathbf{q}} &= \mathbf{v}(\mathbf{q}) + \mathbf{z} \\ \dot{\mathbf{w}} &= -\Lambda \mathbf{A}^{-1}(\mathbf{q})\mathbf{z}. \end{aligned} \tag{12}$$

These equations show that the plant approximately follows the prescribed speed field provided that $\mathbf{z}$ is small. As a special case we mention that if there is no perturbation at all, then $\mathbf{w}$ converges to zero at an exponential rate. This might be seen directly from Equations (11) and (12). In the following the perturbed case will be considered and we show that the error can be kept small.

3.1 The ultimate boundedness of the feedback error

Let us denote by $\lambda_{\min}(\mathbf{A})$ the *singular value* of the quadratic matrix $\mathbf{A}$, that has the least *absolute* value. Of course, $\lambda_{\min}(\mathbf{A}) > 0$ holds if and only if $\mathbf{A}$ is positive definite. Let us denote by $||\cdot||$ the Euclidean norm. We use the same notation for the Euclidean norm of vectors, the induced Euclidean norm of matrices and tensors.

We will assume that the perturbation of $\mathbf{A}(\mathbf{q})$ is decomposed as

$$\hat{\mathbf{A}}(\mathbf{q}) = \mathbf{D}(\mathbf{q})\mathbf{A}(\mathbf{q}). \tag{13}$$

The following theorem gives the conditions of the uniform ultimate boundedness of the error of tracking:

Theorem 1 *Assume that the perturbation of $\mathbf{A}(\mathbf{q})$ is given by Equation (13) and the perturbation of $\mathbf{b}(\mathbf{q})$ is given by $\hat{\mathbf{b}}$. Suppose that $\mathbf{A}(\mathbf{q})$, $\mathbf{b}(\mathbf{q})$, $\mathbf{v}(\mathbf{q})$ and $\mathbf{D}(\mathbf{q})$, $\hat{\mathbf{b}}(\mathbf{q})$ have continuous derivatives and that the following constants are positive:*

$$a = \inf\{\, ||\mathbf{A}(\mathbf{q})|| \mid \mathbf{q} \in D\,\} \tag{14}$$
$$d = \inf\{\, ||\mathbf{D}(\mathbf{q})|| \mid \mathbf{q} \in D\,\} \tag{15}$$
$$\lambda = \inf\{\, \lambda_{\min}(\mathbf{D}(\mathbf{q}) + \mathbf{D}^T(\mathbf{q})) \mid \mathbf{q} \in D\,\}. \tag{16}$$

Then for all $\epsilon > 0$ there exists a gain Λ and an absorption time $T > 0$ such that for all $\mathbf{z}(0)$ that satisfy $||\mathbf{z}(0)|| < K\Lambda$ it holds that $||\mathbf{z}(t)|| < \epsilon$ provided that $t > T$ and the solution can be continued up to time t. In other words the error of tracking $\mathbf{z}$ is uniformly ultimately bounded[7] Here K is a fixed positive constant and $\mathbf{z}(0)$ denotes the initial value of $\mathbf{z}$. Further, $\Lambda \sim 1/\epsilon$ and $T \sim ||\mathbf{z}(0)||/\lambda$.

For convenience $\mathbf{D}(\mathbf{q}) + \mathbf{D}^T(\mathbf{q})$ will be called the symmetrized perturbation matrix and will be abbreviated to SP-matrix. Further, we say that a perturbation of Equation (1) is *signproper* if λ (defined by Equation (16)) is positive. If $\lambda = 0$ then we say the controller represents the inverse dynamics of the plant *semi-signproperly*. The signproperness condition means roughly that SDS can stabilize the control loop only if the angle between the effect of the control

[7] Our definition of uniformly ultimately boundedness is given in Appendix A.

signal on the perturbed and the unperturbed plants is smaller than the right angle for any control signal. The other two conditions, which were posed on the norms of $\mathbf{A}(\mathbf{q})$ and $\mathbf{D}(\mathbf{q})$, mean that these matrices should be uniformly non-singular on the domain of the plant. Now we give an outline of the proof (details can be found in Appendix B).

Proof. The proof is an based on a modification of Liapunov's second method (see Appendix A): Let us take $V(\mathbf{q},\mathbf{z}) = \mathbf{z}^T\mathbf{z} = ||\mathbf{z}||^2$ as a Liapunov function candidate. Computing $\dot{V}(\mathbf{x})$ we have that it is the sum of a negative definite part, namely $-\Lambda\mathbf{z}^T\Big(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})\Big)\mathbf{z}$ and another part – let us call it $\mathbf{g}$ – whose magnitude may be estimated from above as $A_1||\mathbf{z}||^3 + A_2||\mathbf{z}||^2 + A_1||\mathbf{z}||$, where A_1, A_2 and A_3 are fixed constants that depend on the properties of the plant and the perturbation. When the absolute value of the negative definite part is larger than the sum of the absolute values of the other terms then the whole expression is negative. This region contains a ring given by the radii $K\Lambda$ and k/λ, where K, k are fixed and depend on A_1, A_2 and A_3. This proves the theorem. Moreover one sees that the ring can be enlarged both towards zero and towards infinity by choosing larger Λ values.

3.2 Discussion of the theory

In order to develope some insight in the working of SDS control here we consider some special cases of the above theorem for various types of plants. We start with the most simple case:

It is easy to show that the undesirable term $\mathbf{f}$ disappears if the plant's equation is given by

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{u} + \mathbf{b},$$

and the perturbation and the vector field to be followed are constant. Indeed, in this case $\mathbf{D}'(\mathbf{q}) = 0$, $\mathbf{A}'(\mathbf{q}) = 0$, and $\mathbf{d}(\mathbf{q}) = \mathbf{d}$ is constant and thus $\mathbf{d}'(\mathbf{q}) = 0$. Consequently, in this case $\dot{V}$ is negative definite and the original theorem of Liapunov applies. Thus we have the following special case:

Proposition 2 *If* $\mathbf{A}(\mathbf{q})$, $\mathbf{b}(\mathbf{q})$, $\mathbf{D}(\mathbf{q})$, $\mathbf{r}(\mathbf{q})$ *and* $\mathbf{v}(\mathbf{q})$ *are constant fields then* $\mathbf{z}$ *converges to zero and* $\mathbf{w}$ *converges to* $\mathbf{w}(\mathbf{q}) = (\mathbf{E} - \mathbf{A}^{-1}\hat{\mathbf{A}})\mathbf{u} + \mathbf{A}^{-1}(\mathbf{b} - \hat{\mathbf{b}})$ *as time goes to infinity.*

Assume next that the plant is linear, i.e.,

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{u} + \mathbf{B}\mathbf{q}.$$

Then using the above argument one gets that the error signal $\mathbf{z}$ is ultimately bounded in the region

$$K = \{\, \mathbf{z} \mid ||\mathbf{z}|| > \max\left(\frac{2||B - \hat{B}||}{\lambda\Lambda}, \sqrt{\frac{2||B - \hat{B}||v}{\lambda\Lambda}}\right) \},$$

where $v = \sup_{\mathbf{q}} ||\mathbf{v}(\mathbf{q})||$ provided that $\mathbf{u} = \mathbf{u}(\mathbf{q}) \equiv$ const. That is, in this special case we arrive at the original concept of ultimate boundedness [8]. This is because the term of order $O(||\mathbf{z}||^3)$ disappears from $\mathbf{f}$.

These special cases are of particular importance when considering point-to-point tracking problems since then by choosing a large enough value for Λ the plant can be kept in a small region of the desired end-point. If the small region allows to take a first-order approximation of the plant's dynamics then the error of tracking converges to zero. Recently, we have run computer experiments for controlling a bioreactor, which proved challenging for conventional controllers and was suggested as a control benchmark problem in (Ungar, 1992). Such a reactor shows chaotic open-loop behaviour. Our results with

[8] See the appendix for the discussion of various ultimate boundedness concepts.

these controller confirm the above predictions of the theory: we considered set-point tracking tasks when one parameter of the bioreactor was perturbed and we observed that the error of tracking reduced to zero. Results are presented elsewhere (Szepesvári and Lőrincz, 1996b; Szepesvári and Lőrincz, 1996d).

An interesting question is whether for classical mechanical models the perturbation of the "geometry" and other physical properties of the plant result in uniformly positive definite perturbation or not. The answer is positive at least for the following special case. Consider the robot arm working in the three dimensional space with 3 degrees of freedom (see Fig. 1).

We assume a simplified model of the arm's dynamics that seems to be a "reasonable compromise between system complexity (and thus realism) and ease of implementation" (Anderson and Miller, III, 1992). The model is complete in that all joint coupling terms (centripetal and Coriolis torques, variable effective moments of inertia, etc.) are included. It is still an idealized model, however, in that all masses are assumed to be placed at discrete points and effects such as drive train friction are not modelled. The arm is similar to the three major axes (base, upper arm, and forearm) of typical industrial robots. We investigate the properties of the symmetrized perturbation matrix of this robot arm provided that the arm grasps or releases an idealized object (i.e., the mass of the end point changes). Of course, the perturbation matrix is non-linear. We prove – by elementary but rather tedious calculations – that the SP-matrix is positive definite and even that it is uniformly positive definite. The calculations are given in Appendix C. Note that since the dynamics of the robot arm is of second order the present results do not apply directly to this case. The control of higher order plants will be discussed later.

4 Compensatory control by static and dynamic state feedback: consequences for neurocontrollers

In this section we treat the inverse dynamics neurocontrollers within the proposed control scheme. The use of neurocontrollers – because they represent a large and powerful class of *adaptive controllers* – is important when the dynamics of the plant is not known in advance or uncertainties may be present in the dynamics. Adaptive controllers learn to control a plant from control samples. Direct methods aim to develop a control rule without explicitly finding a model of the plant. Indirect methods first go through an identification stage to establish a model before applying other techniques to find the control policy (Vemuri, 1993). Neurocontrollers can also be classified according to how the training data is used for learning. On this basis we distinguish *variational* and *non-variational* learning schemes. In the case of variational learning an error is computed from a desired and actual response, whereas for non-variational schemes there is no desired response. Indirect methods tend to utilize variational schemes, and direct methods utilize both variational and non-variational schemes.

Another classification of learning schemes is based on whether or not learning is interleaved with problem solving. In the former case we say that the learning is *on-line*, otherwise it is *off-line*.

Before discussing stability questions arising when the controller is adapted on-line, first, we show that *the precision of tracking may be increased* if one utilizes a *learnt* neurocontroller, which represents the inverse dynamics of the plant *approximately*, within *SDS control* scheme. This question is of great importance since the inverse dynamics of the plant may not be exactly reproduced using a previously fixed set of models (i.e., with a fixed architecture and adjustable parameters). The error obtained by choosing the best model from the set of possible models is called the *structural approximation error* while the error resulting from sub-optimal weights is called the *learning error*.

To see that the above statement holds, assume that the plant's equation is given by Equation (1) and assume that we approximate $\mathbf{A}^{-1}(\mathbf{q})$ by $\mathbf{P}(\mathbf{q})$ and $\mathbf{b}(\mathbf{q})$ by $\mathbf{s}(\mathbf{q})$. Then, of course, $\mathbf{A}(\mathbf{q})$ "approximates" $\mathbf{P}^{-1}(\mathbf{q})$. Now let us imagine that the inverse dynamics of our controller is "exact", i.e., the plant's equation is given by

$$\dot{\mathbf{q}} = \mathbf{P}^{-1}(\mathbf{q}) + \mathbf{s}(\mathbf{q}) \tag{17}$$

Now, (1) is thought of as the perturbed system and (17) as the unperturbed system. If we apply Theorem 1 we get that under some smoothness conditions and provided that $\inf_{\mathbf{q}} \lambda_{\min}(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})) > 0$, where $\mathbf{D}(\mathbf{q}) = \mathbf{A}(\mathbf{q})\mathbf{P}(\mathbf{q})$, then for large enough gains the error of feedback (in other words the error of tracking) is UUB and the ultimate bound on error is proportional to $1/\Lambda$. The positivity of the symmetrized perturbation matrix follows if $\mathbf{P}$ approximates $\mathbf{A}^{-1}$ sufficiently closely. Without the feedback signal, i.e., when $\Lambda = 0$, the (ultimate) boundedness of the error cannot be guaranteed (a few examples are given in the simulations).

Thus if the above symmetrized perturbation matrix is uniformly positive then the use of the neurocontroller even during its learning phase for SDS Control seems to be advantageous. On the other hand, if the initial controller does not represent the plant (semi-)signproperly then one should be cautious in using the controller for SDS Control. One should therefore have a signproper initial guess of the inverse dynamics of the plant in order to allow learning and feedback to work simultaneously. There are two ways to achieve this. First, one may initialize the controller so that it realizes the everywhere zero function, or one may prelearn a 0th stage model until it is signproper.

However, even for signproperly initialized controllers the system may become unstable during learning, depending on the learning law. This question is considered in the rest of this section. First we consider direct methods (Grossberg and Kuperstein, 1986; Psaltis et al., 1988; Widrow et al., 1978). The usual way of implementing the direct method is the following: a random action (control signal) is tried and the effect of the action (e.g., the direction of motion) is observed. Then we associate the effect with the action that caused it. In this case while learning there is no external signal to follow and thus there is no error term. It is therefore meaningless to compute the dynamic state feedback signal. If the learning phase (i.e., the self-generation of examples) is completed and the controller should track an external signal, then feedback can be switched on. If the adaptivity of the controller is still retained then both the feedback and the learning of the controller may work simultaneously. (Alternatively, this working mode can be assumed from the beginning.) Note that for proper learning the FFC should always associate the true control signal (i.e., the sum of the control signals of the FFC and the FBC) with the actual movement (effect). However, this learning mode requires a well designed external signal to follow in order to ensure the ergodicity of the plant's trajectory as well as exhaustive sampling from the space of control signals (Narendra and Monopoli, 1980).[9] Note that during learning the tracking of the desired trajectory is more precise with SDS Control than without it and this means that if learning is stable without the SDS Feedback Control then the stability of learning is ensured with it too. Furthermore at the end of learning the compensatory control signal becomes small: it has to compensate only the structural approximation error.

The other method, known as the indirect method (or model differentiation) (Jordan, 1990; Werbos, 1988; Widrow, 1986) requires a well designed external signal to follow. In this case the controller makes an informed guess as to which control signal should give rise to the provided external signal. This

[9] Note that exhaustive sampling of control signals restricts the range of control problems where direct methods can be used. For high dimensional control spaces exhaustive sampling may take too long to be practical.

guessed control signal results in a movement or effect that is usually different from the external signal. The difference between these two is then used as an error term. Then, the parameters of the controller are modified so that the error is reduced (usually only on average). Note that here the error is a state or speed error. From this error one should compute the error of the control signal, i.e., the obtained error should be "back propagated" through the plant dynamics. To this end the dynamics of the plant is modelled first (or assumed to be known) and then the model is used to back-propagate the error: thus learning is indirect – it requires an analytical model of the plant's dynamics. However, from our point of view it is more important that in the case of the indirect method the (state or speed) error signal is available during the whole learning phase, i.e., both the SDS Feedback Control and the learning might take place simultaneously. However, this simultaneous use might prevent or at least delay the learning of the true inverse dynamics since the control and thus the inverse dynamics model may seem more precise than they are. Another problem is that overcompensation may render the learning process unstable: large errors in trajectory tracking may be caused by both the overcompensation or the imprecise feedforward control signal. Consequently, this learning method should be cautiously used together with dynamic state feedback. Further research is needed to clarify this point. An appropriate starting point might be to consider the "feedback-error learning" models (Lewis et al., 1993; Miyamoto et al., 1988) when one replaces the adaptive feedback controller by a theoretically justified and thus stable feedback controller that can provide the error signal for the training of the inverse dynamics controller and can stabilize the control loop.

As can be seen from the above discussion direct inverse modeling provided a better fit with SDS Control. In the next section we describe some simulation results with our neurocontroller that make use of direct inverse modeling and non-variational, Hebbian type learning in a CMAC-like architecture (Marr, 1969; Albus, 1971; Miller, 1987; Fomin et al., 1994; Szepesvári and Lőrincz, 1995).

5 Computer simulations

In this section results of computer experiments are presented. The aim of this section is to *illustrate* the theory and the working of the compensation mechanism by simulations. The plant's equation, the neurocontroller and also the perturbations were kept as simple as possible. Despite this simplicity the model is quite complex and goes beyond the limits of the theory. This is because we consider simplified *sensorimotor control*, i.e., some aspects of sensory coding are included. The simulations support the self-improving nature of SDS Feedback. First, we introduce the neurocontroller used in the simulations. This neurocontroller is called the PDA controller since it maps **p**osition-**d**irection pairs to **a**ctions.

5.1 The PDA neurocontroller

The PDA controller was suggested in (Fomin et al., 1994; Szepesvári and Lőrincz, 1995) for closed-loop sensorimotor control as an inverse dynamics controller. The basis of the neurocontroller is a path planning algorithm that uses harmonic functions to encode the collision free trajectories (Connolly and Grupen, 1993; Glasius et al., 1995; Keymeulen and Decuyper, 1992; Lei, 1990; Morasso et al., 1993; Tarassenko and Blake, 1991). We have extended this algorithm to include the learning of an approximate inverse dynamics control of the plant to be controlled (Fomin et al., 1994; Szepesvári and Lőrincz, 1995). Now, we briefly describe the working of this neurocontroller.

Let us first consider the path planning part of the neurocontroller (see Fig. 3).

Sensory neurons provide the input to the network. They may be thought of as discretizing the state space of the plant. Another layer of neurons, the spatially tuned neurons of the geometry discretizing layer develop a problem dependent discretization of the state space: the weights of these neurons are developed in a self-organizing process (a winner-takes-all mechanism). The path planning problem is given in terms of discretization point occupancies. Any discretization point, called *spatially tuned neuron*, can be occupied by an obstacle, the plant, or the target. It is also possible that more than one discretization point is occupied by an object. This results in a coarse coded, distributed representation of the object that in turn results in smoother control signals. Between *neighboring* discretization points laterally oriented *geometrical* connections allow activation to spread: when the activity spreading on the discretization system settles we say that an activity field is formed. We call this the equilibrium activity map. The plant should move along the "gradient" of this activity map. (Here, and below we consider the neural network as a numerical approximation of a continuous system. If the concepts are used with care then one can talk about the gradient field in the discretized system, i.e., the approximation of the corresponding quantity in the continuous system. The expression, directional derivative will also be used in this way.) The activation spreading equation that forms this activity map is of the *diffusion type* and thus the equilibrium field has only one minimum and one maximum ((Connolly and Grupen, 1993; Lei, 1990)). Moreover, these extremes correspond to the position of the plant and the goal, respectively. This is achieved by allowing unit inflow at the position of the plant and unit outflow at the position of the goal. Obstacles may be avoided by setting up appropriate boundary conditions, such as – for example – by forbidding the activity to spread along the lateral connections of the corresponding neurons thus approximating the Neumann boundary condition. If the gradient of the equilibrium map is followed it results in a path from the plant's actual position to the goal position. For on-line motion control the activity map should be continuously upgraded.

This is important if either the obstacles or the goal is moving, or the controller or the sensors are imperfect. For continuous motion the changes of the equilibrium activity map are differential and thus the relaxation time of the spreading activation model is a differential quantity. This enables fast, on-line path planning.

It is also a nontrivial task to follow the gradient of the equilibrium activity map. This task, formalized in Section 2.2, is called *tracking a prescribed speed field*. Solving this task requires a knowledge of the inverse dynamics of the plant. However, in order to ensure collision free motion it is enough to follow a proportional speed field, where the proportionality can be a continuous function of the state provided that the proportionality has a positive lower bound (Szepesvári and Lőrincz, 1996c). This eases the learning problem since it suffices to know a mapping that is proportional to the inverse dynamics mapping. Such a proportional mapping is called the Position-Direction to Action (PDA) mapping. It has been shown in (Fomin et al., 1994; Szepesvári and Lőrincz, 1995) that the realization and learning of PDA mapping can be solved by simple Hebbian learning and by extending the path planning architecture with two additional neuronal layers. To this end we have equipped the path planner neural net with interneurons and control (command) neurons (see Fig. 3).

Control neurons should emit the control signal that moves the plant along the gradient. Interneurons are situated at lateral connections (to each connection there correspond two directives and thus two interneurons) and are connected to the control command neurons by adaptive connections. Equivalently, interneurons can be considered to store control commands. The working mechanism of the neurocontroller is as follows: An interneuron is enabled to "fire" only if it is in the neighborhood of the plant's state represented on the discretization layer. This localization (similar to CMAC and Radial Basis Function methods) enables state dependent non-linear inverse dynamics to be realized. The firing of an interneuron is proportional to the extent of flow along the corresponding connection (i.e., the firing of an interneuron is an approximation of the directional derivative of the steady state activity map). Every firing interneuron sends its control command multiplied by its firing to the control neurons. The control neurons sum up their incoming activities and emit the computed value. This procedure approximates the PDA mapping provided that the control commands of any interneuron move the plant along the direction of the corresponding connection. It has been shown that this approximation is exact in the limit when the fineness of the discretization approaches zero and if the local neighborhood of any given discretization point is spanned by the direction vectors corresponding to the neighboring interneurons of the discretization point under consideration.

The adaptation of the weights between interneurons and control command neurons is based on a general system inverse identification scheme that utilizes associative Hebbian learning: a randomly chosen control command and the interneuron signals are associated with Hebbian learning, where the interneuron signals are computed from the path planning problem where the "plant-position" and "goal-position" are the plant's initial position and the position after the execution of the randomly chosen control command, respectively. It has been shown that in this way the neurocontroller is capable of learning the correct control commands and further it learns a proportional mapping to the main value of the inverse dynamics (see Equation 5). The advantage of self-organized associative learning is that it can not be trapped in local minima. However, its disadvantage is that exhaustive sampling of a high dimensional control space can be very time consuming. Further details of the algorithm as well as computational results to learning may be found in (Fomin et al., 1994; Szepesvári and Lőrincz, 1995).

5.2 *Details concerning the sensorimotor loop*

Both the state space of the plant and the control space are two-dimensional and square-shaped. The objects (the plant and the goal object) are represented by their co-ordinates.

To provide some flavor of a real situation of sensorimotor control the objects of the state space (for example the plant itself and the goal) are projected on an artificial retina of dimensions 20×20. The artificial retina is assumed to "cover" the state space and one may assign co-ordinates to pixels as the center of the pixel in the state space. The image of objects being in the state space were "imaged" onto the artificial retina by using the following method: Every object creates nonzero responses at the pixel closest to its position and weaker responses at the 8 nearest neighbor pixels. In all cases only 9 pixels were excited. The excitation of a pixel is limited to the interval $[0, 1]$ and was supposed as being inversely proportional to the distance between the pixels and the position of the object.

The discretization layer of the neurocontroller is a grid whose dimensions (20×20) match those of the artificial retina. This enables the optimal coverage of the discretizing neurons over the image space of the artificial retina. The space of possible images was discretized by the discretization layer. The feedforward weights of every neuron of the discretization layer form a localized Gaussian shaped spatial filter. The proximity relations between discretizing neurons are based on the neighborhood relations of their spatial filters (Szepesvári et al., 1994; Szepesvári and Lőrincz, 1996a). The spatial filters and the proximity relations between them can be learnt by self-organization (Szepesvári et al., 1994; Szepesvári and Lőrincz, 1996a). In these experiments, however, the spatial filters, the proximity relations as well as the control commands stored by the interneurons were *prewired in an ideal fashion* in order to limit the range of possible errors to *structural approximation errors* only.

In the following we describe 3 illustrative simulations of increasing complexity in the perturbation. The perturbations are of three basic types: homogeneous additive, homogeneous multiplicative and inhomogeneous multiplicative. Also, inhomogeneous perturbations such as errors are present even in the most simple experiments due to imperfect sensing and other structural approximation errors. Results concerning more complex perturbations are not presented since they would not provide further information about the compensation mechanism. We discuss the results immediately after describing the simulations.

5.3 Additive perturbation

In these experiments the plant's equation and the perturbed equation are given by

$$\dot{\mathbf{q}} = \mathbf{u} \tag{18}$$

$$\dot{\mathbf{q}} = \mathbf{u} + \mathbf{b}, \tag{19}$$

respectively, where $\mathbf{b}$ is a fixed nonzero vector. The control task is given by the constant speed vector field $\mathbf{v}(\mathbf{q}) = \mathbf{v}_{\mathrm{WE}}$, where $\mathbf{v}_{\mathrm{WE}}$ points from west to east.[10] Only the FBC was used in this experiment. The perturbation vector is perpendicular to the speed vector field and points from south to north. Typical trajectories of the plant with and without compensation are shown in the upper left subfigure of Fig. 4. (For brevity we call the plant that is controlled by the feedforward controller alone **the plant without compensation** and the plant that is controlled by the static and dynamic state feedback control **the plant with compensation**.) The trajectory of the plant without compensation moves along a straight line deviating strongly from the desired

[10] In this case the path planning algorithm was turned off and the interneuron activities were directly preset to their desired values.

direction. The angle between the horizontal line and the trajectory of the plant reflects the relation between the magnitude of the perturbation vector $\mathbf{b}$ and the speed vector $\mathbf{v}_{WE}$. The trajectory of the plant that used the compensation mechanism starts at the same angle but quickly curves back and relaxes at the desired west to east (horizontal) direction. This shows that the plant with the compensation mechanism is able to track asymptotically the desired speed field. This property becomes more apparent if one considers the features of compensation in the upper right part of Fig. 4. This figure shows the absolute value of the angle between the horizontal direction and the actual speed (dashed line) and the Euclidean length of the compensatory vector, $\mathbf{w}$ (solid line). At the edges of the state space the estimation of the gradient and thus also the estimation of the correct control command are effected by the break of symmetry. The figure shows that the direction approximates the desired direction, and the length of the compensatory vector fluctuates around a constant value. This constant value corresponds to the length of $\mathbf{b}$ since in the ideal case $\mathbf{w}$ would converge to $-\mathbf{b}$ (Proposition 2). Note that due to the approximation errors perfect convergence cannot be achieved, even in this simple case.

5.4 Multiplicative perturbation

In these experiments the unperturbed plant's equation and the task were the same as in the previous section, but as the perturbation affected the matrix "$\mathbf{A}(\mathbf{q})$": the perturbed plant's equation is now given by

$$\dot{\mathbf{q}} = \mathbf{F}(\alpha)\mathbf{u}, \tag{20}$$

where the matrix $\mathbf{F}(\alpha)$ is the rotation matrix corresponding to the angle α. The matrix $\mathbf{F}(\alpha) + \mathbf{F}^T(\alpha)$ is positive definite if and only if $-\pi/2 < \alpha < \pi/2$. For demonstration purposes we have chosen the angle $\alpha = \pi/4$. Again, only the FBC was used in this experiment. The lower left part of Fig. 4 shows the trajectory of the plant without compensation (plus signs) and with compensation (diamonds). Again, the trajectory of the plant is somewhat distorted by the edge effects. The plant without compensation moves close to an angle of $\pi/4$ to the desired direction. The plant with compensation mechanism starts in the same direction but efficiently compensates after a very short distance. The directional changes of the uncompensated plant are due to the rough discretization. These directional changes are also diminished by the compensatory signal. Note that in these experiments the sampling of the controller is more frequent than in the experiments of the previous section. This increased sampling frequency corresponds to an FBC with higher Λ value. This is the reason why the trajectory of the plant with compensation deviates less than in the previous experiment. In the ideal case $\mathbf{w}$ would converge to $(\mathbf{F}^{-1}(\alpha) - \mathbf{E})\mathbf{v}_{WE}$.

5.5 Inhomogeneous perturbation

Finally, let the perturbed plant's equation be given by

$$\dot{\mathbf{q}} = \mathbf{F}(\alpha(\mathbf{q}))\mathbf{u}, \tag{21}$$

where $\mathbf{F}(\cdot)$ is the rotation matrix introduced in the previous section, but here the rotation angle is position dependent:

$$\alpha(\mathbf{q}) = \begin{cases} \pi/2(1 - d(\mathbf{q}, \mathbf{c})); & \text{if } \pi/2(1 - d(\mathbf{q}, \mathbf{c})) > 0 \\ 0; & \text{otherwise.} \end{cases}$$

Function d gives the Euclidean distance between $\mathbf{q}$ and $\mathbf{c}$, where vector $\mathbf{c} = (0, 0)$ is the center of the state space $[-1, 1]^2$. Thus the rotation is the greatest at the center and reaches zero as we approach the edges of the state space. Note that the perturbation is nondifferentiable along the unit circle and that $\mathbf{D}^T(\mathbf{c}) + \mathbf{D}(\mathbf{c}) = 0$; that is, the symmetrized perturbation matrix is non-uniformly positive definite. It is not expected that such small deviations from the theory seriously limit the use of the proposed SDS Controller. The simulations show that this is indeed the case.

The task is changed to move to the upper right corner, i.e., $\mathbf{v}(\mathbf{q})$ is proportional to $(1,1) - \mathbf{q}$. (The scale on the figures correspond to the 20×20 discretization.) At the same time the initial position of the plant becomes the lower left corner. Now, the path planner algorithm is also in effect. This way the optimal path would go through the center ($\mathbf{c}$) of the state space, where the perturbation is the strongest. Typical trajectories are shown in the lower right part of Fig. 4. The trajectories plotted by diamonds, plus signs, and squares correspond respectively to the trajectories of the unperturbed plant, the perturbed plant without compensation, and the perturbed plant that uses the compensation mechanism.

Note that the plant that uses the compensation mechanism overcompensates the perturbation. This might be observed at the end of the trajectory. The overcompensation is due to the strongly nonlinear nature of the perturbation: by the time the controller compensates for the strongest perturbation in the middle of the space the perturbation decreases quickly and overcompensation results. The overcompensation is the consequence of the integration time we applied in the feedback controller. This overcompensation can be made arbitrarily small by using faster compensation, i.e., by increasing the gain of Λ.

5.6 Notes

The presented numerical examples show the difference between feedforward and feedback control strategies. If the problem is perfectly learnt then the control problem is solved without error. Feedforward control is fast; however, it requires experience and learning. If, on the other hand, the problem is not perfectly learnt then feedforward control may become unstable and feedback should be applied to stabilize the control loop. Generally speaking, feedback takes time since the error has to develop and, on the other hand, the detection and initiation of the feedback control signal may take time. If the feedback signal is fast and strong then the system may become sensitive to noise; this problem sets limitations on the time required to apply the feedback control strategy. Another important consequence of using the compensatory mechanism, which was supported by our simulations, is that it corrects structural approximation errors, too.

6 Discussion

In this section we discuss some questions concerning SDS Feedback Control. First we compare conventional feedback controllers with our method, then the effect of nonstationary perturbations as well as sensitivity to noise are discussed. We then consider how the proposed control scheme (speed field tracking using SDS Control) can be utilized to control higher order plants and finally, we weigh some open questions.

6.1 Comparison to other methods

The main difference between conventional and SDS Control is that in designing a conventional feedback controller, some knowledge of the plant's dynamics (or its inverse dynamics) is required. If the dynamics of the plant is not known then one must use an adaptive method; in other words, one should teach an appropriate controller, e.g. an inverse or forward system identification should take place first. If an analytical model of the dynamics of the plant is learnt then one still has the opportunity to design a dynamic state feedback controller in the conventional way. However, as it has been shown here, the learnt inverse dynamics controller can also be used for dynamic state feedback. Moreover, the resulting compound controller can compensate perturbations quickly under relatively mild conditions. Our opinion is that since our FBC closely fits the controlled plant our feedback may well be faster than a linear feedback controller would be. Moreover, as it has been discussed before the simultaneous learning of the inverse dynamics model and its utilization within the SDS Feedback is possible. The preferred learning method is then direct system identification (see Section 4). In terms of the approach presented in this paper the adaptive scheme is most advantageous in cases when the plant's dynamics is completely or partially unknown.

It is important to note here that the primary goal of using an inverse dynamics controller in the SDS control mode is to stabilize control loops. Thus the role of the proposed scheme is similar to the role of PD/PID controllers which are widely used for this purpose (see, e.g. (Miyamoto et al., 1988; Lewis et al., 1995)). However, SDS control overcomes several limitations of PD/PID based stabilization. First, since PD/PID controllers transform state space error into compensatory control signals via matrix multiplication by a positive definite matrix, they require that the dimension of the state space and the control signals be the same. However, often these dimensions are not the same. This restriction is overcomed in our method, since the model of inverse dynamics is used to compile state space signals into control signals. Second, PD/PID controllers require special plants: for our plant model they would require that $\mathbf{A}(\mathbf{q})$ is positive definite. Although this assumption holds for robot-arms, at the same time for general non-linear plants, e.g. for chemical plants this assumption may be too strong.

In the proofs we assumed that the perturbation is stationary, i.e., if it is switched on then it remains unchanged forever. However, this assumption is unrealistic. As far as nonstationary perturbations are concerned the proof of Theorem 1 should be modified. In fact, if the perturbed system is given by

$$\dot{\mathbf{q}} = \mathbf{A}(t, \mathbf{q})\mathbf{u} + \mathbf{b}(t, \mathbf{q})$$

then $\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{z}$ would contain additional terms such as, e.g., $\frac{\partial}{\partial t}\mathbf{A}(t, \mathbf{q})$. Note that these additional terms do not increase the order of $\mathbf{f}$ – considering $\mathbf{f}$ as a polynomial of $\mathbf{z}$. This means that if the changes are slow, i.e., these terms are bounded, then for large enough Λ one may keep the ultimate boundedness of the error signal. However, it is mentioned that in real world applications these changes are usually fast (e.g., when a robot arm grasps a heavy object). In such a case the error signal may become very large and the system may become unstable.

Any noise disturbing the control loop can be considered as a nonstationary perturbation, but noise usually does not admit a time-derivative and what is more the variation of noise over time is not even bounded. Therefore, our machinery cannot be applied unless these unpleasent features of noise are ruled out. In what follows we assume that the noise affecting our system has bounded amplitude and bandwidth. Then an important issue is where the noise enters the system. One usually assumes that noise affects the output of the controller. In such a case the noise can be viewed as an additive perturbation of the plant and can be compensated by our SDS Feedback mechanism with high enough Λ values. If the noise, on the other hand, affects the inputs of the controller, e.g., the state of the plant, then it can be viewed as the perturbation of the inverse dynamics model. The boundedness of the noise-amplitude implies the boundedness of the corresponding perturbation, seeing that, the inverse dynamics is by assumption a differentiable function with uniformly bounded derivatives. In this case, again, the noise can be compensated by our system. Note that this is not so for conventional linear feedback controllers. The most delicate case of the SDS scheme is when the noise enters the system just before the compensatory vector is integrated, i.e., the noise affects $\dot{\mathbf{w}}$. Such a noise can easily make the system unstable, in spite of the fact that this type of noise results also in an additive perturbation of the plant, since the perturbation now takes the form

$$\Lambda \int_0^T n(t)dt,$$

where $n(t)$ denotes the noise. Now, even the boundedness of the integral cannot be ensured for the general case. Moreover, the amplitude of the perturbation will be proportional to Λ. This means that increasing Λ will also increase the perturbation of the system. This problem, however, is the problem of every *dynamic* state feedback controller provided if noise can enter precisely before the point where the compensatory control signal is integrated through time.

6.3 Controlling higher order plants

Since higher order systems can be written in the form of a first order system it follows that SDS Control of higher order plants is by no means problematical provided, that the plant may be equipped with the necessary control units. On the other hand, speed field design for higher order plants may become complicated. This can be demonstrated by rewriting the dynamics of a higher order plant in the form of a first order differential equation whereupon one may arrive at a singular matrix field, $\mathbf{A}(\mathbf{q})$. Consequently the inverse field of $\mathbf{A}(\mathbf{q})$ is non-unique (the dimension of the control space may be smaller than that of the "phase-space".) This means that not all speed fields can be tracked with zero error: the speed field to be tracked should be carefully designed.

One method that solves this problem is best illustrated by a second order plant for which only $\ddot{\mathbf{q}}$ is directly controllable (as in the case of a robotic manipulator), i.e., when the plant's equation is given by

$$\ddot{\mathbf{q}} = \mathbf{A}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{u} + \mathbf{b}(\mathbf{q}).$$

Assume that we have designed a speed field – still in the state space. Let us denote it by $\mathbf{v} = \mathbf{v}(\mathbf{q})$. Using this speed field we have to set up an acceleration field $\mathbf{a}(\mathbf{q}, \dot{\mathbf{q}})$ in an appropriate way since in the case of the robotic manipulator it is only the acceleration of the plant that can be controlled directly. In case of speed field tracking the speed to be tracked may be given simply in the form of the difference between the (possibly time dependent) desired position and the actual position. The analogy for "acceleration field tracking" is that the difference between the desired speed and the actual speed should be tracked:

$$\mathbf{a}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{v}(\mathbf{q}) - \dot{\mathbf{q}}.$$

Computer simulations indicate the stability of this approach (results will be published elsewhere).

6.4 Open questions

One shortcoming of our approach is that the perturbations should be sign-proper in order to keep the error term bounded. Experiments indicate, however, that humans are capable of compensating not uniformly positive definite perturbations extraordinarily rapidly (Young, 1969). In our framework such "structural" changes should be detected and the appropriate sign changes should be incorporated in the control. The detection of such changes may be based on the observation that with such a perturbation the error term keeps growing to infinity. However, the detection of the nature of an "inversion" seems to be far from trivial.

It is intriguing to think about fast action problems. Such actions may not allow the direct use of the feedback controller since the time is just far too short to develop the compensatory signal. However, the feedback controller may be used during the learning procedure: if the task to be controlled is such that it may be practiced starting from slow speed and the high speed action could be built up step by step, then the feedback controller may be utilized since the task is now slow enough and the feedback controller may play a role. If the very same learning procedure allows for the increase of the gain factor Λ during training then the feedback controller may be able to keep up with the needs of the feedforward control problem. In this way the controller can produce very precise motion that can be stored in a "procedural" memory for later reuse.

Another reason to change Λ that if the plant is highly non-linear then in some regions of the state space a lower gain is sufficient, while in other regions a higher gain is needed to ensure a certain stability. A state dependent gain would reduce sensitivity to noise. Another option is to use an adaptive (time varying) gain; the gain should be increased if the error of tracking is not sufficient and otherwise it should be decreased.

One can try to use some aspects of passivity (Goodwin and Sin, 1984; Landau, 1979; Lewis et al., 1993; Slotine and Li, 1991) of the plant. In this case one may start from the Liapunov function of the plant which satisfies the passivity criterion and use the components of this function to design a modified dynamic state feedback control as in (Lewis et al., 1995). It seems possible that for such passive systems the ultimate boundedness of the error can be extended to the whole error space. In this case the speed of nonstationary perturbations would not be limited by the magnitude of the gain factor.

7 Conclusions

We have shown that the so called SDS Controller is capable of compensating inhomogeneous, non-linear, non-additive perturbations of non-linear plants that admit an inverse dynamics. Such perturbations arise, for example, when a robot arm grasps or releases a heavy object. The SDS Controller is composed of two identical copies of an inverse dynamics controller. One copy acts as the original closed loop controller while the other identical copy is used to develop the compensatory signal. The advantage of this compound controller is that it can develop a control signal for unseen perturbations and thus can control the plant more precisely than the closed loop feedforward controller alone. Also SDS Feedback Control is advantageous to error feedback control since the feedforward controller can provide almost precise control signals. Roughly speaking, we have proven that an arbitrary non-linear perturbation can be compensated by our method provided that the perturbation is sign-proper. The proof is based on a customized Liapunov function approach. The compensatory signal is built up very rapidly. This follows from the general theory, but just for linear plants. For non-linear plants it was demonstrated by computer simulations. We have shown for a particular robot arm that the conditions of our stability theorem are implied when perturbations caused by changing the load of a robot arm are considered.

The main advantage of the suggested architecture is that the very same system is used for feedback and feedforward control and the learning problem is therefore relaxed since only one system is available for training. Learning and SDS Control may take place simultanously; the only thing to remember is that the training signal must always be the sum of the feedforward and feedback control signals. Another advantage is that the stability theorem requires only a qualitative model of the inverse dynamics. The use of the inverse dynamics model in the SDS Feedback Control mode can reduce the the approximation errors which may considerably relax the number of parameters that are required to achieve a given precision in control, This, in turn, may enable the use of fast learning, local approximation based neural networks, that are otherwise known to suffer from combinatorial explosion in the dimension of the state space. Finally, SDS Feedback can work for overcontrolled plants – a property which distinguishes SDS Feedback from other stabilization methods, such as PID controllers.

Appendix

A An extension to Liapunov's second method

Some notions on stability are needed for the subsequent developments. Let $\mathbf{R}$ denote the set of real numbers, and $\mathbf{R}^n$ denote the real n-dimensional vectors. Consider the autonomous system

$$\dot{\mathbf{x}} = f(\mathbf{x}), \tag{A.1}$$

where $\mathbf{x}$ is the element of D, D is a compact subset of $\mathbf{R}^n$, and f is a vector valued smooth function over D. The solution of Equation (A.1) corresponding to the initial condition $x(0) = \xi$, is denoted by $\varphi(t;\xi)$ ($\xi \in D$). It is assumed that the output of Equation (A.1) is

$$\mathbf{y} = h(\mathbf{x}),$$

where $\mathbf{y} \in \mathbf{R}^m$, ($m > 0$ integer) and h is continuous. Let $||\cdot||$ denote an arbitrary norm over $\mathbf{R}^m$ and let U be an arbitrary subset of $\mathbf{R}^m$. We say that the output of the above system is *uniformly ultimately bounded* (UUB) *w.r.t.*

the set U if there is a bound $b > 0$ and a number $T > 0$ such that for each solution $\varphi(t;\xi)$ for which $h(\xi) \in U$ it holds that $||h(\varphi(t;\xi))|| < b$ provided that $t > T$ and $\varphi(t;\xi)$ is defined for t. If $h(\mathbf{x}) = \mathbf{x}$, then we say that the *system* is UUB. Since $\varphi(t_1+t_2;\xi) = \varphi(t_1;\varphi(t_2;\xi))$ for all ξ and $t_1, t_2 > 0$, it follows that if the output of the system is UUB and $h(\varphi(t;\xi)) \in U$ for some ξ and t then for all $t' > t+T$ it holds that $||h(\varphi(t';\xi))|| < b$. T will be called the absorption time. It may be worth noting that in Ref. (La Salle and Lefschetz, 1961) only the system's uniform ultimate boundedness was considered and only w.r.t. $\mathbf{R}^n$. Another difference is that in our case uniformity refers to bound b as well as to the absorption time T whereas in (La Salle and Lefschetz, 1961) it refers only to the bound.

Let the Liapunov derivative of a real valued differentiable function $V = V(\mathbf{x})$ w.r.t. Equation (A.1) be denoted by $\dot{V}$ and given by

$$\dot{V}(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i}(\mathbf{x}) f_i(\mathbf{x}),$$

where $f_i(\mathbf{x})$ is the ith component of $f(\mathbf{x})$. The Liapunov derivative admits the noteworthy property that if $\mathbf{x}(t)$ is the solution of (A.1) then $\frac{\mathrm{d}}{\mathrm{dt}}(V(\mathbf{x}(t))) = \dot{V}(\mathbf{x}(t))$ holds for all t. The following theorem is a slight modification of a standard extension of Liapunov's second theorem (La Salle and Lefschetz, 1961).

Theorem 3 *Let us consider the autonomous differential equation given by Equation (A.1) and its output*

$$\mathbf{y} = h(\mathbf{x}).$$

where $h : \mathbf{R}^n \to \mathbf{R}^m$, h is continuous and $h(0) = 0$. Assume that we are given positive numbers $k < K$, and $\beta > 0$ and a real valued function $V = V(\mathbf{x})$ defined on D that has continuous partial derivatives. Assume that D contains *a neighborhood of zero. Let*

$$R = \{\, \mathbf{x} \in D \;\mid\; k \leq ||h(\mathbf{x})|| \leq K \,\}$$

and assume that the following conditions are satisfied.

(i) $V = W(||h(\mathbf{x})||)$*, where* W *is a strictly increasing function;*
(ii) if $\mathbf{x} \in R$ *then* $\dot{V}(\mathbf{x}) < -\beta$*.*

Then the output $\mathbf{y}$ *of Equation (A.1) is UUB w.r.t. the set* $\{\, \mathbf{x} \in D \;\mid\; ||h(\mathbf{x})|| \leq K \,\}$*, where the ultimate bound of the output is* k*.*

Note that since $h(0) = 0$ and h is continuous the region $\{\, \mathbf{x} \in D \;\mid\; ||h(\mathbf{x})|| < K \,\}$ contains a neighborhood of zero provided that D contains a neighborhood of zero.

Proof. Let us fix an arbitrary $\xi \in R$ and consider the function $v(t) = V(\varphi(t;\xi))$. It holds that $W(0) \leq v(t)$ and $v(0) \leq W(K)$. The latter inequality holds since if $\mathbf{x} \in R$ then $h(\mathbf{x}) \in S_k^K = \{\, \mathbf{y} \;\mid\; k \leq ||\mathbf{y}|| \leq K \,\}$, and thus $||h(\mathbf{x})|| \leq K$ and thus $W(||h(\mathbf{x})||) \leq W(K)$. Further $W(K) > W(0)$ by the strict monotonocity of W. According to the definition of Liapunov's derivative it holds that $v'(t) = \frac{\mathrm{d}}{\mathrm{dt}} v(t) = \dot{V}(\varphi(t;\xi))$. Thus $v'(t) < -\beta$ as long as $\varphi(t;\xi) \in R$. If one integrates both sides of this inequality and utilizes that $\varphi(0;\xi) = \xi \in R$ it leads to the inequality $v(t) \leq -\beta t + v(0)$ that holds from time $t = 0$ until $\varphi(t;\xi) \in R$. Since $v(t)$ is bounded from below by $W(0)$ we have that

$$W(0) \leq v(t) \leq -\beta t + W(K);$$

this holds until the solution is in R. Consequently the solution must leave R within the time

$$T = \frac{W(K) - W(0)}{\beta}$$

independently of the initial value of the solution, ξ. We claim that this T satisfies the conditions of the theorem. To see whether this is the case let $T(\xi)$ be the time when the solution leaves R. ($T(\xi)$ exists since S_k^K is compact and the output of any of the solutions is continuous.) We claim that $||h(\varphi(t;\xi))|| = k$. Let $t_1, t_2 \in [0, T(\xi)]$ such that $t_1 < t_2$. It then holds that $v(t_1) > v(t_2)$ and thus using Property i of V we get that $||h(\varphi(t_1;\xi))|| > ||h(\varphi(t_2;\xi))||$, i.e., *the norm of the output decreases while the solution is in* R. Since $\varphi(0;\xi) = \xi \in R$ then $||h(\varphi(0;\xi))|| \leq K$ and, consequently, by the time the solution leaves R the value of $||h(\varphi(t;\xi))||$ cannot be equal to K. Now let us prove that any solution that enters the open set $U = \{\mathbf{x} \mid ||h(\mathbf{x})|| < k\}$ remains within it until the solution can be continued. For the proof let us assume in contrast that the solution that entered U once, leaves it at time T_1. This means that $||h(\varphi(T_1;\xi))|| = k$ and there exists a time interval $(T_1 - \delta, T_1)$ during which the solution is still in U. Now let us consider $v'(t)$ during this interval. Since $v'(t)$ is continuous and $v'(T_1) < -\beta$ there is a subinterval of $(T_1 - \delta, T_1)$ during which $v'(t)$ is negative, i.e., v is decreasing. Let t be the element of this subinterval. We then have $v(t) > v(T_1)$. Using Property i again we get that $k = ||h(\varphi(T_1;\xi))|| < ||h(\varphi(t;\xi))||$, which contradicts the assumption that $\varphi(t;\xi) \in U$. Thus we have that if a solution enters the set U, it remains inside until it can be continued. Now, if ξ is such that $||h(\xi)|| \leq K$ then $||h(\varphi(t;\xi))|| < k$ holds for all $t > T$ for which $\varphi(t;\xi)$ is defined. That is, the considered system is UUB.

Moreover, it holds that for every solution that starts from $R_0 = \{\mathbf{x} \in D \mid \dot{V}(\mathbf{x}) < 0\}$ it holds that the output of the solution enters every sphere $S_k =$

$\{\, \mathbf{y} \mid ||\mathbf{y}|| < k \,\}$ within a finite time, provided that $R_0 \cap S_k \neq \emptyset$. This can be proved by varying β in the above proof. For our purposes a special case of Theorem 3 is needed. Let us assume that system (A.1) is decomposed into two parts:

$$\begin{aligned} \dot{\mathbf{x}}_1 &= f_1(\mathbf{x}) \\ \dot{\mathbf{x}}_2 &= f_2(\mathbf{x}), \end{aligned} \tag{A.2}$$

where $\mathbf{x}_1 \in \mathbf{R}^{n_1}$, $\mathbf{x}_2 \in \mathbf{R}^{n_2}$, $n_1, n_2 > 1$ and $n_1 + n_2 = n$. Further assume that the output of system (A.2) is $h(\mathbf{x}) = \mathbf{x}_1$. Then Theorem 3 reads as follows:

Corollary 4 *Let us consider the autonomous differential equation given by (A.2). Assume that the domain of this equation contains a neighborhood of zero. Let $||\cdot||$ denote an arbitrary norm on $\mathbf{R}^{n_1}$ and let*

$$R = \{\, \mathbf{x} \in D \mid k \leq ||\mathbf{x}_1|| \leq K \,\},$$

where $k < K$ are positive numbers and $\mathbf{x}_1$ denotes the vector formed from the first n_1 components of $\mathbf{x}$. Assume further that we are given a fixed positive number, β. Now suppose that there exists a real-valued function $V = V(\mathbf{x})$ defined on D, V has continuous partial derivatives on D, and V satisfies the following properties:

(i) $V(\mathbf{x}) = W(||\mathbf{x}_1||)$, where W is a strictly increasing function;
(ii) if $\mathbf{x} \in R$ then $\dot{V}(\mathbf{x}) < -\beta$.

Then the output $\mathbf{y} = \mathbf{x}_1$ of system (A.2) is UUB w.r.t. the set $\{\, \mathbf{x} \in D \mid ||\mathbf{x}_1|| \leq K \,\}$ and bound b.

This corollary states that under the required conditions Equation (A.2) is *partially uniformly bounded*. Some partial stability concepts were considered by Rumiantsev (Rumiantsev, 1957).

B Proof of the uniform ultimate boundedness of the error

In this section we give a detailed proof of Theorem 1. We will need the following functions:

$$\mathbf{r}(\mathbf{q}) = \hat{\mathbf{b}}(\mathbf{q}) - \mathbf{b}(\mathbf{q}) \tag{B.1}$$

and

$$\mathbf{d}(\mathbf{q}) = \mathbf{v}(\mathbf{q}) - \hat{\mathbf{v}}(\mathbf{q}). \tag{B.2}$$

Elementary calculations show that

$$\mathbf{d}(\mathbf{q}) = (\mathbf{E} - \mathbf{D}(\mathbf{q}))(\mathbf{v}(\mathbf{q}) - \mathbf{b}(\mathbf{q})) + \mathbf{D}(\mathbf{q})\mathbf{r}(\mathbf{q}). \tag{B.3}$$

The proof is the application of Corollary 4. Let us take $\mathbf{x}_1 = \mathbf{z}$, $\mathbf{x}_2 = \mathbf{q}$, and $V(\mathbf{x}) = \mathbf{z}^T\mathbf{z} = ||\mathbf{z}||^2$. Let us now compute $\dot{V}(\mathbf{x})$. First, let us take the time derivative of $\mathbf{z}$:

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathbf{z} = \hat{\mathbf{A}}(\mathbf{q})\dot{\mathbf{w}} + (\hat{\mathbf{A}}'(\mathbf{q})\dot{\mathbf{q}})\mathbf{w} + \mathbf{d}'(\mathbf{q})\dot{\mathbf{q}}.$$

Here $\mathbf{d}'(\mathbf{q})$ denotes the gradient of vector field $\mathbf{d}(\mathbf{q})$ (that is $\mathbf{d}'(\mathbf{q}) = \nabla_{\mathbf{q}}\mathbf{d}(\mathbf{q})$) and $\hat{\mathbf{A}}'(\mathbf{q})$ denotes the gradient of the matrix field $\hat{\mathbf{A}}(\mathbf{q})$, that is $\hat{\mathbf{A}}'(\mathbf{q}) = \nabla_{\mathbf{q}}\hat{\mathbf{A}}(\mathbf{q})$ is a matrix of order 3 and its general element is given by $\partial\hat{a}_{ij}(\mathbf{q})/\partial q_k$ [11] (Lovelock and Rund, 1975).

Now let us reconsider the r.h.s. of the above equation. Observe that $\hat{\mathbf{A}}(\mathbf{q})\dot{\mathbf{w}} = -\Lambda\mathbf{D}(\mathbf{q})\mathbf{z}$ and let $\mathbf{f} = \hat{\mathbf{A}}'(\mathbf{q})\dot{\mathbf{q}}\mathbf{w} + \mathbf{d}'(\mathbf{q})\dot{\mathbf{q}}$. Using $\mathbf{f}$ it holds that

$$\frac{\mathrm{d}}{\mathrm{dt}}\mathbf{z} = -\Lambda\mathbf{D}(\mathbf{q})\mathbf{z} + \mathbf{f}. \tag{B.4}$$

[11] Here $\nabla_{\mathbf{q}}$ denoted the column vector $(\partial/\partial q_1, \ldots, \partial/\partial q_n)^T$.

($\mathbf{f}$ will be expressed as the function of $\mathbf{q}$ and $\mathbf{z}$.) Now, let us compute $\dot{V}(\mathbf{x})$:

$$\dot{V}(\mathbf{x}) = \frac{\mathrm{d}}{\mathrm{dt}}\mathbf{z}^T\mathbf{z} = -\Lambda\mathbf{z}^T\left(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})\right)\mathbf{z} + 2\mathbf{z}^T\mathbf{f}.$$

We should like to show that there exist constants $0 < k < K$, such that if Λ is large enough and if $k/\Lambda \leq ||\mathbf{z}|| \leq K\Lambda$ then $\dot{V}(\mathbf{x}) = \dot{V}(\mathbf{z},\mathbf{q}) < -\beta$, where β is a positive number (to be defined later) that may depend on Λ. Thus we would like to prove that in a ring R around zero it holds that

$$\Lambda\mathbf{z}^T\left(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})\right)\mathbf{z} \geq 2\mathbf{z}^T\mathbf{f}.$$

We now estimate the l.h.s. of the above equation from below and the r.h.s. from above.

Since $\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})$ is symmetric it holds that

$$\inf_{\mathbf{z}} \frac{\mathbf{z}^T\left(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})\right)\mathbf{z}}{||\mathbf{z}||^2} \geq \lambda_{\min}(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q}))$$

and, moreover,

$$\Lambda\mathbf{z}^T\left(\mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})\right)\mathbf{z} \geq \Lambda\lambda||\mathbf{z}||^2, \tag{B.5}$$

where λ is defined in Equation (16).

Now let us estimate $\mathbf{z}^T\mathbf{f}$ from above. First we expand $\mathbf{f}$ by substituting $\dot{\mathbf{q}}$ from Equation (12) and $\mathbf{w}$ from Equation (11):

$$\mathbf{f} = \hat{\mathbf{A}}'(\mathbf{q})(\mathbf{v}(\mathbf{q}) + \mathbf{z})\hat{\mathbf{A}}^{-1}(\mathbf{q})(\mathbf{z} - \mathbf{d}(\mathbf{q})) + \mathbf{d}'(\mathbf{q})(\mathbf{v}(\mathbf{q}) + \mathbf{z}).$$

After reordering the terms according to the "powers" of $\mathbf{z}$ we arrive at

$$\begin{aligned}\mathbf{f} = {} & \hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z} + \\ & -\hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{d}(\mathbf{q}) + \hat{\mathbf{A}}'(\mathbf{q})\mathbf{v}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z} + \mathbf{d}'(\mathbf{q})\mathbf{z} +\end{aligned}$$

$$-\hat{\mathbf{A}}'(\mathbf{q})\mathbf{v}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{d}(\mathbf{q}) + \mathbf{d}'(\mathbf{q})\mathbf{v}(\mathbf{q}).$$

Since $\mathbf{z}^T\mathbf{f} \leq |\mathbf{z}^T\mathbf{f}|$ we have that

$$\begin{aligned}\mathbf{z}^T\mathbf{f} \leq\ & |\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z}| + \\ & |\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{d}(\mathbf{q})| + |\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{v}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z}| + |\mathbf{z}^T\mathbf{d}'(\mathbf{q})\mathbf{z}| + \\ & |\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{v}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{d}(\mathbf{q})| + |\mathbf{z}^T\mathbf{d}'(\mathbf{q})\mathbf{v}(\mathbf{q})|.\end{aligned}$$

In what follows we estimate the individual addends of the above formula. Since $\hat{\mathbf{A}}(\mathbf{q}) = \mathbf{D}(\mathbf{q})\mathbf{A}(\mathbf{q})$ we have that $\hat{\mathbf{A}}'(\mathbf{q}) = \mathbf{D}'(\mathbf{q})\mathbf{A}(\mathbf{q}) + \mathbf{D}(\mathbf{q})\mathbf{A}'(\mathbf{q})$ and $\hat{\mathbf{A}}^{-1}(\mathbf{q}) = \mathbf{A}^{-1}(\mathbf{q})\mathbf{D}^{-1}(\mathbf{q})$. Using some properties of the induced matrix norms for the first term we get that

$$\begin{aligned}|\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z}| &\leq |\mathbf{z}^T\mathbf{D}'(\mathbf{q})\mathbf{A}(\mathbf{q})\mathbf{z}\mathbf{A}^{-1}(\mathbf{q})\mathbf{D}^{-1}(\mathbf{q})\mathbf{z}| + |\mathbf{z}^T\mathbf{D}(\mathbf{q})\mathbf{A}'(\mathbf{q})\mathbf{z}\mathbf{A}^{-1}(\mathbf{q})\mathbf{D}^{-1}(\mathbf{q})\mathbf{z}| \\ &\leq ||\mathbf{z}||^3\, ||\mathbf{A}^{-1}(\mathbf{q})||\, ||\mathbf{D}^{-1}(\mathbf{q})|| \Big(||\mathbf{D}'(\mathbf{q})||\, ||\mathbf{A}(\mathbf{q})|| + ||\mathbf{D}(\mathbf{q})||\, ||\mathbf{A}'(\mathbf{q})||\Big).\end{aligned}$$

Since $\mathbf{A}(\mathbf{q})\mathbf{A}^{-1}(\mathbf{q}) = \mathbf{E}$, then $||\mathbf{A}^{-1}(\mathbf{q})|| \leq 1/||\mathbf{A}(\mathbf{q})|| \leq 1/a$, where a is given by Equation (14). Similarly, $||\mathbf{D}^{-1}(\mathbf{q})|| \leq 1/d$, where d is given by Equation (15). Since $\mathbf{A}$, $\mathbf{D}$ and $\mathbf{A}'$ and $\mathbf{D}'$ are continuous and D is compact, then $A = \sup_{\mathbf{q}\in D} ||\mathbf{A}(\mathbf{q})||$, $A' = \sup_{\mathbf{q}\in D} ||\mathbf{A}'(\mathbf{q})||$, $D = \sup_{\mathbf{q}\in D} ||\mathbf{D}(\mathbf{q})||$, and $D' = \sup_{\mathbf{q}\in D} ||\mathbf{D}'(\mathbf{q})||$ are finite. Now let $\alpha = D'A + DA'$. Then we have that $|\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z}| \leq ||\mathbf{z}||^3\alpha/(a\, d)$. Arguing similarly for the rest of the addends we get the following inequalities:

$$\begin{aligned}|\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{z}\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{d}(\mathbf{q})| &\leq ||\mathbf{z}||^2 p\alpha/a, \\ |\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{v}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{z}| &\leq ||\mathbf{z}||^2 v\alpha/a, \\ |\mathbf{z}^T\mathbf{d}'(\mathbf{q})\mathbf{z}| &\leq ||\mathbf{z}||^2 p' \\ |\mathbf{z}^T\hat{\mathbf{A}}'(\mathbf{q})\mathbf{v}(\mathbf{q})\hat{\mathbf{A}}^{-1}(\mathbf{q})\mathbf{d}(\mathbf{q})| &\leq ||\mathbf{z}||p\, v\alpha/a, \text{and} \\ |\mathbf{z}^T\mathbf{d}'(\mathbf{q})\mathbf{v}(\mathbf{q})| &\leq ||\mathbf{z}||p'\, v\end{aligned}$$

where $p = \sup\{\, ||\mathbf{d}(\mathbf{q})|| \mid \mathbf{q} \in D\,\}$, $p' = \sup\{\, ||\mathbf{d}'(\mathbf{q})|| \mid \mathbf{q} \in D\,\}$, and $v = \sup\{\, ||\mathbf{v}(\mathbf{q})|| \mid \mathbf{q} \in D\,\}$. Here v is finite and for p we have the estimate $p \leq (1+D)(v+b)+r$, where $b = \sup\{\, ||\mathbf{b}(\mathbf{q})|| \mid \mathbf{q} \in D\,\}$, and $r = \sup\{\, ||\mathbf{r}(\mathbf{q})|| \mid \mathbf{q} \in D\,\}$ are finite values. Similarly, p' is finite provided that (in addition to the above continuity assumptions) $\mathbf{D}' = \mathbf{D}'(\mathbf{q})$, $\mathbf{v}' = \mathbf{v}'(\mathbf{q})$, $\mathbf{b}' = \mathbf{b}'(\mathbf{q})$ and $\mathbf{r}' = \mathbf{r}'(\mathbf{q})$ are continuous.

Thus

$$\mathbf{z}^T\mathbf{f} \leq ||\mathbf{z}||^3\alpha/(a\,d) + ||\mathbf{z}||^2\Big((p+v)\alpha/a + p'\Big) + ||\mathbf{z}||v\Big(p\alpha/a + p'\Big).$$

Now let A_1, A_2 and A_3 be the coefficients of the terms $||\mathbf{z}||^3$, $||\mathbf{z}||^2$ and $||\mathbf{z}||$ in the above equation divided by $\lambda/2$.

We require that $\Lambda||\mathbf{z}||^2 \geq A_1||\mathbf{z}||^3 + A_2||\mathbf{z}||^2 + A_3||\mathbf{z}|| + \beta$. This inequality holds if the following inequalities hold: $\Lambda||\mathbf{z}||^2/4 \geq A_1||\mathbf{z}||^3$, $\Lambda||\mathbf{z}||^2/4 \geq A_2||\mathbf{z}||^2$, $\Lambda||\mathbf{z}||^2/4 \geq A_3||\mathbf{z}||$, and $\Lambda||\mathbf{z}||^2/4 \geq \beta$. Now let us choose Λ such that $\Lambda \geq 4A_2$. The last two inequalities are satisfied if

$$||\mathbf{z}|| \geq \max\Big(\frac{4A_3}{\Lambda}, \sqrt{\frac{\beta}{\Lambda}}\Big).$$

β is chosen so that $\frac{4A_3}{\Lambda} \geq \sqrt{\frac{\beta}{\Lambda}}$; that is, let β smaller than or equal to $16A_3^2/\Lambda$. Finally, the first inequality is satisfied provided that

$$||\mathbf{z}|| \leq \frac{\Lambda}{4A_1}.$$

(If $A_1 = 0$ then any $\mathbf{z}$ will do.) Summing up, let Λ satisfy $\Lambda \geq 4\max(A_2, \sqrt{A_3A_1}, A_3/\epsilon)$ and let $\beta \leq 16A_3^2/\Lambda$. Then, according to Corollary 4 there exists a positive number, $T > 0$ such that if

$$||\mathbf{z}(0)|| < \frac{\Lambda}{4A_1}$$

then for all $t > T$ for which $\mathbf{z}(t)$ is defined there holds

$$||\mathbf{z}(t)|| < \epsilon$$

and this proves the theorem.

C Perturbation of a robot arm

In this section we prove that for a particular type of robot arm grasping or releasing heavy objects the perturbation is positive definite. For this let us consider a robot arm with three degrees of freedom (Anderson and Miller, III, 1992) (see Figure 1). Assume that the mass of the end-point of the arm changes. First we compute the resulting perturbation matrix and then answer the question of positivity.

The robot is similar to the three major axes (base, upper arm and forearm) of typical industrial robots. Let J, M_1, M_2, L_1, L_2 denote the rotational inertia of the base, the point mass between the upper arm and the forearm, the point mass at the end of arm (including payload), the length of the upper arm, and the length of the forearm, respectively. Then the dynamics of the robot arm is given as

$$\ddot{\mathbf{q}} = \mathbf{A}(\mathbf{q})\mathbf{u} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}),$$

where $\mathbf{q} = (q_1, q_2, q_3)$ is the vector of angular positions of the robot (q_1 is the angular position of the robot base axis, q_2 is the angular elevation of the upper arm above horizontal, q_3 is the angular elevation of the forearm above horizontal), $\mathbf{u} = (u_1, u_2, u_3)$ is the torque vector of actuators (u_1, u_2 and u_3 denote the torque of the base, the upper arm and the forearm actuators, respectively), $\mathbf{A}(\mathbf{q})$ is the inverse of the inertia matrix and $\mathbf{b}(\mathbf{q}, \dot{\mathbf{q}})$ represents the Coriolis and centripetal forces and the gravity loading. Here

$$\mathbf{A}(\mathbf{q}) = \begin{pmatrix} \frac{1}{J+(M_1+M_2)L_1^2\cos^2 q_2+M_2L_1L_2\cos q_2\cos q_3+M_2L_2^2\cos^2 q_3} & 0 & 0 \\ 0 & \frac{M_2L_2^2}{a} & -\frac{(M_1+M_2)L_1^2}{a} \\ 0 & -\frac{M_2L_1L_2\sin(q_2+q_3)}{a} & \frac{(M_1+M_2)L_1^2}{a} \end{pmatrix},$$

where $a = (M_1 + M_2)M_2L_1^2L_2^2 - (M_2L_1L_2\sin(q_2 + q_3))^2$. We are interested in replacing M_2 by some other load, say M. Let us denote $\mathbf{A}(\mathbf{q})$ when M_2 is replaced by M by $\mathbf{A}_M(\mathbf{q})$. Let $\mathbf{D}(\mathbf{q}) = \mathbf{A}_M(\mathbf{q})\mathbf{A}_{M_2}^{-1}(\mathbf{q})$. Then $\mathbf{D}(\mathbf{q})$ is equal to

$$\begin{pmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{pmatrix},$$

where

$$\begin{aligned}
d_{11} &= \frac{J + L_1^2M_1\cos^2 q_2 + L_1^2M_2\cos^2 q_2 + M_2L_1L_2\cos q_2\cos q_3 + M_2L_2^2\cos^2 q_3}{J + L_1^2M_1\cos^2 q_2 + L_1^2M\cos^2 q_2 + ML_1L_2\cos q_2\cos q_3 + ML_2^2\cos^2 q_3} \\
d_{22} &= b(-MM_1L_2 - MM_2L_2 + M_2M_1L_1\sin q + MM_2L_1\sin q) \\
d_{23} &= bM_1L_2(M_2 - M) \\
d_{32} &= -bM_1L_1(M_2 - M)\sin q \\
d_{33} &= b(MM_1L_1\sin q + MM_2L_1\sin q - M_2M_1L_2 - MM_2L_2),
\end{aligned}$$

where $q = q_2 + q_3$ and

$$b = \frac{M_1 + M_2\cos^2 q}{M(M_1 + M_2)(L_1\sin q - L_2)(M_1 + M\cos^2 q)}.$$

We should like to know whether $\mathbf{C}(\mathbf{q}) = \mathbf{D}^T(\mathbf{q}) + \mathbf{D}(\mathbf{q})$ is positive definite and also its minimal singular value. The singular values of $\mathbf{D}(\mathbf{q})$ are

$$\lambda_1 = \frac{J + L_1^2 M_1 \cos^2 q_2 + L_1^2 M_2 \cos^2 q_2 + M_2 L_1 L_2 \cos q_2 \cos q_3 + M_2 L_2^2 \cos^2 q_3}{J + L_1^2 M_1 \cos^2 q_2 + L_1^2 M \cos^2 q_2 + M L_1 L_2 \cos q_2 \cos q_3 + M L_2^2 \cos^2 q_3},$$
$$\lambda_2 = \frac{M_1^2 + M_1 M_2 + M_2(M_1 + M_2)\cos^2 q}{M_1^2 + M_1 M_2 + M(M_1 + M_2)\cos^2 q}, \text{ and}$$
$$\lambda_3 = \frac{M_2}{M} \frac{M_1^2 + M_1 M + M_2(M_1 + M)\cos^2 q}{M_1^2 + M_1 M_2 + M(M_1 + M_2)\cos^2 q}.$$

$\lambda_2 > \lambda_3$ is equivalent to $M > M_2$. (These formulae were computed by Maple V©.) Using the well known arithmetical-geometrical mean inequality one can show that both the numerator and the denominator of λ_1 are positive. We know that $\lambda_{\min}(\mathbf{C}(\mathbf{q})) = \inf_{||\mathbf{z}||_2=1} \mathbf{z}^T \mathbf{C}(\mathbf{q})\mathbf{z}$. Hence,

$$\begin{aligned}
\inf_{||\mathbf{z}||_2=1} \mathbf{z}^T \mathbf{C}(\mathbf{q})\mathbf{z} &= \inf_{||\mathbf{z}||_2=1} \mathbf{z}^T (\mathbf{D}(\mathbf{q}) + \mathbf{D}^T(\mathbf{q}))\mathbf{z} \\
&\geq \inf_{||\mathbf{z}||_2=1} \mathbf{z}^T \mathbf{D}(\mathbf{q})\mathbf{z} + \inf_{||\mathbf{z}||_2=1} \mathbf{z}^T \mathbf{D}^T(\mathbf{q})\mathbf{z} \\
&= 2 \inf_{||\mathbf{z}||_2=1} \mathbf{z}^T \mathbf{D}(\mathbf{q})\mathbf{z}.
\end{aligned}$$

We know that $\inf_{||\mathbf{z}||_2=1} \mathbf{z}^T \mathbf{D}(\mathbf{q})\mathbf{z} = \lambda_{\min}(\mathbf{D}(\mathbf{q}))$ as long as $\mathbf{D}(\mathbf{q})$ is diagonizable. We claim that $\mathbf{D}(\mathbf{q})$ is diagonizable for all $\mathbf{q}$. Fix an arbitrary $\mathbf{q}$ and let $\mathbf{D} = \mathbf{D}(\mathbf{q})$. If all the eigenvalues of $\mathbf{D}$ were disjoint then of course $\mathbf{D}$ would be diagonizable. We know that $\mathbf{D}$ has at least two distinct eigenvalues: λ_2 and λ_3. (Otherwise $M = M_2$ in which case $\mathbf{D}$ is the unit matrix and is thus diagonizable). Assume further that, for example, $\lambda_1 = \lambda_2$. We claim that the eigenspace of $a = \lambda_1 = \lambda_2$ is two dimensional. This suffices for $\mathbf{D}$ to be diagonizable. Indeed, a must be the root of the characteristic polynomial (let us denote it by $p(x)$) of the matrix

$$\mathbf{D}_0 = \begin{pmatrix} d_{22} & d_{23} \\ d_{32} & d_{33} \end{pmatrix},$$

since the multiplicity of a is two. Moreover, a can be only a single-multiplicity root of this characteristic polynomial, since λ_2 and λ_3 are different. Now let us take an arbitrary eigenvector of $\mathbf{D}$ corresponding to a. Let us denote it by $\mathbf{x} = (x_1, x_2, x_3)^T$. It must hold that $(\mathbf{D} - a\mathbf{E})\mathbf{x} = 0$. Substituting $\mathbf{D}$ must hold that $0x_1 = 0$ and

$$\begin{pmatrix} d_{22} - a & d_{23} \\ \\ d_{32} & d_{33} - a \end{pmatrix} \begin{pmatrix} x_2 \\ \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ \\ 0 \end{pmatrix}.$$

We know that the rank of the matrix on the l.h.s. of the equation is exactly one since a is a single multiplicity root of $p(x)$, the characteristic polynomial of $\mathbf{D}_0$. Thus we have that x_1 can be chosen arbitrarily and that another component, say x_2 can be chosen arbitrarily, too. Thus the eigenspace of a is indeed 2-dimensional and since the eigenvectors of disjoint eigenvalues are independent of each other we get that $\mathbf{D}$ is always diagonizable.

Returning to the robot arm, we get that

$$\lambda_{\min}(\mathbf{C}(\mathbf{q})) \geq 2\min(\lambda_1, \lambda_2, \lambda_3) > 0.$$

Let $\lambda = \inf_{\mathbf{q}} \lambda_{\min}(\mathbf{C}(\mathbf{q}))$. Since λ_1, λ_2 and λ_3 are continuous functions of $\mathbf{q}$, and $\mathbf{q}$ is constrained in a compact space (the angle space) we get that there exists a point $\mathbf{q}$ such that $\lambda_{\min}(\mathbf{C}(\mathbf{q})) = \lambda$. From this one has that $\lambda > 0$. If we know consider the dependence of the eigenvalues on M one can see that all the eigenvalues are inversely proportional to M. Since Λ, the gain of the SDS Feedback Control, is also inversely proportional to λ we have that Λ should be chosen proportional to M.

Acknowledgement

We are grateful to András Krámli for his invaluable comments and suggestions.

D Figure Captions

- **Figure 1** *Compensatory control by doubling the inverse dynamics controller.* The SDS Controller is composed of two identical copies of an inverse dynamics controller (IDC). One copy acts as the original feedforward controller while the other identical copy is used to develop the compensatory signal, i.e., it is used in a feedback mode. The feedforward and the feedback controllers utilize the planned and the experienced speeds, respectively, to develop the control signal.
- **Figure 1** *Idealized 3-joint robotic manipulator.* The dynamics of this 3-joint robotic manipulator is highly non-linear. A typical perturbation is when the manipulator grasps (or releases) an object. This is modelled by changing the mass (M_2) at the end-effector. Compensation of this perturbation is hard, especially when the mass of the object is large compared to the mass of the manipulator.
- **Figure 3** *Architecture of the PDA neurocontroller.* The discretizing neurons have spatially tuned filters that input the sensory information. The neighboring (or geometrical) connections connect discretizing neurons that represent neighboring discretization points. Neighboring connections are utilized for spreading activation. Interneurons measure the sustained spreading activities and perform associative learning with the control neurons. Interneurons which reside at discretizing neurons corresponding to the actual state of the plant activate their connections to control neurons which sum up the incoming activities and output the result.
- **Figure 4** *Numerical studies of the FFC-FBC system.* Typical trajectories in the presence of additive perturbation (upper left figure), multiplicative perturbation (lower left figure), inhomogeneous perturbation (lower right figure), and features of the compensatory vector for the case of additive perturbation (upper right figure) are shown. The trajectory figures show 20x20 pixel regions. The task during the additive and multiplicative per-

turbations was to travel from west to east. The task for the inhomogeneous transformation was to travel from south-west to north-east. Uncompensated plants show the combined effects of finite-sized samples, finite resolution and disturbed activity patterns on the discretizing layer.

· Additive perturbation: the plant without compentsation leaves the state space while the plant that uses the compensation mechanism can build up the correct compensation term after a few time steps. The length of the compensation vector and the deviation angle from the optimal direction are shown in the upper right figure.

· Multiplicative perturbation: the plant without compensation proceeds along an angle to the correct direction. The build up time of the compensation vector now is faster due to a larger gain value.

· Inhomogeneous perturbation: the perturbation (again a rotation effect) was the strongest in the middle of the state space and was zero at the edges of the state space. The trajectory plotted by diamonds corresponds to the unperturbed plant that used the compensation mechanism. The trajectory plotted by squares corresponds to the perturbed plant that used the compensation mechanism. Due to the inhomogeneous nature of the perturbation and to the large integration time, the controller strongly overcompensates: the plant moves from the upper side of the optimal path to the lower side when the perturbation decreases rapidly from a large value to zero. This error can be made arbitrarily small under suitable conditions.

References

Albus, J. (1971). A theory of cerebellar function. *Mathematical Biosciences*, 10:25–61.

Anderson, C. and Miller, III, W. (1992). Challenging control problems. In *Neural Networks for Control*, Neural Network Modeling and Connectionism, pages 475–510. MIT Press, Cambridge.

Ben-Israel, A. and Greville, T. (1974). *Generalized Inverses: Theory and Applications*. Pure and Applied Mathematics, Wiley-Interscience. J. Wiley & Sons, New York.

Connolly, C. and Grupen, R. (1993). On the application of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946.

Dean, T. and Wellman, M. (1991). *Planning and Control*. Morgan Kaufmann, San Mateo, CA, USA.

Fomin, T., Szepesvári, Cs., and Lőrincz, A. (1994). Self-organizing neurocontrol. In *Proc. of IEEE WCCI ICNN'94*, volume 5, pages 2777–2780, Orlando, Florida. IEEE Inc.

Glasius, R., Komoda, A., and Gielen, S. (1995). Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 8(1):125–133.

Goodwin, G. and Sin, K. (1984). *Adaptive filtering, prediction and control*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Grossberg, S. and Kuperstein, M. (1986). *Neural Dynamics of Adaptive Sensory-motor Control: Ballistic Eye Movements*. Elsevier, Amsterdam.

Hwang, Y. and Ahuja, N. (1992). Gross motion planning – a survey. *ACM Computing Surveys*, 24(3):219–291.

Isidori, A. (1989). *Nonlinear Control Systems*. Springer-Verlag, Berlin.

Jordan, M. (1990). Learning and the degrees of freedom problem. In *Attention and Performance, XIII.*, Hillsdale. NJ: Erlbaum.

Kawato, M., Furukawa, K., and Suzuki, R. (1987). A hierarchical neural-network model for control and learning of voluntary movements. *Biological Cybernetics*, 57:169–185.

Keymeulen, D. and Decuyper, J. (1992). On the self-organizing properties of topological maps. In *Toward a Practice of Autonomous Systems, Proc. of the First European Conf. on Artificial Life*, pages 64–69. MIT Press.

La Salle, J. and Lefschetz, S. (1961). *Stability by Liapunov's Direct Method* *with Applications*. Mathematics in Science and Engineering. Academic Press, New York.

Landau, Y. (1979). *Adaptive Control: the Model Reference Approach*. Marcel Dekker, New York.

Lei, G. (1990). A neural model with fluid properties for solving labyrinthian puzzle. *Biological Cybernetics*, 64(1):61–67.

Lewis, F., Abdallah, C., and Dawson, D. (1993). *Control of Robot Manipulators*. MacMillan, New York.

Lewis, F., Liu, K., and Yesildirek, A. (1995). Neural net robot controller with guaranteed tracking performance. *IEEE Trans. on Neural Networks*, 6(3):703–715.

Locano-Pèrez, T. and Wesley, M. (1979). An algorithm for planning collision-free paths among polyhedral objects. *Communications of ACM*, 22(10):560–570.

Lovelock, D. and Rund, H. (1975). *Tensors, differential forms, and variational principles*. Pure and Applied Mathematics. A Wiley-Interscience Series of Texts, Monographs, and Tracts. Wiley-Interscience, New York.

Marr, D. (1969). A theory of cerebellar cortex. *Journal of Physiology (London)*, 202:437–470. Reprinted in Vaina, L., ed. (1991) *From Retina to the Neocortex: Selected Papers of David Marr*, Boston, Birkhauser pp. 129-203.

Miller, W. (1987). Sensor based control of robotic manipulators using a general learning algorithm. *IEEE Journal of Robotics and Automation*, 3:157–165.

Miller, W. I., Sutton, R., and Werbos, P., editors (1990). *Neural Networks for Control*. MIT Press, Cambridge, Massachusetts.

Minsky, M. (1961). Steps towards artificial intelligence. In *Proc. of the Institute of Radio Engineers*, pages 8–30. Reprinted in *Computers and Thought*, E.A. Feigenbaum and J. Feldman, editors, 406–450, McGraw-Hill, New York, 1963.

Miyamoto, H., Kawato, M., Setoyama, T., and Suzuki, R. (1988). Feedback-error-learning neural network for trajectory control of a robotic manipulator. *Neural Networks*, 1:251–265.

Morasso, P., Sanguineti, V., and Tsuji, T. (1993). Neural network architecture for robot planning. In *Proc. of ICANN'93*, pages 256–261, Amsterdam, The Netherlands. Springer-Verlag, London.

Narendra, K. and Monopoli, R. (1980). *Applications of adaptive control*. Academic Press, New York.

Narendra, K. and Parthasarathy, K. (1990). Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1(1):4–27.

Ortega, R. and Yu, T. (1987). Theoretical results on robustness of direct adaptive controllers: A survey. In *Proc. 10th IFAC World Congress*, pages 26–31, Munich.

Psaltis, D., Sideris, A., and Yamamura, A. (1988). A multilayered neural network controller. *IEEE Control Systems Magazine*, 8:17–21.

Rumiantsev, V. (1957). On the stability of a motion in a part of variables. *Univ. Ser. I Mat. Meh.*, 4:9–16. (in Russian).

Slotine, J.-J. and Li, W. (1991). *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ, USA.

Szepesvári, Cs., Balázs, L., and Lőrincz, A. (1994). Topology learning solved by extended objects: a neural network model. *Neural Computation*, 6(3):441–458.

Szepesvári, Cs. and Lőrincz, A. (1995). Integrated architecture for motion control and path planning. *Journal of Robotic Systems*. submitted.

Szepesvári, Cs. and Lőrincz, A. (1996a). Approximate geometry representation and sensory fusion. *Neurocomputing*, 12(2–3):267–287.

Szepesvári, Cs. and Lőrincz, A. (1996b). High precision neurocontrol of a chaotic bioreactor. In *2nd World Congress of Nonlinear Analysts*. (invited talk).

Szepesvári, Cs. and Lőrincz, A. (1996c). Neurocontrol I: Self-organizing speed-field tracking. *Neural Network World*, pages 875–896.

Szepesvári, Cs. and Lőrincz, A. (1996d). Neurocontrol II: High precision control achieved using approximate inverse dynamics models. *Neural Network World*, pages 897–920.

Tarassenko, L. and Blake, A. (1991). Analogue computation of collision-free paths. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pages 500–505. IEEE.

Ungar, L. (1992). A bioreactor benchmark for adaptive network-based process control. In *Neural Networks for Control*, Neural Network Modeling and Connectionism, pages 387–402. MIT Press, Cambridge.

Vemuri, V. (1993). Artificial neural networks in control applications. *Advances in Computers*, 36:203–254.

Werbos, P. (1988). Generalization of back propagation with applications to a recurrent gas market model. *Neural Networks*, 1:339–356.

Widrow, B. (1986). Adaptive inverse control. In *Proc. of the Second IFAC Workshop on Adaptive Systems in Control and Signal Processing*, pages 1–5, Lund, Sweden. Lund Institute of Technology.

Widrow, B., McCool, J., and Medoff, B. (1978). Adaptive control by inverse modeling. In *20th Asilomar Conference on Circuits, Systems and Computers.*

Young, L. (1969). On adaptive manual controls. *IEEE Trans. Man-Machine Systems*, 10:292–231.

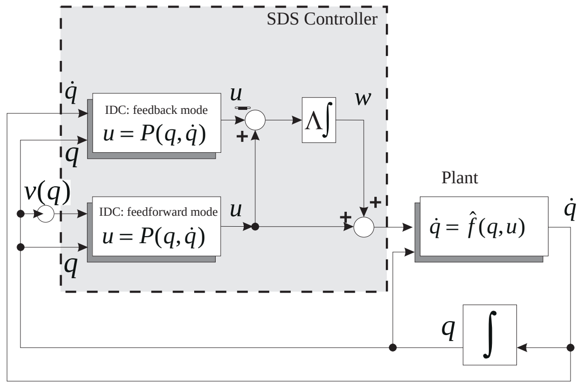

Fig. 1. Compensatory control by doubling the inverse dynamics controller.

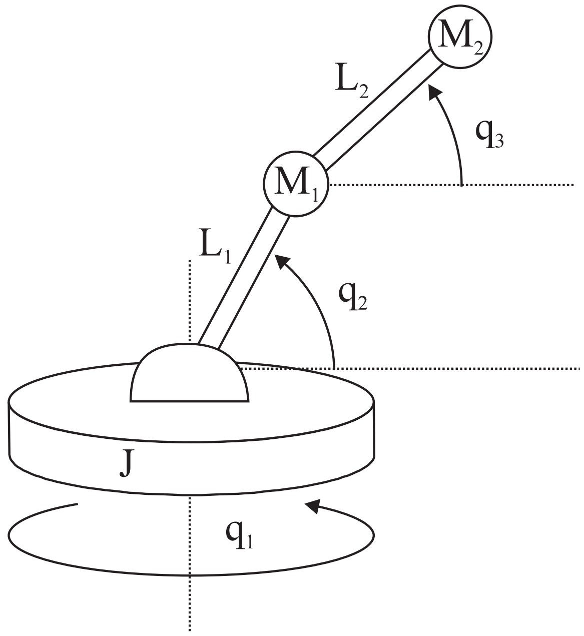

Fig. 2. Idealized 3-joint robotic manipulator.

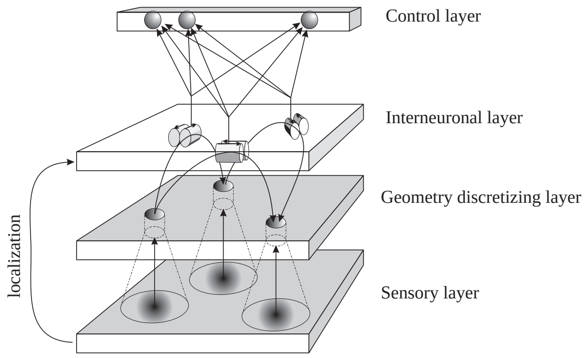

Fig. 3. Architecture of the PDA neurocontroller.

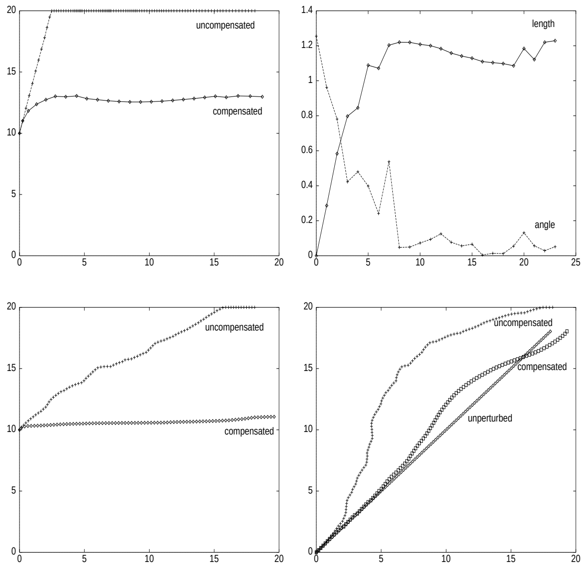

Fig. 4. Numerical studies of the FFC-FBC system.