

# Neurocontrol II: High Precision Control Achieved Using Approximate Inverse Dynamics Models

Csaba Szepesvári<sup>†‡</sup>, and András Lőrincz<sup>†</sup>  
{szepes,lorincz}@iserv.iki.kfki.hu

<sup>†</sup> Department of Photophysics  
Institute of Isotopes of the  
Hungarian Academy of Sciences  
Budapest, P.O. Box 77, Hungary, H-1525

<sup>‡</sup> Bolyai Institute of Mathematics  
University of Szeged  
Szeged, Hungary, H-6720

August 15, 1996

## Abstract

It is common that artificial neural networks (ANNs) are used for approximating the inverse dynamics of a plant. In the accompanying paper a self-organising ANN model for associative identification of the inverse dynamics was introduced. Here we propose the use of approximate inverse dynamic models for both Static and Dynamic State (SDS) feedback control. This compound controller is capable of high-precision control even when the inverse dynamics is just qualitatively modeled or the plant's dynamics is perturbed. Properties of the SDS Feedback Controller in learning the inverse dynamics as well as comparisons with other methods are discussed. An example is presented when a chaotic plant, a bioreactor, is controlled using the SDS Controller. We found that the SDS Controller can compensate model mismatches that otherwise would lead to an intolerably large error if a traditional controller were used.

## Contents

|          |                                                             |           |
|----------|-------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                                         | <b>4</b>  |
| <b>2</b> | <b>Preliminaries</b>                                        | <b>5</b>  |
| 2.1      | Terms                                                       | 5         |
| 2.2      | Forward, inverse dynamics and the speed field tracking task | 6         |
| <b>3</b> | <b>Theory of SDS Feedback Controllers</b>                   | <b>8</b>  |
| 3.1      | The perturbed equations                                     | 8         |
| 3.2      | Compensatory Dynamic State Feedback Control                 | 8         |
| 3.3      | Uniform ultimate boundedness                                | 10        |
| 3.4      | The ultimate boundedness of the feedback error              | 12        |
| 3.5      | Discussion of the theory                                    | 13        |
| 3.6      | Compensatory control using neurocontrollers                 | 14        |
| <b>4</b> | <b>Computer simulations</b>                                 | <b>18</b> |
| 4.1      | Inverse dynamic approximated by a neural network            | 18        |
| 4.2      | Controlling a chaotic plant                                 | 20        |
| <b>5</b> | <b>Discussion of SDS Feedback</b>                           | <b>22</b> |
| 5.1      | Conventional vs SDS Feedback Control                        | 22        |
| 5.2      | Nonstationary perturbations and noise sensitivity           | 22        |
| 5.3      | Open questions                                              | 24        |
| <b>6</b> | <b>Conclusions</b>                                          | <b>25</b> |
| <b>7</b> | <b>Acknowledgments</b>                                      | <b>26</b> |
| <b>8</b> | <b>Figure captions</b>                                      | <b>26</b> |

## 1 Introduction

A vast amount of work has dealt with neural networks for controlling a plant with known, partially known, or unknown dynamics. 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 of a manipulator 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 state space of the plant and extend the sensory set (e.g., measure 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, Kawato, Setoyama, and Suzuki 1988; Lewis, Liu, and Yesildirek 1995. This latter option has several advantages but may also be disadvantageous.

The advantage of the extra Feedback Controller (FBC) is that it allows the Feedforward Controller (FFC) to work with a broader range of problems since the FBC can compensate for errors. The other side of the coin is that error compensation can not remain perfect for non-linear control problems for a finite period of time in the general case. In order to appreciate this, one should assume that ideal compensation is achievable. Clearly, this results in the loss of the input of the FBC and this means that the compensation signal remains unchanged. However, as the plant moves the "ideal" compensation signal changes and thus the compensation signal of the controller becomes imperfect. Also, 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.

In Szepesvári and Lőrincz 1995 we proposed to use the same inverse dynamics controller (a controller that has learnt and approximated the inverse dynamics of the plant) for both FFC and FBC. The FFC is extended by an external, compensatory control signal for compensating perturbations of the plant's dynamics. 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. This latter control signal is computed by the FBC.

The said proposal 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.

## 2 Preliminaries

### 2.1 Terms

The terminology will be defined first since the 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<sup>1</sup> 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

<sup>1</sup>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.

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.

### 2.2 Forward, inverse dynamics and the speed field tracking task

Let  $\mathbf{R}^{m \times n}$  denote real  $m \times n$  matrices. We say that a matrix  $\mathbf{A}$  admits a generalized inverse<sup>2</sup> if there is a matrix  $\mathbf{X}$  for which  $\mathbf{AXA} = \mathbf{A}$  holds. 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  $\mathbf{x} = \mathbf{Xb} + (\mathbf{E} - \mathbf{XA})\mathbf{y}$ , where  $\mathbf{E}$  is the unit matrix, provided that the considered linear equation has a solution at all Ben-Israel and Greville 1974. Here  $\mathbf{y}$  denoted an arbitrary vector of the appropriate dimensions. 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:

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

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  $\mathcal{D}$ ) of the state variable  $\mathbf{q}$  is compact and is simply connected; that  $n \leq m$ , and for each  $\mathbf{q} \in \mathcal{D}$  the rank of matrix  $\mathbf{A}(\mathbf{q})$  is equal to  $n$ ; that is, the matrix is nonsingular. As a consequence the plant is

<sup>2</sup>Sometimes it is called the pseudo-inverse, or simply the inverse of matrix  $\mathbf{A}$ .

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  $A^{-1}$  is non-unique.

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

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

$$v(q) = b(q) + A(q)u(q). \quad (2)$$

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. Further discussion of speed field tracking can be found in Szepesvári and Lőrincz 1996.

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

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

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

$$u(q) = p(q, v(q))$$

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  $y(q, t) = 0$  and thus

$$p(q, v) = A^{-1}(q)(\dot{q} - b(q)). \quad (4)$$

This assumption simplifies the calculations and is justified later.

### 3 Theory of SDS Feedback Controllers

#### 3.1 The perturbed equations

Assume that the equation of motion of the plant changes and the new equation system reads as follows:

$$\dot{q} = \hat{b}(q) + \hat{A}(q)u, \quad (5)$$

where  $\hat{A}(q)$  is a nonsingular matrix field. Let us first assume that we seek a static state feedback compensatory control signal  $w = w(q)$ , such that the control signal  $u(q) + w(q)$  solves the original speed field tracking problem for the perturbed plant. One can check that for the compensatory control signal<sup>3</sup>

$$w(q) = \hat{A}^{-1}(q)(v(q) - \hat{v}(q)) \quad (6)$$

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

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

Indeed,

$$\begin{aligned} \dot{q} &= \hat{b}(q) + \hat{A}(q)(u(q) + w(q)) \\ &= \hat{v}(q) + v(q) - \hat{v}(q) \\ &= v(q). \end{aligned}$$

Unfortunately, the learning of  $w(q)$  is as complex as it is to estimate  $\hat{A}^{-1}(q)$  and  $\hat{b}(q)$  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.

#### 3.2 Compensatory Dynamic State Feedback Control

First, observe that  $w(q)$  satisfies the equality

$$u(q) = p(q, v(q, w(q))), \quad (7)$$

<sup>3</sup>Note that the compensatory control signal  $w(q) + (E - A^{-1}(q)A(q))y(q, t)$ , where  $y = y(q, t)$  is arbitrary, results in  $\dot{q} = v(q)$ , too. Thus  $w(q)$  can be viewed as the mean or main part of perfect compensatory signals

where

$$\mathbf{v}(\mathbf{q}, \mathbf{w}) = \dot{\mathbf{q}} = \mathbf{b}(\mathbf{q}) + \mathbf{A}(\mathbf{q})(\mathbf{u}(\mathbf{q}) + \mathbf{w}). \quad (8)$$

Indeed, if both sides of Equation (7) are multiplied by  $\mathbf{A}(\mathbf{q})$  and  $\mathbf{b}(\mathbf{q})$  is added we arrive at the equation

$$\mathbf{v}(\mathbf{q}) = \mathbf{v}(\mathbf{q}, \mathbf{w}(\mathbf{q})) \quad (9)$$

which is equivalent to the desired  $\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q})$ . The simplest error-feedback law is to let  $\mathbf{w}$  change until Equation (7) is satisfied. Then we get the following equations:

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

where  $\Lambda$  is a fixed positive number.<sup>4</sup> 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 Static and Dynamic State 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 the same controller computes the differential feedback control signal by using the actual speed ( $\dot{\mathbf{q}}$ ) of the plant. Now the feedforward control signal is subtracted from the feedback control signal, the result is integrated through time, and added to the feedforward control signal. The sum is used as the input to the plant.

<sup>4</sup>Equivalently one might consider the feedback equation

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

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 neighbourhood 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. Below it will be indicated that under some well defined conditions  $\mathbf{w}$  can be kept as close to the ideal control signal as desired by choosing large enough  $\Lambda$ .

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

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

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

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

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.

### 3.3 Uniform ultimate boundedness

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

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad (13)$$

where  $\mathbf{x}$  is the element of  $D$ ,  $D$  is a compact subset of  $\mathbf{R}^n$ , and  $\mathbf{f}$  is a vector valued smooth function over  $D$ . The solution of Equation (13) corresponding to the initial condition  $\mathbf{x}(0) = \xi$ , is denoted by  $\varphi(t; \xi)$  ( $\xi \in D$ ). It is assumed that the output of Equation (13) is

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

where  $\mathbf{y} \in \mathbf{R}^t$ , and  $h$  is continuous. Let  $\|\cdot\|$  denote an arbitrary norm over  $\mathbf{R}^t$  and let  $U$  be an arbitrary subset of  $\mathbf{R}^t$ . 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  $\xi$  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  $\xi$  and  $t$  then for all  $t' > t + T$  it holds that  $\|h(\varphi(t'; \xi))\| < b$ .  $T$  will be called the absorption time.

Let us assume that system (13) is decomposed into two parts:

$$\begin{aligned}\dot{\mathbf{x}}_1 &= f_1(\mathbf{x}) \\ \dot{\mathbf{x}}_2 &= f_2(\mathbf{x}),\end{aligned}\tag{14}$$

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 (14) is  $h(\mathbf{x}) = \mathbf{x}_1$ . The following theorem is proven in Szepesvári and Lőrincz 1995

**THEOREM 3.1** *Let us consider the autonomous differential equation given by (14). Assume that the domain of this equation contains a neighbourhood of zero. Let  $\|\cdot\|$  denote an arbitrary norm on  $\mathbf{R}^n$  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,  $\beta$ . 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:

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

Then the output  $\mathbf{y} = \mathbf{x}_1$  of system (14) is UUB w.r.t. the set  $\{\mathbf{x} \in D \mid \|\mathbf{x}_1\| \leq K\}$  and bound  $b$ .

This theorem states that under the required conditions Equation (14) is partially uniformly bounded. Some partial stability concepts were considered by Rumiantsev Rumiantsev 1957

### 3.4 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 is 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

$$\hat{\mathbf{A}}(\mathbf{q}) = \mathbf{D}(\mathbf{q})\mathbf{A}(\mathbf{q})\tag{15}$$

and let

$$\mathbf{r}(\mathbf{q}) = \hat{\mathbf{b}}(\mathbf{q}) - \mathbf{b}(\mathbf{q}).\tag{16}$$

Further, let

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

Elementary calculations show that

$$\dot{\mathbf{d}}(\mathbf{q}) = (\mathbf{E} - \mathbf{D}(\mathbf{q}))(\mathbf{v}(\mathbf{q}) - \mathbf{b}(\mathbf{q})) + \mathbf{D}(\mathbf{q})\mathbf{r}(\mathbf{q}).\tag{17}$$

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

**THEOREM 3.2** *Assume that the perturbation of  $\mathbf{A}(\mathbf{q})$  is given by Equation (15) and the perturbation of  $\mathbf{b}(\mathbf{q})$  is given by Equation (16). Suppose that  $\mathbf{A}(\mathbf{q})$ ,  $\mathbf{b}(\mathbf{q})$ ,  $\mathbf{v}(\mathbf{q})$  and  $\mathbf{D}(\mathbf{q})$ ,  $\mathbf{r}(\mathbf{q})$  have continuous derivatives and that the following constants are positive:*

$$a = \inf\{\|\mathbf{A}(\mathbf{q})\| \mid \mathbf{z} \in D\}\tag{18}$$

$$d = \inf\{\|\mathbf{D}(\mathbf{q})\| \mid \mathbf{z} \in D\}\tag{19}$$

$$\lambda = \inf\{\lambda_{\min}(\mathbf{D}(\mathbf{q}) + \mathbf{D}^T(\mathbf{q})) \mid \mathbf{q} \in D\}.\tag{20}$$

Then for all  $\epsilon > 0$  there exists a gain  $\Lambda$  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  $t$ . Here  $K$  is a fixed positive constant and  $\mathbf{z}(0)$  denotes the initial value of  $\mathbf{z}$ . Further,  $\Lambda \sim O(1/\epsilon)$ .

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 *non-invertive or uniformly positive definite* if  $\lambda$  (defined by Equation (20)) is positive.

The proof, which is based on the Liapunov function  $V(\mathbf{x}) = \mathbf{z}^T \mathbf{z}$ , is given in Szepesvári and Lőrincz 1995.

### 3.5 Discussion of the theory

In this section we consider some special cases of the above theorem for various types of systems. We start with the most simple case:

It is easy to show that the undesirable non-quadratic terms of the derivative of the Liapunov function ( $\dot{V}$ ) disappear if the plant's equation is given by

$$\dot{\mathbf{q}} = \mathbf{A}\mathbf{u} + \mathbf{b}, \quad (21)$$

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

**PROPOSITION 3.3** *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

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

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

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 three dimensional space with 3 degrees of freedom (see Fig. 2). 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 are interested in 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 nonlinear. One can prove that the SP-matrix is positive definite and even that it is uniformly positive definite. The calculations are given in Szepesvári and Lőrincz 1995.

### 3.6 Compensatory control using neurocontrollers

In this section we treat the inverse dynamics neurocontrollers within the general control scheme discussed above. The use of neurocontrollers – because they represent a large 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 bases 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*.

First we consider the effect of simultaneously training and using a neurocontroller such as the dynamic state feedback controller. Thus, in this case the same learning neurocontroller is used to control the plant. First we discuss that *the precision of tracking may be increased* if one uses a neurocontroller, which represents the inverse dynamics of the plant just *approximately*, for *SDS control*. 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  $A^{-1}(q)$  by  $P(q)$  and  $b(q)$  by  $s(q)$ . Then, of course,  $A(q)$  "approximates"  $P^{-1}(q)$ . Now let us imagine that the inverse dynamics of our controller is "exact", i.e., the plant's equation is given by

$$\dot{q} = P^{-1}(q) + s(q) \quad (22)$$

Now, (1) is thought of as the perturbed system and (22) as the unperturbed system. If we apply Theorem 3.2 we get that under some smoothness conditions and provided that  $\inf_q \lambda_{\min}(D^T(q) + D(q)) > 0$ , where  $D(q) = A(q)P(q)$ , 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  $P$  approximates  $A^{-1}$  sufficiently closely. If  $\lambda = \inf_q \lambda_{\min}(D^T(q) + D(q)) > 0$ , we say that *the controller represents the inverse dynamics of the plant sign-properly*. If  $\lambda = 0$  then we say the controller represents the inverse dynamics of the plant *semi-sign properly*. 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 for SDS Control seems to be advantageous even during the learning phase. On the other hand, if the initial controller

does not represent the plant (semi-)sign properly then one should be cautious in using the controller for SDS Control since the boundedness of the compensatory signal cannot be ensured. 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, it is still possible that controllers being signproperly may render the system unstable during learning, depending on the learning rule applied. This question is considered in the rest of this section. First we consider direct methods Grossberg and Kuperstein 1986; Psaltis, Sideris, and Yamamura 1988; Widrow, McCool, and Medoff 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 then 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 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.<sup>5</sup> 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 compensates the structural approximation error only.

<sup>5</sup>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.

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 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 should one 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 be taking 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, Abdallah, and Dawson 1993; Miyamoto, Kawato, Setoyama, and Suzuki 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. The aim here is to *illustrate* the theory by simulations. The plant's equation, the neurocontroller and also the perturbations were kept as simple as possible in order to illustrate the way of working of the compensation mechanism. Despite this simplicity the model used is quite complex and goes beyond the limits of the theory. This is

because we consider simplified sensorimotor control, i.e., some aspect of sensory coding is included. The simulations support the self-improving nature of SDS Feedback.

## 4 Computer simulations

We present two sets of computer experiments. The first example illustrates when a neural network approximates the inverse dynamics, i.e., this approach involves the non-parametric approximation of the inverse dynamics, while in the second example it is assumed that the form of the inverse dynamics is known but some parameters are severely mismatched.

### 4.1 Inverse dynamic approximated by a neural network

In these experiments the so called PDA controller is used to model the inverse dynamics of a simple plant and the very same architecture is used to generate the speed field to be tracked. For details of this neural network see Szepesvári and Lőrincz 1996.

The spatial filters, the proximity relations as well as the control commands stored by the interneurons, although could be learnt Szepesvári and Lőrincz 1996, were *prewired in an ideal fashion* in order to exclude the additional effects of imperfect learning and thus *only structural approximation errors are present in the simulations*.

The plant's equation and the perturbed equation are given by

$$\dot{\mathbf{q}} = \mathbf{u} \quad (23)$$

$$\dot{\mathbf{q}} = \mathbf{F}'(\alpha(\mathbf{q}))\mathbf{u}, \quad (24)$$

where  $\mathbf{F}'(\cdot)$  is the rotation matrix introduced earlier and the rotation angle has the same position dependence as before:

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

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 not everywhere differentiable and that  $\lambda_{\min}(\mathbf{D}''(\mathbf{c}) + \mathbf{D}(\mathbf{c})) = 0$ ; that is, the symmetrized perturbation matrix is non-uniformly positive definite. This way the simulation goes beyond the present theory.

The task was changed to start from the lower left corner and 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  $20 \times 20$  discretization.) 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 Fig. 3. 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  $\Lambda$ . The gain increase, however, may make the feedback more sensitive to noise – a topic that will be discussed in Section 5.2.

The uncompensated plant not only deviates strongly from the desired path but also reveals the underlying discretization by the small humps of the motion. This structural approximation error is also compensated by the SDS Controller.

The presented numerical example shows 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 also the error detection and the development of the feedback control signal may take some time. If the feedback signal is fast and strong then the system has a very short sampling time and 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 is that it corrects structural approximation errors, too.

## 4.2 Controlling a chaotic plant

Chemical systems can be relatively simple in that they have few variables, but still troublesome to control due to strong nonlinearities which are difficult to model accurately. A prime example is the bioreactor. In its simplest form, a bioreactor is simply a tank containing water and cells (e.g., yeast or bacteria) which consume nutrients ("substrate") and produce products (both desired and undesired) and more cells. The simplest version of the bioreactor is a continuous flow stirred tank reactor (CSTR) in which cell growth depends only on the nutrient being fed to the system. The target values to be controlled are the cell mass yield and the nutrient concentration. A basic set of equations for such a bioreactor is:

$$\dot{c}_1 = -c_1(u + v) + c_1(1 - c_2)e^{\frac{c_2}{\gamma}} \quad (25)$$

$$\dot{c}_2 = -c_2(u + v) + Sv + c_1(1 - c_2)e^{\frac{c_2}{\gamma}} \frac{1 + \beta}{1 + \beta - c_2}, \quad (26)$$

where  $c_1$  and  $c_2$  are, respectively, dimensionless mass and substrate conversions Agrawal, Lee, Lim, and Ramkrishna 1982. The control parameter,  $u$ , is the flow rate through the reactor, while the control parameter,  $v$ , is the flow rate used to raise the substrate concentration. The constants  $\beta$  and  $\gamma$  determine the rate of cell growth and nutrient consumption, while  $S$  defines the dimensionless form of substrate concentration.

This problem has proved challenging for conventional controllers and was suggested as a control benchmark problem in Ungar 1992. The system is difficult to control for several reasons: the uncontrolled equations are highly nonlinear and exhibit limit cycles. Optimal behaviour occurs in near an unstable region. Note that for  $\beta = 0.02$ ,  $\gamma = 0.48$  and  $S = 1$  a Hopf bifurcation occurs at  $u = 0.829$ ,  $v = 0.0004$ .

Our experiments with this plant consisted of two phases corresponding to two sets of target values for  $c_1^*$  and  $c_2^*$ . First, the system was brought to a steady state at  $(c_1^*, c_2^*) = (0.0737, 0.8760)$  and then the target values were

changed to  $(c_1^*, c_2^*) = (0.1287, 0.8688)$  which corresponds to a stable fixed point of the reactor with flow rates  $(u^*, v^*) = (0.8012, 0.0004)$ . Then the target cell mass was increased again, now to  $(c_1^*, c_2^*) = (0.1737, 0.7978)$  with equilibrium control values  $(u^*, v^*) = (1.0652, 0.0004)$ . Control was made at intervals of 0.5 s, while simulating the bioreactor with Euler-methods and with  $dt = 0.01$  s. One phase lasted 50 seconds. The said change in set-point is sufficient to shift from a stable regime into the domain of attraction of a limit cycle (see Fig. 4). Even if the correct model is known, small errors in parameters give rise to very inaccurate control when a feedforward controller is used: an error of 2% in  $\gamma$  leads to 50% error in the target cell mass Brengel and Seider 1989.

We tested SDS Control using the inverse dynamics corresponding to Equation (25) with  $\hat{\gamma} = 1.1\gamma$ . Since the change of  $\gamma$  results in an additive perturbation of the inverse dynamics, thus SDS Control can be applied without any restrictions and the theory predicts that SDS Control will yield bounded tracking error. The speed field was given by  $v(c_1, c_2) = \lambda(c_1^* - c_1, c_2^* - c_2)$ , where  $\lambda$  was determined so that with ideal control  $e(t^*)/e(0) = 1/e$  for  $t^* = 10/3$  s, where  $e(t)$  is the error of either  $c_1$  or  $c_2$  at time  $t$ . The error of tracking is shown in Figure 5. The first of the three subfigures correspond to the case when the inverse dynamics model was perfect, while the second and third subfigures correspond to the cases when the inverse dynamics was imperfect with and without dynamic feedback, respectively. Note that there is almost no difference between the first and second subfigures, meaning that SDS Feedback could very efficiently compensate for the mismatched inverse dynamics model. However, when there was no feedback the error percentage of  $c_1$  was larger then 60%. This figure suggests that SDS Control is able to perfectly compensate the perturbation, although the perturbation is highly nonlinear. This also follows from the theory since the aim of the control was to bring the system to a steady state and thus Proposition 3.3 can be applied to the linearized plant's equation after the plant reaches a sufficiently small neighborhood of the designated equilibrium state. Figure 6 shows the control variables and the ideal control values in the three cases. This figure reinforces the impression that SDS Control is very efficient. Figure 7 shows the feedback signal in the unperturbed and the perturbed cases (corresponding to the l.h.s. and r.h.s. subfigures, respectively). Notice that feedback was non-zero even in the unperturbed case because of the delay of control. It can be seen that after a short period of time the feedback values settled

quickly. In these experiments A, the feedback gain, was 1.

## 5 Discussion of SDS Feedback

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 some open questions.

### 5.1 Conventional vs. SDS Feedback Control

The main difference between these two types of controls is that in designing a conventional feedback controller, a 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 has still the opportunity to design a dynamic state feedback controller in the conventional way. However, as it has been shown, one can simply use the learnt inverse dynamics controller for dynamic state feedback as well. Moreover, the resulting compound controller can compensate perturbations quickly under relatively mild conditions. Since the present FBC closely fits the controlled plant the feedback may well be more precise than a linear feedback controller would be. Moreover, simultaneous learning of the inverse dynamics model and the SDS Feedback is possible when the learning method is direct system identification (see section 3.6). The fully self-organizing approach presented in this paper seems most advantageous if the plant's dynamics is completely or partially unknown.

### 5.2 Nonstationary perturbations and noise sensitivity

In the proofs we assumed that the perturbation is stationary, i.e., if it is switched on then it remains unchanged forever. However, this is not a realistic assumption. As far as nonstationary perturbations are concerned the proof of Theorem 3.2 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  $\frac{d}{dt}z$  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 non-quadratic terms in the (f) in the derivative of the Liapunov function. This means that if the changes are slow, i.e., these terms are bounded, then for large enough  $\Lambda$  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 unpleasant features of noise are ruled out. In what follows we assume that the noise affecting the system has bounded amplitude and bandwidth. Another 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 the SDS Feedback mechanism with high enough  $\Lambda$  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 the present system. Note that this is not so for conventional linear feedback controllers.

The most delicate case 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

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

where  $\mathbf{n}(t)$  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  $\Lambda$ . This means that increasing  $\Lambda$  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.

### 5.3 Open questions

One shortcoming of our approach is that the perturbations should be non-invertive 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  $\Lambda$  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  $\Lambda$  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, Abdallah, and Dawson 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, Liu, and Yesildirek 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.

## 6 Conclusions

Complex control problems in structured environments were considered in the present work the task being to control a plant with previously unknown dynamics while avoiding obstacles and experiencing perturbations in the plants dynamics. The solution is based on the designation of an appropriate speed field, which when tracked ensures collision free motion. This approach is more robust than the earlier models for collision free motion, namely, trajectory tracking. A well known spreading activation neural model for fast speed field planning on the discretization of the state space was utilized. This model was augmented by interneurons and control neurons, with interneurons being connected to control neurons. The particular architecture allows to control plants having non-linear inverse dynamics. The resulting control signal is smooth. In the second part of the article it was 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. Under relatively mild conditions it was shown that an arbitrary non-linear perturbation can

be compensated by the present method provided that the perturbation is non-invertive. The compensatory signal can be built up very rapidly as it follows from the general theory for linear plants. Compensatory control for non-linear plants was demonstrated by computer simulations.

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 simultaneously if the training signal is always the sum of the feedforward and feedback control signals. Another advantage is that the SDS Feedback Control mode can reduce the approximation errors of the inverse dynamics. This relaxes the number of parameters that are required to achieve a given precision in control.

## 7 Acknowledgments

We are grateful to Prof. András Krámli for his invaluable comments and suggestions. This work was partially founded by OTKA grants T917110, T914330, T014566, and US-Hungarian Joint Fund Grant 168/91-A 519/95-A.

## 8 Figure captions

**Figure 1. Compensatory control by doubling the inverse dynamics controller** The proposed compound controller is capable of compensating homogeneous perturbations. The same inverse dynamics controller plays two roles: it is used to compute the control signal (i.e., it is used as a Feedforward Controller, FFC), as well as the compensatory signal (Feedback Controller, FBC).

**Figure 2. 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 modeled 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. Typical trajectories in the presence of inhomogeneous perturbations.** In these experiments the plant's equation changed inhomogeneously. The change (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 task was to travel from the lower left corner to the upper right corner of the state space within the  $20 \times 20$  pixel region. The figure shows three trajectories. The first (plotted by diamonds) is a line, almost straight and it corresponds to the unperturbed plant that used the compensation mechanism. The second (plotted by plus signs) shows the trajectory of the perturbed plant without compensation. The large deviation from the optimal trajectory is due to the perturbation while the small oscillation like disturbances are due to the discretization. The third 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 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.

**Figure 4. Behaviour of the bioreactor for different constant flow rates.** The figures show the evolution of the bioreactor state variables as described by Equation (25) with  $(u, v) = (0.8012, 0.0004)$  and  $(u, v) = (1.0652, 0.0004)$  for the l.h.s. and r.h.s. subfigures, respectively. In both cases the reactor was started from a 0.0001 neighborhood of an equilibrium state – in the first case the equilibrium state is stable, while in the second case it is unstable and the plant's state approaches a limit cycle. The vertical axis corresponds to the time variable. The left and the right axis represent the  $c_2$  and  $c_1$  values, respectively.

**Figure 5. Bioreactor control: Error of control vs. time.** The error percentage of control is given as  $e_i = 100(c_i - c_i^*)/c_i^*$ ,  $i = 1, 2$ , where  $(c_1^*, c_2^*)$  is the desired set point of the bioreactor. After 50 s a new set-point is designated that corresponds to an unstable equilibrium of the plant. When feedback is in effect the error quickly reduces to zero. Without feedback the error remains high even if the inverse dynamics

is moderately imprecise. For details of the experiment see the text.

**Figure 6. Bioreactor control: Control values vs. time.**  $(u^*, v^*)$  are the ideal equilibrium control values, while  $(u, v)$  are the actual control values. After 50 s a new set-point is designated that corresponds to an unstable equilibrium of the plant. Thus the ideal control values can not be assumed from the beginning. For details of the experiment see the text.

**Figure 7. Bioreactor control: Feedback values vs. time.** The graphs labelled by "fb-u" and "fb-v" represent the feedback signal corresponding to the control variables  $u$  and  $v$ , respectively. For details of the experiment see the text.

## References

- Agrawal, P., C. Lee, H. Lim, and D. Ramkrishna (1982). Theoretical investigations of dynamic behavior of isothermal continuous stirred tank biological reactors. *Chemical Engineering Science* 37.
- Anderson, C. and W. Miller, III (1992). Challenging control problems. In *Neural Networks for Control*, Neural Network Modeling and Connectionism, pp. 475–510. MIT Press, Cambridge
- Ben-Israel, A. and T. Greville (1974). *Generalized Inverses: Theory and Applications*. Pure and Applied Mathematics, Wiley-Interscience, New York: J. Wiley & Sons.
- Bregel, D. and W. Seider (1989). A multi-step nonlinear predictive controller. *Ind. Eng. Chem. Res.* 28, 1812–1822
- Goodwin, G. and K. Sin (1984). *Adaptive filtering, prediction and control*. Prentice-Hall, Englewood Cliffs, NJ, USA.
- Grossberg, S. and M. Kuperstein (1986). *Neural Dynamics of Adaptive Sensory-motor Control: Ballistic Eye Movements*. Elsevier, Amsterdam
- 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.

- Landau, Y. (1979). *Adaptive Control: the Model Reference Approach*. Marcel Dekker, New York.
- Lewis, F., C. Abdallah, and D. Dawson (1993). *Control of Robot Manipulators*. New York: MacMillan.
- Lewis, F., K. Liu, and A. Yesildirek (1995, May). Neural net robot controller with guaranteed tracking performance. *IEEE Trans. on Neural Networks* 6(3), 703–715.
- Lovelock, D. and H. Rund (1975). *Tensors, differential forms, and variational principles*. Pure and Applied Mathematics. A Wiley-Interscience Series of Texts, Monographs, and Tracts. Wiley-Interscience, New York.
- Minsky, M. (1961). Steps towards artificial intelligence. In *Proc. of the Institute of Radio Engineers*, pp. 8–30. Reprinted in *Computers and Thought*, E.A. Feigenbaum and J. Feldman, editors, 406–450, McGraw-Hill, New York, 1963.
- Miyamoto, H., M. Kawato, T. Setoyama, and R. Suzuki (1988). Feedback-error-learning neural network for trajectory control of a robotic manipulator. *Neural Networks* 1, 251–265.
- Narendra, K. and R. Monopoli (1980). *Applications of adaptive control*. Academic Press, New York.
- Psaltis, D., A. Sideris, and A. Yamamura (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 W. Li (1991). *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, NJ, USA.
- Szepesvári, C. and A. Lőrincz (1995). Dynamic state feedback neurocontroller for compensatory control. *Neural Networks*, submitted.
- Szepesvári, C. and A. Lőrincz (1996). An artificial neural network for robust, adaptive control in structured environments. *Neural Network World*, submitted.
- Ungar, L. (1992). A bioreactor benchmark for adaptive network-based process control. In *Neural Networks for Control*, Neural Network Modeling and Connectionism, pp. 387–402. MIT Press, Cambridge.
- Vemuri, V. (1993). Artificial neural networks in control applications. *Advances in Computers* 36, 203–251.
- Werbes, 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*, Lund, Sweden, pp. 1–5. Lund Institute of Technology.
- Widrow, B., J. McCool, and B. Medoff (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.

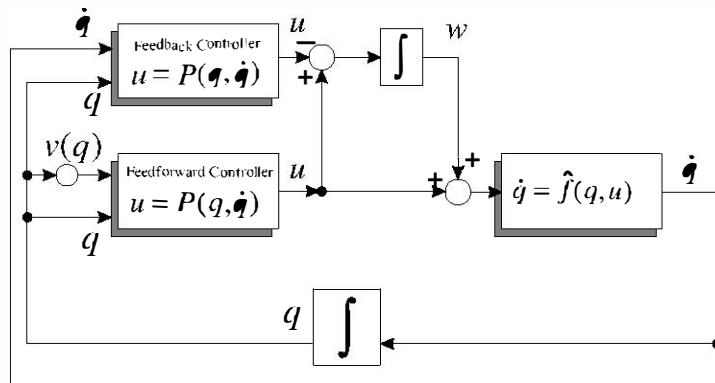


Figure 1: Compensatory control by doubling the inverse dynamics controller

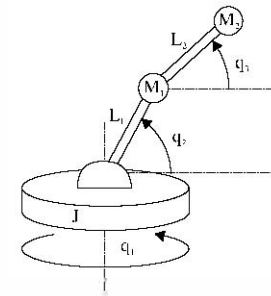


Figure 2: Idealized 3-joint robotic manipulator

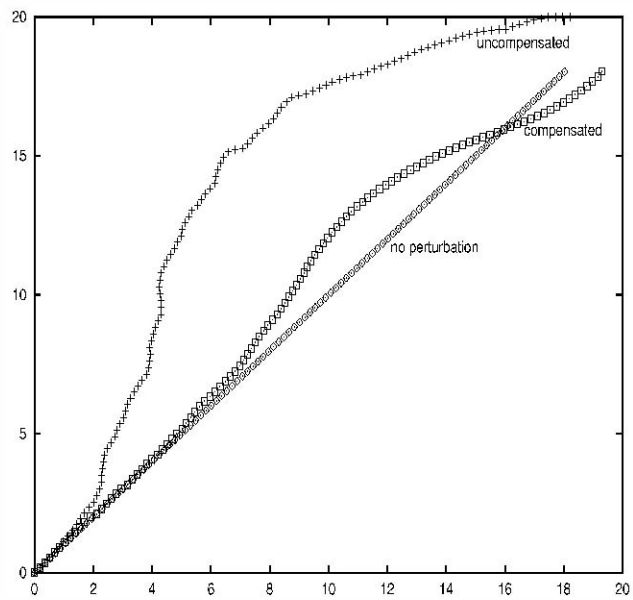
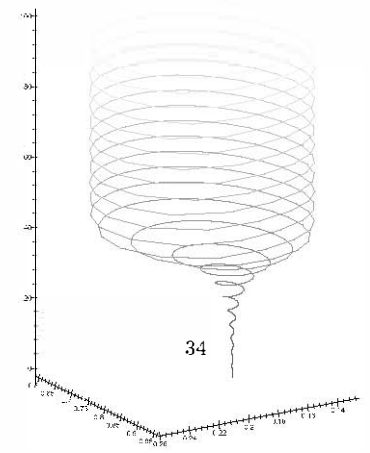
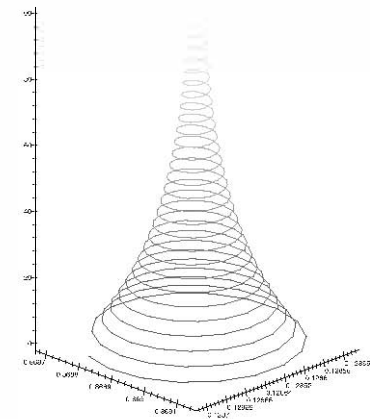


Figure 3: Typical trajectories in the presence of inhomogeneous perturbation

33



34

Figure 4: Behaviour of the uncontrolled bioreactor for different control values

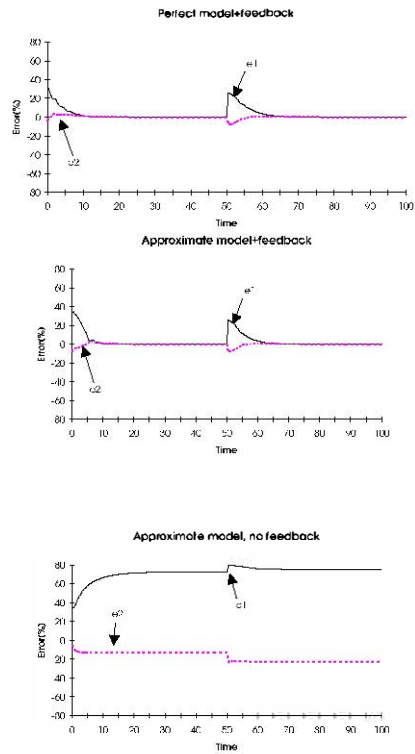


Figure 5: Bioreactor control: Error of control in percent vs. time

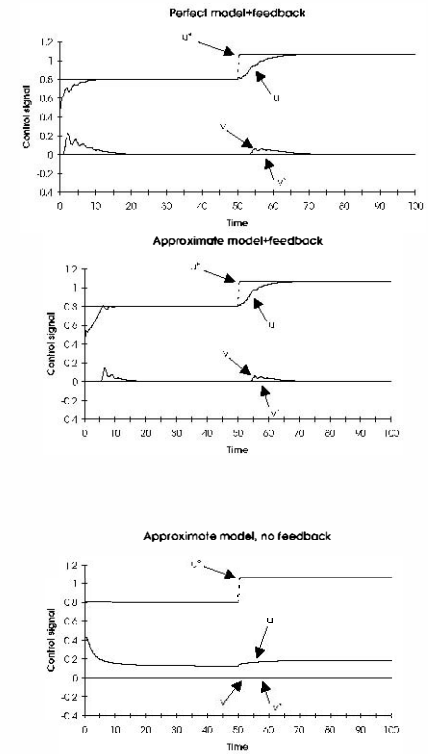


Figure 6: Bioreactor control: Control values vs. time

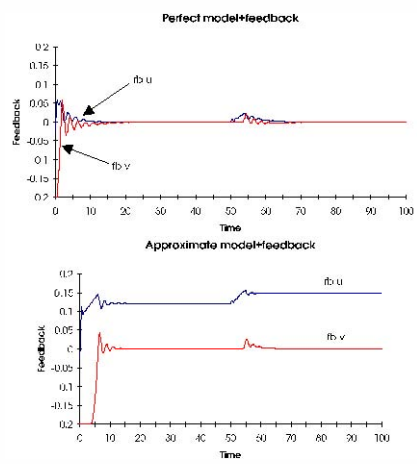


Figure 7: Bioreactor control: Feedback vs. time