APPROXIMATE INVERSE-DYNAMICS BASED ROBUST CONTROL USING STATIC AND DYNAMIC FEEDBACK

CSABA SZEPESVÁRI
Department of Adaptive Systems, Institute of Isotopes of the Hungarian Academy of Sciences, Budapest, P.O. Box 77, Hungary, H-1525

Research Group on Artificial Intelligence
University of Szeged, Szeged, Aradi vrt. tere 1, Hungary, H-6720

ANDRÁS LŐRINCZ
Department of Adaptive Systems,
University of Szeged, Szeged, Aradi vrt. tere 1, Hungary, H-6720

Department of Chemical Physics, Institute of Isotopes of the Hungarian Academy of Sciences, Budapest, P.O. Box 77, Hungary, H-1525

It is rigorously shown that inverse-dynamics models can be used to stabilize plants of any order provided that the inverse-dynamic model is used in a mixed mode fashion, in that of a 'Static and Dynamic' State-feedback (SDS) mode. When the resulting controller is used for tracking increasing the gain of the dynamic feedback decreases the tracking error. Yet another attractive feature of the SDS scheme is that the inverse-dynamics model can be tuned on-line by *any* adaptation mechanism without cancelling stability if the conditions of the non-adaptive stability theorem hold at any time instant. Computer simulations of the control of a chaotic bioreactor and a 'realistic' robotic manipulator demonstrate the robustness of the approach. It is shown that SDS control will yield zero asymptotic error when controlling the bioreactor using an inverse-dynamics model which when used in a traditional mode would yield intolerably large errors. In the case of the robotic arm simulations the effects of perturbation and sampling frequency are investigated and the SDS control is compared with the non-adaptive computed torque method. A fully self-organizing associative neural network architecture that can be used to approximate the inverse-dynamics in the form of a Position-and-Direction-to-Action (PDA) map is also described. Similarities between the basal ganglia – thalamocortical loops and the SDS scheme are discussed and it is argued that the SDS scheme could be viewed as a model of higher order motor functions of these areas.

1 Introduction

The issues in controlling generic non-linear plants include the design of stable control laws based on a good understanding of the plant under study and its environment, optimal control in which a cost minimizing controller is sought for, along with collision-free trajectory planning for autonomous vehicles or robot manipulators, adaptive control for compensation of parameter uncertainties, and robust control. In this chapter we present a recently introduced two-step control system which makes it possible to incorporate optimal control and adaptive robust control in a single model [1]. Our control system consists of two parts and so it is somewhat similar to Model Predictive Control (MPC) [2]. To help to motivate our departure from MPC a brief overview of it is first dealt with.

In MPC (also called receding horizon control) a finite horizon optimal control problem is solved and the resulting controller is used during a time interval shorter than the horizon of the optimal control problem. The process is then repeated indefinitely. Stability and robustness is ensured by including artificial constraints such as conservative control and state constraints, in the optimal control problem [3,4,5]. Adaptive model predictive control is even harder to achieve since the solution set of the optimal control problem corresponding to rough parameter estimates may well be empty. This can be overcome by introducing parameter perturbations which restore controllability but do not increase the estimation cost [6,7]. MPC is a natural area for the application of artificial neural networks (ANNs) which can then be used to identify and implement a plant model.

Solutions of optimal control problems with *stationary* plant dynamics and cost structure can be given in terms of an optimal static state-feedback law [8]. Such a cost structure may arise in minimum energy control or in path planning problems. If the dynamics of the plant is given by $\dot{\mathbf{q}} = f(\mathbf{q}, \mathbf{u})$, where $\mathbf{q}$ is the state of the plant, $\mathbf{u}$ is the control and $\mathbf{u}^*(\mathbf{q})$ denotes the optimal feedback law then the equation of motion of the plant becomes

$$\dot{\mathbf{q}} = \mathbf{v}^*(\mathbf{q}) \tag{1}$$

where $\mathbf{v}^*(\mathbf{q}) = f(\mathbf{q}, \mathbf{u}^*(\mathbf{q}))$. Equivalently, one may start from Eq. 1 and search for a static state-feedback law $\mathbf{u} = \mathbf{u}^*(\mathbf{q})$ which governs the plant to track the *speed-field* $\mathbf{v}^*(\mathbf{q})$. This corresponds to the approach employed in the present paper. The speed-field to be tracked need not be the solution of an optimal control problem but may be designed by other means, and one can then take into account a variety of objectives and constraints. One such objective might be the robustness of control, i.e. $\mathbf{v}^*$ could be designed in such a way that tracking it with a bounded error ($||\dot{\mathbf{q}} - \mathbf{v}^*(\mathbf{q})|| < b$) results in a particular desirable behaviour. Note that since in practice exact models are not available it is only bounded tracking that can be hoped for. As another example assume that the task is to follow a given trajectory. Then $\mathbf{v}^*$ could be designed in such a way that the desired trajectory attracts the solutions of Eq. 1, i.e. that the equilibrium solution of Eq. 1 should be totally stable. Another important issue is the efficiency of the speed-field planning algorithm. Just as in MPC, one may want to redesign the speed field frequently so that changes in the environment (caused, e.g. by moving obstacles) are always reflected in the actual speed-field. Yet another design limitation, which is also considered by MPC and which probably affects $\mathbf{v}^*$, are constraints on the controls.

It is not only robust control but also adaptive control which can be used to deal with model uncertainties. Currently, the topic of adaptive control of non-linear plants is an area of active research. ANNs have been used both in on-line as well as off-line learning modes for the approximation of various plant functions, such as the plant's kinematics [9,10], the plant's (forward) dynamics [11,12,13], and the plant's inverse dynamics [14,15,16]. Recently interest has grown in theoretical issues such as stability of on-line adapted ANN controllers.

Most of the rigorous stability results are developed for feedback-linearisable systems where an on-line adapting inverse-dynamics model is utilised to cancel out non-linearities and render the system (approximately) linear and controllable. Once achieved it is then open to traditional methods available for linear systems, such as PD or PID controllers. An approximate inverse-dynamics can be found by estimating a forward model and then computing the inverse-dynamics from the forward model using the methods of static feedback design [17]. Another approach is to estimate the inverse-dynamics directly [18]. In the latter case it is usually assumed that the state vector of the normal form [19] is available for measurement – a requirement which can be hard to satisfy, but is met for mechanical systems such as a rigid-link robotic manipulators. The most popular among several variants of this approach is the so-called computed torque method [20], where the sum of the output of a stabilizing PD controller and the desired acceleration is inputted into the inverse-dynamics model. Other variants include the feedback-error learning model of Gomi and Kawato [21], where inverse-dynamics is used to cancel the non-linearities and an external PD controller is applied to stabilize the system, or the method of Yabuta and Yamada who apply the computed torque method but without the PD controller [22]. This latter approach can be dangerous, as was noted by the authors themselves and others [23], because in certain cases the system may become unstable.

All the adaptive control schemes which are known to be stable require a determination of the exact form of the parameter tuning (learning) procedures. Additionally, and the best of our knowledge, all the general schemes to date that have proved to be stable presume a linear parametrization (a few exceptions exist but it seems that there is no general solution to this problem [24,25]). Linear systems suffer from the "curse of dimensionality" which – as it is hoped by many in the ANN field – can be circumvented by using non-linearly parametrized systems [26,27]. Beyond robustness our control scheme has the outstanding property that the approximate inverse-dynamics can also be adapted on-line using *any* adaptation mechanisms, as long as the conditions of the *static* control theorem hold *uniformly.* Note that the obtained stability result is *global*: there is no need to bound the initial error estimate.

This chapter is organized as follows. In Section 2 we introduce two variants of the SDS control scheme, the first variant having been designed to control first order plants (i.e. plants whose relative degree is one), while the second variant can be used to control plants of any order. An outline of the stability proofs of the control schemes is given, whose details will be published elsewhere [28]. Then two examples are furnished to illustrate the workings of our theory. In Section 3.1 simulations for controlling a chaotic bioreactor by the first SDS variant are presented. In the section following it the second scheme is used to control a 3-joint robotic manipulator when the physical quantities of the robot arm, such as the payload and friction coefficients, are imprecise. While the experimental and theoretical results were found to be in good agreement we also felt it necessary to discuss some issues outside of the domain of the theory, such as the effect of low sampling rates. In Section 4 the neural network architecture which can implement the speed-field planning mechanism and approximation of the inverse-dynamics in one go is quickly reviewed. Afterwards the biological relevance of the SDS scheme is discussed in Section 5. Finally the chapter is rounded off with a summary and further discussion.

2 Theoretical Results: Robustness and Stable On-line Adaption

Let $D \subseteq \mathbf{R}^n$ denote the domain of the plant's state with the equation of motion given by

$$\mathbf{u} = \mathbf{A}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{b}(\mathbf{q}), \tag{2}$$

where $\mathbf{q}$ is the state vector of the plant and $\mathbf{u} \in \mathbf{R}^m$ is the control. For the sake of notational simplicity the dependence of $\mathbf{A}$ and $\mathbf{b}$ on $\mathbf{q}$ will from now on not be explicitly represented. Now let us assume that we have an estimate of the true inverse-dynamics function $\Phi(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{A}\dot{\mathbf{q}} + \mathbf{b}$, given by $\hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}})$. The SDS Feedback Control equations can then be written as

$$\begin{aligned} \mathbf{u} &= \mathbf{u}_f(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}(\mathbf{q})) + \mathbf{w}, && (3) \\ \dot{\mathbf{w}} &= \Lambda\left(\hat{\mathbf{\Phi}}(\mathbf{q}, \mathbf{v}(\mathbf{q})) - \hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}})\right), && (4) \end{aligned}$$

where $\mathbf{u}_f$ is the so called feedforward controller (to be specified later), $\Lambda > 0$ is the gain of feedback, and the desired motion is determined by a speed-field tracking task that prescribes the speed vector $\dot{\mathbf{q}}$ of the plant as a function of the state vector:

$$\dot{\mathbf{q}} = \mathbf{v}(\mathbf{q}). \tag{5}$$

Speed field tracking is not typical in the control literature, but arises naturally if we consider stationary optimal-control problems such as path planning tasks [29]. Conventional control tasks, such as *point-to-point control* and *trajectory tracking* cannot be exactly rewritten in the form of speed-field tracking and vice versa [1,30]. As was argued in the Introduction the speed-field tracking task has the advantage that the designer can incorporate several objectives into the form of the speed-field to be tracked hence extend the model's range of possibilities.

In what follows the usual definition of positivity for fields of square matrices will be required.

DEFINITION 2.1 *Let $\mathbf{M} : D \to \mathbf{R}^{p\times p}$, $p > 0$. $\mathbf{M}$ is said to be positive definite uniformly over D iff for all $\mathbf{q} \in D$ the term $\mathbf{M}(\mathbf{q})$ is positive definite and there exists an $\epsilon > 0$ such that $\lambda_{\min}(\mathbf{M}(\mathbf{q})) > \epsilon$ holds for all $\mathbf{q} \in D$. Uniform negative definiteness can be similarly defined.*

If $\mathbf{M}$ is a real quadratic matrix then let $\mathbf{M} > 0$ denote that $\mathbf{M}$ is positive definite. Similarly, if $\mathbf{M}$ is a matrix field over D, let $\mathbf{M} > 0$ denote that $\mathbf{M}$ is uniformly positive definite over D .

THEOREM 2.2 *Let the plant whose equation is given by Eq. 2 be governed by Eqs. 3 and 4 over the domain D. Let $\hat{\mathbf{v}}(\mathbf{q}) = \mathbf{u}_f(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}(\mathbf{q}))$ be the speed-field followed by the plant which is governed by the feedforward controller alone. If*

1. *$\mathbf{u}_f$ does not depend on $\dot{\mathbf{q}}$, that is, $\mathbf{u}_f(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) = \mathbf{u}_f(\mathbf{q}, \mathbf{v})$*
2. *$\hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{A}}\dot{\mathbf{q}} + \hat{\mathbf{b}}$*
3. *$\mathbf{A}^T\mathbf{A}$ and $\hat{\mathbf{A}}^T\mathbf{A}$ are uniformly positive definite over D ('sign-properness' assumption)*
4. *$\mathbf{v}(\mathbf{q}), \hat{\mathbf{v}}(\mathbf{q})$ and $\mathbf{A}(\mathbf{q})$ are uniformly bounded and have uniformly bounded derivatives w.r.t. $\mathbf{q}$ over D,*

then for all $\Lambda > 0$ the error of tracking $\mathbf{v}(\mathbf{q})$, $\mathbf{e} = \mathbf{v}(\mathbf{q}) - \dot{\mathbf{q}}$, is eventually uniformly bounded and the eventual bound b of the tracking-error can be made arbitrarily small; more specifically $b = \mathcal{O}(1/\Lambda)$, and the eventual bound for the time reaching $||\mathbf{e}|| \leq b$ is proportional to Λ.

The proof is based on an extension of Liapunov's second method [1]. One can make use of the error equation

$$\mathbf{A}\mathbf{e} = \mathbf{A}(\mathbf{v}(\mathbf{q}) - \hat{\mathbf{v}}(\mathbf{q})) + \mathbf{w} \tag{6}$$

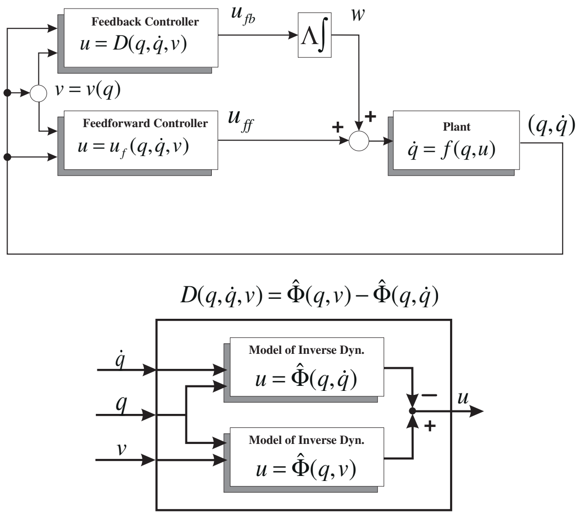


Figure 1: The general SDS scheme
The 'Static and Dynamic' State-Feedback Controller is composed of two controllers. The approximate inverse-dynamics function $\hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}})$ is used to develop the compensatory signal $D(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v})$ which then renders the tracking error to remain bounded. The feedforward loop driven by $\mathbf{u} = \mathbf{u}_f(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v})$ alone should be bounded, and should have restricted growth over the allowed domain.

and the function $L = \frac{1}{2}\mathbf{e}^T \left(\mathbf{A}^T \mathbf{A}\right) \mathbf{e}$, which is then shown to be an appropriate semi-Liapunov function of the closed-loop system, thus completing the proof.

Note that *we do not assume that either* $\mathbf{A}$ *or* $\hat{\mathbf{A}}$ *is invertible* (not even in the generalized[31] sense). Note too that the assumptions of this theorem follow from the assumptions of the original SDS stability theorem [1], which is cited below:

THEOREM 2.3 *Assume that the feedforward controller is given by the approximate inverse dynamics*

$$\mathbf{u}_f(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) = \hat{\mathbf{\Phi}}(\mathbf{q}, \mathbf{v}) = \hat{\mathbf{A}}\mathbf{v} + \hat{\mathbf{b}} \tag{7}$$

and that the following assumptions hold:

1. $\mathbf{A}$ *and* $\hat{\mathbf{A}}$ *are invertible (in the generalized sense)*

2. $\mathbf{A}^{-1}$ *and* $\mathbf{D} = \hat{\mathbf{A}}^{-1}\mathbf{A}$ *are bounded away from singularities uniformly over* D

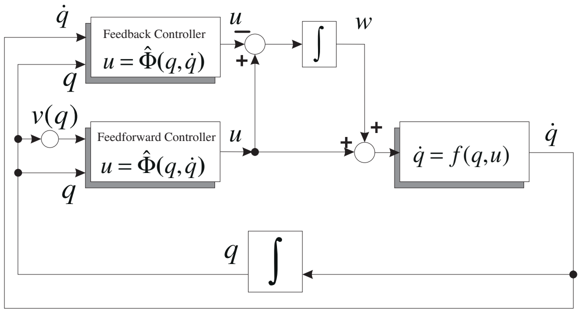


Figure 2: Doubling the role of the inverse-dynamics controller
If the plant is a first-order type the approximate inverse-dynamics based controller can be used as the feedforward controller while preserving stability.

3. *the symmetrized perturbation matrix* $\mathbf{D}+\mathbf{D}^T$ *satisfies the sign-properness condition* $\mathbf{D} + \mathbf{D}^T > 0$

4. $\mathbf{A}^{-1}, \mathbf{b}, \mathbf{v}, \mathbf{D}, \hat{\mathbf{b}}$ *are bounded and have continuous, bounded derivatives over* D

Then for all $\epsilon > 0$ *there exists a gain* Λ *and absorption time* $T > 0$ *such that for all* $\mathbf{e}(0)$ *that satisfy* $||\mathbf{e}(0)|| < K\Lambda$ *it holds that* $||\mathbf{e}(t)|| < \epsilon$*, provided* $t > T$ *and the solution can be extended up to time* t*. Here* K *is a fixed positive constant and* $\mathbf{e}(0)$ *denotes the initial value of* $\mathbf{e}$*. Further,* $\Lambda \sim O(1/\epsilon)$*.*

The original proof relies on the Liapunov function candidate $L = \mathbf{e}^T\mathbf{e}$. It can readily be seen that the assumptions of Theorem 2.2 follow from the assumptions used in this theorem. In particular, if $\mathbf{A}$ ($\hat{\mathbf{A}}$) is invertible over D then $\mathbf{A}^T\mathbf{A} > 0$ ($\hat{\mathbf{A}}^T\hat{\mathbf{A}} > 0$). Note furthermore that our new sign-properness assumption $\hat{\mathbf{A}}^T\mathbf{A} > 0$ follows from the original sign-properness assumption $\mathbf{D} + \mathbf{D}^T > 0$ since $\mathbf{D} > 0$ is equivalent to $\mathbf{D} + \mathbf{D}^T > 0$ and $\hat{\mathbf{A}}^T\mathbf{A} = (\hat{\mathbf{A}}^T\hat{\mathbf{A}})(\hat{\mathbf{A}}^{-1}\mathbf{A}) = (\hat{\mathbf{A}}^T\hat{\mathbf{A}})\mathbf{D}$. So the assumptions made in Theorem 2.2 are weaker than those in the original theorem. Yet the inferences made using Theorem 2.2 are stronger since the boundedness of the initial error $\mathbf{e}(0)$ is no longer required. This result shows that the closed-loop system is *globally* stable.

The control scheme, described by Eqs. 3,4 and specified by the assumptions stated in Theorem 2.2, is shown in Fig. 1. The particular form of the feedforward controller cited in Theorem 2.3 also enables one to specialize the controller to the one shown in Fig. 2.

As it can be seen in Fig. 2, two identical forms of the estimated inverse-dynamics function are utilized, one in a feedforward position with its input

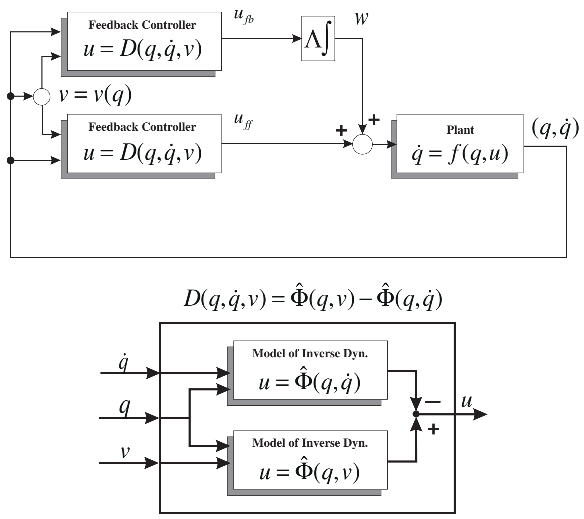


Figure 3: Robust control of plants of any order
This scheme differs from the one depicted in Fig. 1 since now both the feedforward and the feedback controllers are given in terms of approximate inverse-dynamics. Our theorem also enables one to adapt the inverse-dynamics on-line; the stability of the control loop will not be affected.

from the state vector of the plant and the 'desired speed vector' $\mathbf{v}(\mathbf{q})$, while the other is in the feedback position with its inputs from the state vector of the plant and the measured or 'experienced' speed vector, $\dot{\mathbf{q}}$. The difference of the outputs of the two controllers is computed (Eq. 4), amplified and time integrated and then added to the output of the feedforward controller (Eq. 3).[a] Note that the boundedness of $\hat{\mathbf{v}}$ means that the plant is essentially stabilized by $\mathbf{u}_f$ in the sense that $\dot{\mathbf{q}} - \mathbf{v}(\mathbf{q}) = \hat{\mathbf{v}}(\mathbf{q}) - \mathbf{v}(\mathbf{q})$ is bounded. This postulate may be hard to meet without the invertibility condition on $\mathbf{A}$. If $\mathbf{A}$ is invertible then the order (more precisely, the relative degree[32]) of the plant can be shown to be equal to one.

A slight modification is needed in order to generalize the control scheme to plants of any order, the modified scheme being shown in Fig. 3. The reader will duly note that that the only real difference between the Figs. 1 and 3 is that the feedforward controller is replaced by the feedback controller. Now we

[a]It is somewhat unfortunate that the speed vector $\dot{\mathbf{q}}$ and thus an additional sensory system is needed for the control, but other methods also require this[32]. Still it is feasible that our method can be extended in a way similar to the one suggested by Hsu et al.[33].

will give the corresponding, modified version of Theorem 2.2.

THEOREM 2.4 *Assume that the feedforward controller has the form*

$$\mathbf{u}_f(\mathbf{q}, \dot{\mathbf{q}}, \mathbf{v}) = \hat{\mathbf{\Phi}}(\mathbf{q}, \mathbf{v}) - \hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}),$$

(which is the same as the input of the feedback integrator). Further, assume that the following hold:

1. $\hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}}) = \hat{\mathbf{A}}\dot{\mathbf{q}} + \hat{\mathbf{b}}$
2. $\mathbf{A}^T\mathbf{A}$, $\mathbf{A}^T\hat{\mathbf{A}}$ *and* $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$ *are uniformly positive definite over* D
3. $\mathbf{A}$, $\mathbf{v}$, $\mathbf{b}$ *are bounded and have uniformly bounded derivatives w.r.t.* $\mathbf{q}$ *over* D

Then for all $\Lambda > 0$ *the error of tracking* $\mathbf{v}(\mathbf{q})$, $\mathbf{e} = \mathbf{v}(\mathbf{q}) - \dot{\mathbf{q}}$, *is eventually uniformly bounded and, further, the eventual bound* b *of the tracking-error can be made arbitrarily small. More specifically* $b = \mathcal{O}(1/\Lambda)$, *and the eventual bound for the time reaching* $||\mathbf{e}|| \leq b$ *is proportional to* Λ.

The proof of this theorem is just like to the first one and relies on a Liapunov-function approach. First of all the relation

$$(\mathbf{A} + \hat{\mathbf{A}})\mathbf{e} = \mathbf{A}\mathbf{v}(\mathbf{q}) + \mathbf{b} - \mathbf{w} \tag{8}$$

can be employed to show that $L = \frac{1}{2}\mathbf{e}^T\left[(\mathbf{A} + \hat{\mathbf{A}})^T(\mathbf{A} + \hat{\mathbf{A}})\right]\mathbf{e}$ is an appropriate semi-Liapunov function. Then the proof is complete.

The subtle point of this theorem (compared to the first one) is the requirement that $\hat{\mathbf{A}}^T\hat{\mathbf{A}}$ should be uniformly positive definite over D. At the same time the condition on the stability of the feedforward part is superfluous here, but the other conditions are the same as before. The theorem gives rise to a global stability result just like that in Theorem 2.2. Notice too that the particular form of the feedforward and feedback controllers make it unnecessary to build an estimate of $\mathbf{b}$.

The main difference between Theorems 2.2 & 2.4 and their proofs is that in the error equation corresponding to the first scheme (Eq. 6) the r.h.s. is a function of the approximated inverse-dynamics through $\hat{\mathbf{v}}$, whereas in the case of Eq. 8 there is no such dependence. This latter fact can be exploited to show that the above proof remains valid if $\hat{\mathbf{A}}$ and $\hat{\mathbf{b}}$ vary in time but the conditions of the theorem remain valid at every instant. Thus we get the following important corollary:

COROLLARY 2.5 *Suppose that the conditions of Theorem 2.4 hold and also that* $\hat{\mathbf{A}} = \hat{\mathbf{A}}(t)$ *and* $\hat{\mathbf{b}} = \hat{\mathbf{b}}(t)$. *Next assume that* $\mathbf{A}^T\hat{\mathbf{A}}(t)$ *and* $\hat{\mathbf{A}}^T(t)\hat{\mathbf{A}}(t)$ *are uniformly positive-definite over* D *and for all* $t > 0$, *and that* $\hat{\mathbf{A}}(t)$ *is bounded. Then the conclusions of the above theorem still hold.*

The lower bound of Λ is inversely proportional with $\inf_{t,\mathbf{q}} \lambda_{\min}(\mathbf{A}^T(\mathbf{q})\hat{\mathbf{A}}(\mathbf{q},t))$ and $\inf_{t,\mathbf{q}} \lambda_{\min}(\hat{\mathbf{A}}^T(\mathbf{q},t)\hat{\mathbf{A}}(\mathbf{q},t))$ and proportional to $\sup_{t,\mathbf{q}} ||\hat{\mathbf{A}}(\mathbf{q},t)||+||\hat{\mathbf{A}}(\mathbf{q},t)||^2$. The uniform positive-definiteness conditions of the corollary follow, e.g. when $\hat{\mathbf{A}}$ is bounded away from singularities uniformly over D: an assumption often required in adaptive control [32]. It is clear, too that the stability result does not depend on the specific adaptation mechanism utilized, which is a fairly rare condition in adaptive control theory. However, one has to provide an additional proof to show that the conditions required for $\hat{\mathbf{A}}$ are obeyed.

3 Computer simulations

3.1 Control of a chaotic plant

Chemical systems can be relatively simple as they usually have only a few variables, but still troublesome to control due to strong nonlinearities which are difficult to model accurately. A prime example of this 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 (the "substrate"), produce products (both wanted and unwanted) and more cells. The simplest version of the bioreactor is a continuous-flow stirred-tank reactor in which cell growth depends only on the nutrient being fed into 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:

$$\begin{aligned} \dot{c}_1 &= -c_1(u+v) + c_1(1-c_2)e^{\frac{c_2}{\gamma}} \\ \dot{c}_2 &= -c_2(u+v) + Sv + c_1(1-c_2)e^{\frac{c_2}{\gamma}} \frac{1+\beta}{1+\beta-c_2}, \end{aligned} \tag{9}$$

where c_1 and c_2 are, respectively, dimensionless mass and substrate conversions [34]. 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 β and γ 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 has been suggested as a control benchmark problem [35]. The system is difficult

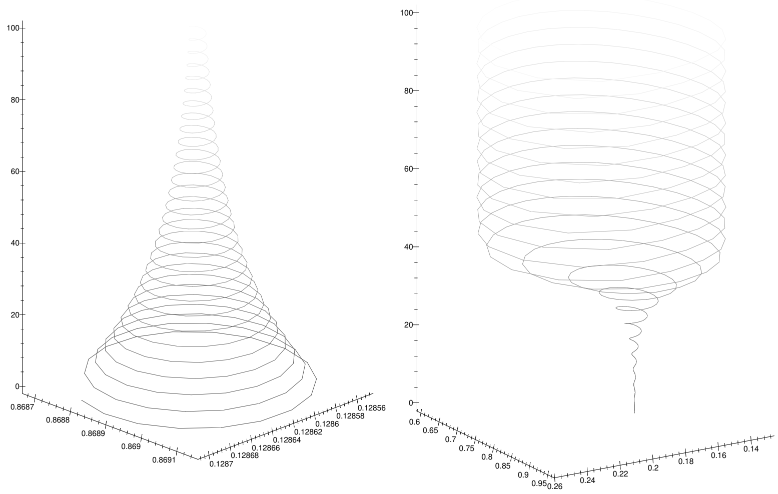


Figure 4: Behaviour of the bioreactor for different constant flow rates
The figures show the evolution of the bioreactor state variables as described by Eq. 9 with (u, v) = (0.8012,0.0004) and (u, v) = (1.0652,0.0004) for the l.h.s. and r.h.s. subfigures, respectively. In both cases the reactor was started 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, while the left and right axes represent the c_2 and c_1 parameters, respectively.

to control for several reasons. Firstly, the uncontrolled equations are highly nonlinear and exhibit limit cycles. Secondly, optimal behaviour occurs near an unstable region. Note that for $\beta = 0.02$, $\gamma = 0.48$ and $S = 1$ a Hopf bifurcation occurs at $u = 0.829$, $v = 0.0004$ as shown in Fig. 4.

Our experiments with this plant consisted of two phases corresponding to two sets of target values for c_1^* and c_2^*. In the first stage 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 $(c_1^*, c_2^*) = (0.1287, 0.8688)$, corresponding to a stable fixed point of the reactor with flow rates $(u^*, v^*) = (0.8012, 0.0004)$. In the second stage the target cell mass was increased again, this time to $(c_1^*, c_2^*) = (0.1737, 0.7978)$ with equilibrium control values $(u^*, v^*) = (1.0652, 0.0004)$. This change in set-point was sufficient to shift from a stable regime into the domain of attraction of a limit cycle. Even if the correct model is known, small errors in parameters give rise to rather inaccurate control when a feedforward controller is used: an error of 2% in γ leads to 50% error in the target cell mass [36]. The control was applied with a zero order hold of 0.5 seconds while simulating the bioreactor

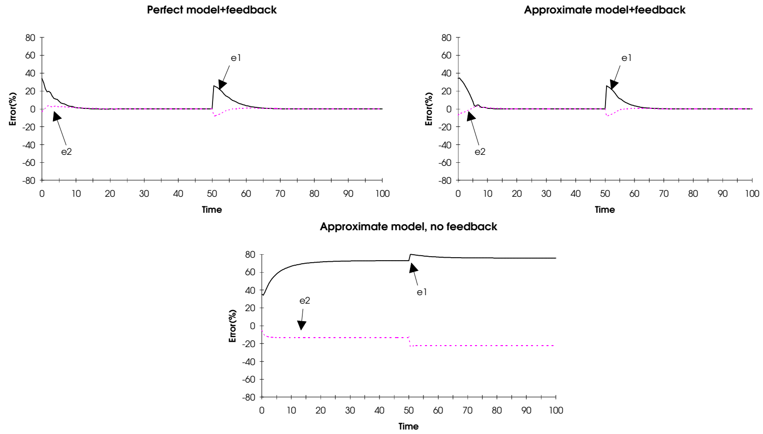
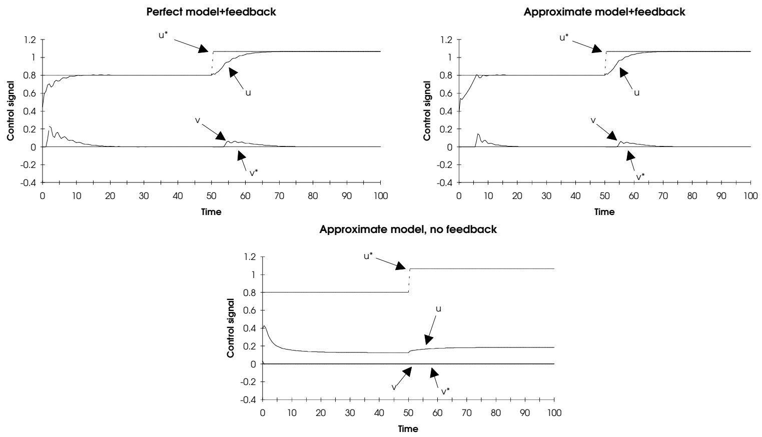


Figure 5: Error of control in percent vs. time
The error percentage of control is denoted by $e_i = 100(c_i - c_i^*)/c_i^*$, $i = 1, 2$, where (c_1^*, c_2^*) is the desired set-point of the bioreactor. After 50 seconds a new set-point was designated that corresponded to an unstable equilibrium state of the plant. When feedback is in effect the error quickly reduces to zero. But without feedback the error remains high even if the inverse-dynamics is moderately imprecise. For details of the experiment see the text. (with permission[30])

with Euler's method and setting $dt = 0.01$ s. Both phases lasted 50 seconds.

We tested the first order SDS Control (Eqs. 3,4 and 7) using the inverse-dynamics corresponding to Eq. 9 with $\hat{\gamma} = 1.1\gamma$. Since the change of γ results in an additive perturbation of the inverse-dynamics, the SDS Control can be applied without any restrictions and the theory predicts that SDS Control will yield a bounded tracking error. The speed-field was given by $v(c_1, c_2) = \lambda(c_1^* - c_1, c_2^* - c_2)$, where λ was determined such 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 Fig. 5.

The first of the three subfigures of Fig. 5 corresponds to the case when the inverse dynamics model was ideal, 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, implying 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 than 60%. Fig. 5 suggests that SDS Control is able to *perfectly* compensate for the perturbation even though the perturbation is highly nonlinear. This also follows from the theory since it can be shown that provided the plant reaches a sufficiently small neighborhood of the designated equilibrium state and can be linearized, then the feedback can compensate in a perfect fashion [1]. Fig. 3.1 depicts the control variables and the ideal control values in the three cases. This figure reinforces the impression that SDS Control is efficient. Throughout these experiments the feedback gain Λ was 1.



The vector (u^*, v^*) gives the ideal equilibrium control, while (u, v) is the actual control. After 50 seconds a new set-point is designated that corresponds to an unstable equilibrium of the plant so the ideal control values can not be assumed from the start. For details of the experiment see the text. (with permission[30])

3.2 Controlling a second order plant

Consider the following equation describing the motion of a rigid-link robot arm (see Fig. 6)

$$\mathbf{u} = \mathbf{M}(\theta)\ddot{\theta} + \mathbf{V}(\theta, \dot{\theta}) - k\dot{\theta}.$$

Here $\theta = (\theta_1, \theta_2, \theta_3)$ is the vector of angular positions of the robot (θ_1 is the angular position of the robot base axis, θ_2 is the angular elevation of the upper arm above horizontal, θ_3 is the angular elevation of 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), $\mathbf{M}(\theta)$ is the inertia matrix, $\mathbf{V}(\theta, \dot{\theta})$ represents the Coriolis, centripetal forces and the gravity loading, and $k > 0$ is the friction coefficient. Note that this equation has the form of Eq. 2 with $\mathbf{q}^T = [\theta^T, \ \dot{\theta}^T]$, $\mathbf{A}(\mathbf{q}) = [\mathbf{0}, \ \mathbf{M}(\theta)]$ and $\mathbf{b}(\mathbf{q}) = \mathbf{V}(\theta, \dot{\theta}) - k\dot{\theta}$. Unfortunately, the additive term is not bounded if $||\dot{\theta}||$

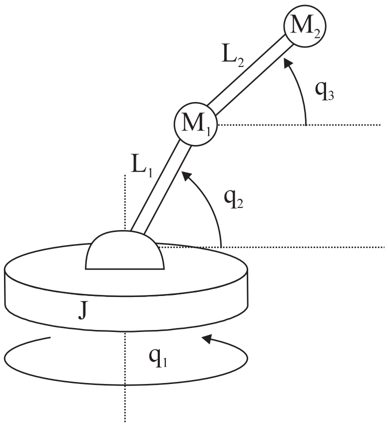


Figure 6: Idealized 3-joint robotic manipulator
The dynamics of this 3-joint robotic manipulator is highly non-linear. A typical perturbation is when the manipulator grasps (or releases) an object. This can be modelled, say, by changing the mass (M_2) at the end-effector. The compensation for such a perturbation is difficult, especially when the mass of the object is large compared to the mass of the manipulator.

goes to infinity. This means that the SDS scheme should be applied only if one can ensure *a priori* that $\dot{\theta}$ remains bounded (more specifically, a bound can be given for $\dot{\theta}$) during the control. Since the plant is a mechanical system boundedness can be achieved according to the law of energy conservation provided that the controls remain bounded, which can be achieved e.g. by some variant of the σ-modification scheme or a projection method. This condition was not implemented in the experiments, so the results once again show the robustness of the SDS scheme.

Assume that the task is defined in terms of a certain speed-field $\mathbf{s}(\theta)$ given in the configuration space, i.e., the task is to make sure

$$\dot{\theta} = \mathbf{s}(\theta).$$

We admittedly abuse the terminology here, in that we use the expression "speed-field" to mean a vector-field both over the configuration- and the state-spaces. The speed-field over the state-space needs another component beyond the speed-field over the configuration space, the field of desired accelerations. So let this be defined as the difference between the desired and the actual speed:

$$\mathbf{a}(\theta, \dot{\theta}) = \mathbf{s}(\theta) - \dot{\theta}$$

In terms of $\mathbf{q}$ the desired speed-field is given by

$$\mathbf{v}(\mathbf{q})^T = [\mathbf{s}(\theta)^T,\ \mathbf{a}(\theta,\dot\theta)^T] = [\mathbf{s}(\theta)^T,\ \mathbf{s}(\theta)^T - \dot\theta^T].$$

Expressed in another way, $\mathbf{v}(\mathbf{q})^T = [\mathbf{v}(P_1\mathbf{q})^T, \quad \mathbf{v}(P_1\mathbf{q})^T - (P_2\mathbf{q})^T]$, where P_1 and P_2 project their arguments to the second and first coordinates, respectively (i.e. $P_1\mathbf{q} = \theta$ and $P_2\mathbf{q} = \dot\theta$). Let $\hat{\mathbf{\Phi}}(\mathbf{q},\dot{\mathbf{q}}) = \hat{\mathbf{A}}(\mathbf{q})\dot{\mathbf{q}} + \hat{\mathbf{b}}(\mathbf{q})$, and let the 'feedforward' controller be the differencing controller of Theorem 2.4:

$$\begin{aligned}\mathbf{u}_f(\mathbf{q},\dot{\mathbf{q}}) &= \mathbf{u}_f(\theta,\dot\theta,\ddot\theta)\\ &= \hat{\mathbf{\Phi}}(\mathbf{q},\mathbf{v}(\mathbf{q})) - \hat{\mathbf{\Phi}}(\mathbf{q},\dot{\mathbf{q}}).\end{aligned}$$

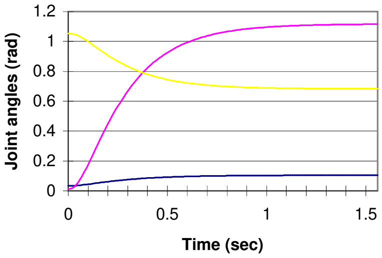
Additionally,

$$\begin{aligned}\hat{\mathbf{\Phi}}(\mathbf{q},\mathbf{v}(\mathbf{q})) &= \hat{\mathbf{A}}(\mathbf{q})[\mathbf{s}(\theta),\ \mathbf{a}(\theta,\dot\theta)] + \hat{\mathbf{V}}(\theta,\dot\theta) - \hat{k}\dot\theta\\ &= \hat{\mathbf{M}}(\theta)\mathbf{a}(\theta,\dot\theta) + \hat{\mathbf{V}}(\theta,\dot\theta) - \hat{k}\dot\theta.\end{aligned}$$

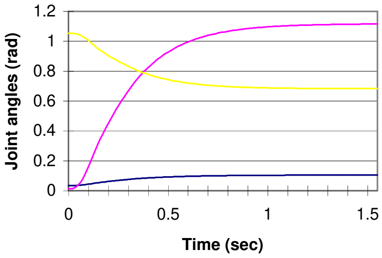
So if we let $\hat{\mathbf{\Psi}}(\theta,\dot\theta,\ddot\theta) = \hat{\mathbf{M}}(\theta)\ddot\theta + \hat{\mathbf{V}}(\theta,\dot\theta) - \hat{k}\dot\theta$ then $\hat{\mathbf{\Phi}}(\mathbf{q},\mathbf{v}(\mathbf{q})) = \hat{\mathbf{\Psi}}(\theta,\dot\theta,\mathbf{a}(\theta,\dot\theta))$ and $\hat{\mathbf{\Phi}}(\mathbf{q},\dot{\mathbf{q}}) = \hat{\mathbf{\Psi}}(\theta,\dot\theta,\ddot\theta)$. Note that here the domain D is equal to $K^3 \times \mathbf{R}^3$, where K is the unit circle. Conditions *1* and *2* of Theorem 2.4 are equivalent to the conditions that $\hat{\mathbf{M}}^T\mathbf{M} > 0, \hat{\mathbf{M}}^T\hat{\mathbf{M}} > 0$, and $\mathbf{M}^T\mathbf{M} > 0$ all hold uniformly over D, and that $(\mathrm{d}/\mathrm{d}\mathbf{q})\mathbf{M}$ and $(\mathrm{d}/\mathrm{d}\mathbf{q})\mathbf{b}$ are bounded over D. These latter conditions depend only on the dynamics of the robot arm and are met. The positivity conditions are also met, e.g. when the load changes [1].

It should be mentioned that one can use the 'simplified inverse-dynamics' $\hat{\mathbf{\Psi}}_0(\theta,\ddot\theta) = \hat{\mathbf{\Psi}}(\theta,0,\ddot\theta)$ because letting $\dot\theta = 0$ in $\hat{\mathbf{\Psi}}$ corresponds to an additive perturbation (since $\mathbf{M}$ does not depend on $\dot\theta$) and this can be compensated for without any additional requirements. The simplified inverse-dynamics is important for the case of learning schemes utilizing local approximators, as it means that only the configuration space (the 3-d space for the 3-joint robotic arm) and not the tangent bundle of the configuration space (a 6-d space) need be discretized. This is an important property of the control scheme since the number of discretization units scales with the dimension in the exponent.

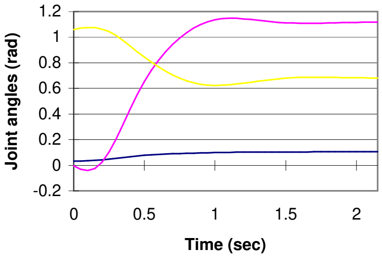
The exact forms of $\mathbf{M}$ and $\mathbf{V}$ used in the computer experiments are taken again from the benchmark problem section [37] of the book edited by Miller et al. [38], the corresponding equations being detailed in the Appendix. The equations were sampled at a rate $dt = 0.01$s using Euler's method and the control time interval was varied. In one set of experiments we used a high-frequency control ($dt = 0.01$s), while in another set a lower sampling rate was tried ($dt = 0.05$s). It was evident that the Euler method with the given sampling rate limits the range of the control torques, since otherwise one may loose numerical stability.



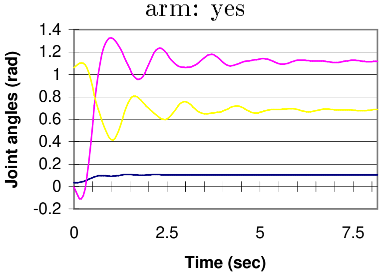
Sampling: 0.01 s, perturbed arm: no



Sampling: 0.01 s, perturbed arm: yes



Sampling: 0.05 s, perturbed arm: no



Sampling: 0.05 s, perturbed arm: yes

Figure 7: Control of a robotic manipulator using the SDS scheme
In the upper (lower) row the results with frequent (slow) samplings are shown, whereas in the left (right) column results with unperturbed (perturbed) arm are presented. The subfigures show the joint-angles as a function of time. For details of the experiment see the text.

In the trials we made use of the following values: $J = 0.5$ kg m^2, $M_1 = 10$ kg, $M_2 = 10$ kg, $L_1 = 0.6$ m, $L_2 = 0.8$ m, $k = 20$ kg m^2/s, $g = 9.81$ m/s^2 (see Fig. 6).

We considered point-to-point tasks with the desired speed-field $\mathbf{s}(\theta) = \lambda_p(\theta_0 - \theta)$, the acceleration field $\mathbf{a}(\theta, \dot{\theta}) = \mathbf{s}(\theta) - \dot{\theta}$, and control equations

$$
\begin{aligned}
\mathbf{v} &= \hat{\mathbf{\Psi}}_0(\theta, \mathbf{a}(\theta, \ddot{\theta})) - \hat{\mathbf{\Psi}}_0(\theta, \ddot{\theta}) \\
\mathbf{u} &= \lambda_f \mathbf{v} + \mathbf{w} \\
\dot{\mathbf{w}} &= \Lambda \mathbf{v}.
\end{aligned}
$$

In the latter three the following set of values were used: $\lambda_f = \Lambda = 0.3$, and $\lambda_p = 8.0$. Basically, λ_p governs the speed of motion, and its value was chosen so that the desired point can be reached withing 2-4 seconds under normal conditions.

The non-adaptive computed torque method with desired trajectory $\theta_d = \theta_0$, $\dot{\theta}_d = 0$ and $\ddot{\theta}_d = 0$ was tried too. The corresponding equations which govern the dynamics can be given by

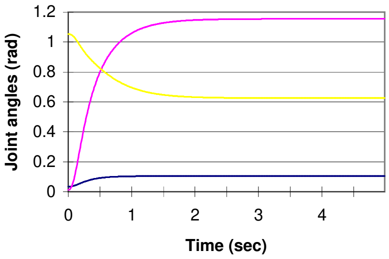
$$\begin{aligned}\mathbf{u} &= \lambda_f \hat{\mathbf{\Psi}}_0(\theta, \ddot{\theta}_d + k_v(\dot{\theta}_d - \dot{\theta}) + k_p(\theta_d - \theta)) \\ &= \lambda_f \hat{\mathbf{\Psi}}_0(\theta, \mathbf{a}(\theta, \ddot{\theta}))\end{aligned}$$

In the trials we set $k_v = 1$ and $k_p = \lambda_p$. In addition λ_f was chosen to be smaller than one used to model an underestimated inverse-dynamics.

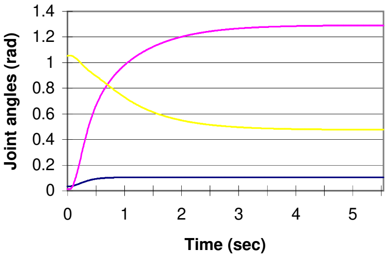
Now typical results of eight different experiments are presented. In the trial the applied control method (SDS, computed torque), the perturbation of the plant (no perturbation, perturbation with $M_1 = M_2 = 20$ kg, $J = 0.75$ kg m^2 and $k = 30$ kg m^2/s), and the sampling rate of control (high frequency sampling with $dt = 0.01$ s and low frequency sampling with $dt = 0.05$ s) were attempted. The start state was $\theta(0) = (+0.0332, +0.0125, +1.0529)$ with $\dot{\theta} = \ddot{\theta} = \mathbf{0}$, while the desired end-point was $(+0.1044, +1.1156, +0.6828)$. Note that the angle of the robot-base axis needed to be changed only slightly. The experiments were halted when the sum of the positional and speed errors was smaller than 0.05.

Fig. 7 shows the results for the SDS scheme. It can be seen that in the high frequency case the perturbation had almost no effect on the working. It was observed that in the initial phase it was the feedforward part which governed the plant, while later the role of the feedback part supressed that of the feedforward part. When sampling was slow and the perturbation was significant (the mass of the robot was doubled) damped oscillations sometimes arose. Increasing the gain of control was able to help this a little, but in practice it is more expedient to choose lower gains to minimize energy consumption.

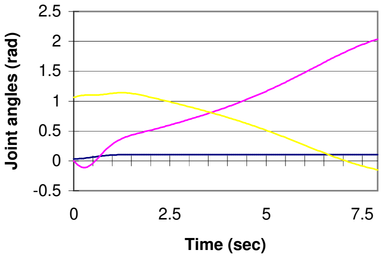
Fig. 8 shows the analogous results for the computed torque scheme. The computed torque method seemed to be more sensitive to the sampling rate: in the low-frequency case the controller was unable to move the robotic manipulator to the desired end-point. Even in the perfect model and for high frequency sampling the final error of the computed torque method was considerably larger than that for the SDS controller. (In the case of the computed torque method $\theta_{\mathbf{final}} = (+0.1045, +1.1547, +0.6254)$, while for the SDS controller $\theta_{\mathbf{final}} = (+0.1044, +1.1156, +0.6828) = \theta_d$.) Of course, in general the dynamic part of the SDS Controller makes the controller more precise. Besides this note that the inverse-dynamics model should explicitly contain the sampling rate [37] higher frequency sampling corresponding to a stronger controller. This in turn makes the computed torque controller somewhat weak in the case of low frequency sampling. The SDS Controller did not suffer from



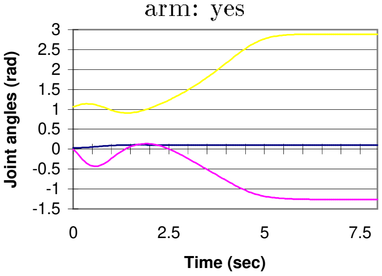
Sampling: 0.01 s, perturbed arm: no



Sampling: 0.01 s, perturbed arm: yes



Sampling: 0.05 s, perturbed arm: no



Sampling: 0.05 s, perturbed arm: yes

Figure 8: Control of a robotic manipulator with the non-adaptive computed torque method In the upper (lower) row the results with the frequent (slow) sampling are shown, while in the left (right) column the results with the unperturbed (perturbed) arm are presented. The subfigures show the joint-angles as a function of time. For details of the experiment see the text.

this problem since the dynamic part could build up the appropriate control torque.

4 An Artificial Neural Network Implementation

A neural network architecture capable of planning speed-fields and approximating inverse-dynamics in a unified way is shown in Fig. 9. Here we describe the working mechanism suited to path planning problems [29]. This particular architecture was suggested earlier by the authors [39,40].

The layers of the architecture are called the sensory layer, the geometry discretizing layer, the interneural layer, and the control layer. The layers are connected as follows. Spatially-tuned feedforward connections bring the sensory information to the geometry discretizing layer. The sensory neurons

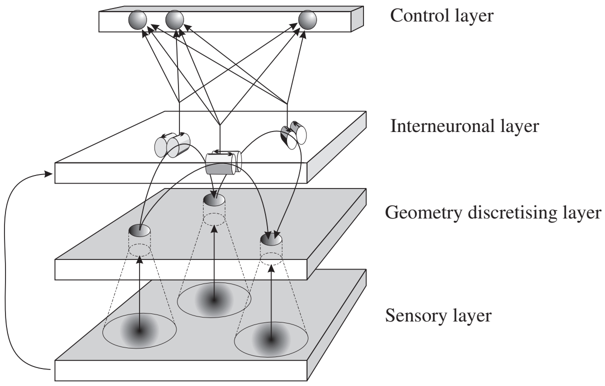


Figure 9: Architecture of the neural network

provide the input to the network, and they may be thought of as discretizing the state-space of the plant. The neurons of the geometry-discretizing layer should develop a problem-dependent discretization of the state-space, so the weights of these neurons may be developed in a self-organizing process such as the winner-takes-all mechanism [41,42]. The path-planning problem is given in terms of discretization point occupancies, the discretization points being identified with the neurons of the geometry discretizing layers. Any discretization point can be occupied by an obstacle, the plant, or the target. It is also possible that more than one discretization point might be occupied by an object. This results in a coarse coded, distributed representation of the object that in turn will result in smoother control signals. It is of course assumed that sensory inputs corresponding to start, target and obstacle entities are recognized by some higher order system.

The geometry-discretizing layer has recurrent intralayer *geometrical* connections that connect neighbouring nodes, whose connections can be learnt in a self-organized way, too [41,42]. Here the path-planning algorithm might be implemented as follows. The laterally-oriented geometrical connections between neighbouring discretization points allow activation to spread: when the activity-spreading on the discretization system settles down 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 which, as a vector-field, can be identified with the speed-field sought. (We consider the neural net as a numerical approximation of a continuous system. If the concepts are used with care one can then talk about the gradient field in the discretized system, i.e. an approximation of the corresponding quantity in the continuous system.) The activation spreading equation that forms this activity map can be of the *diffusion type*[43,44,45,46], when the equilibrium field only has one minimum and one maximum. As a matter of fact, these extremes correspond to the position of the plant and the goal, respectively. All this is achieved by allowing a unit inflow at the position of the plant and a unit outflow at the position of the goal. Any obstacles may be avoided by setting up appropriate boundary conditions, like forbidding the activity to spread along the lateral connections of the discretization points (neurons) occupied by obstacles, thus approximating to 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 if either the controller or 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.

Now let us return to the steady-state diffusion field case. The diffusing activities are sensed by the interneurons. The speed-field is represented by interneural activities which are summed up to give the estimated control response. The interneurons are simple linear I/O units that serve as sensors and also as the starting point of associative feedforward connections that point to control neurons. The free-space learnt control-connections allow the plant to move along the designed speed-field, if the control connections belonging to interneurons of the start-node geometry connections are controlling the motion. Obstacle avoidance without further training is really a natural consequence of the structure. The network can be made fully self-organizing with Hebbian-learning [41,42], the only exception being the aforementioned higher-order recognition module that initializes the formation of the steady-state geometry-discretizing-layer activities. The interneuron-to-control neuron connections (the control connections) form the direct, associative identification of the inverse-dynamics in the form of a 'position-direction to action' (PDA) map, its learning properties being scrutinized elsewhere [47]. Another benefit of this architecture is that the number of control-units scales linearly with the number of discretization units, and the discretized space is not the tangent bundle of the configuration space, but rather the configuration space itself. The speed-field tracking formulation, the freedom to use a simplified inverse-dynamics, and the interneuron concept all contribute to an overall simplification of things.

5 Biological relevance

The SDS scheme has been suggested as a suitable candidate for constructing a model of higher-order motor functions of the basal ganglia – thalamocortical loops [48], since several special properties of the SDS scheme are highlighted when dealing with these motor areas. The case of identical feedforward and feedback controllers makes extensive use of differencing and allows one to use the simplified inverse-dynamics. The SDS model of motor control requires the existence of cortical neurons that compute the desired speed vector $\mathbf{v}(\mathbf{q})$, an estimate of the desired control vector $\hat{\mathbf{\Phi}}(\mathbf{q}, \mathbf{v})$, the experienced (measured) speed vector $\dot{\mathbf{q}}$, an estimate of the "experienced control vector" $\hat{\mathbf{\Phi}}(\mathbf{q}, \dot{\mathbf{q}})$, and the appropriate differences.

The SDS model of basal ganglia – thalamocortical loops identifies these differencings between desired and experienced channels with the functional consistency found in the basal ganglia [48]. According to Alexander, the various projection to the external and the internal segments of the globus pallidus are similar in functionality, that can be seen via the activations of the medium spiny neurons of the putamen, the targets of cortical afferents arising from the motor areas. An activation of striated medium spiny neurons associated with the different arms of the indirect pathway will tend to increase the output of the basal ganglia. In contrast, an activation of medium spiny neurons associated with the direct pathway tends to decrease the output of the basal ganglia. The net result is that a cortically-initiated activation of the direct (indirect) pathway will tend to enhance (suppress) reentrant thalamocortical excitation by a decreased (increased) inhibitory outflow from the basal ganglia to the thalamus [49].

It is reasonable to assume that information about the position and about the experienced and desired directions have been formulated somewhere, possibly outside of the basal ganglia – thalamocortical loops and that this information forms the basis of computation in these loops. The basal ganglia then performs the differencing: The desired acceleration may be expressed as the difference between the desired direction and the experienced direction. The feedforward control vector can be computed by means of subtracting the estimate of the "experienced control vector" from the estimate of the desired control vector. The feedback control vector one notes is the time-integrated value of the feedforward control vector. Using these properties of the SDS Feedback scheme it is possible to characterize the model neurons of these control areas: model neurons representing the desired acceleration will have higher firing rates in the preparatory phase, whereas the firing rates of model neurons representing experienced direction are motion related. The model neurons that represent the desired direction have mixed characteristics. Motion related model neuronal activities too may be divided into directional and muscle-like categories, the former one corresponding to the experienced direction and the latter one expressing components of the control vector. The consequences on the classification of model neurons are in general agreement with the experimental classification [50,51].

Two other features of the SDS scheme, which must be taken into account when considering an implementation, are: (1) The sign of the differencing related to the desired and experienced estimates of the true inverse-dynamics function is subject to particular task settings (consider, say the problem of mirror writing when the trajectory to be followed by the pen is watched through a mirror); (2) The feedback channel should be sign proper in all of the components. Both constraints involve separate feedback channels and recognition-based channel selection for sign proper results. The finding that the basal ganglia – thalamocortical loops are organized in distinct parallel pathways [52] is then suggested as the division of the "task-space" into subsets with sign-proper feedback channels [48]. The division of the task-space and computation of the acceleration field requires separate neuronal subsets that compute task related and limb related components for both preparatory, motion related and mixed activities. This is yet another feature of neurons belonging to these loops [53].

The feedback control vector should be computed by means of some integration procedure. The model predictions concerning this integration are not restrictive. But it is reasonable to assume that the biological system performs some form of leaky integration. One hypothetical possibility about this 'integration' is that it results from a neuron subset receiving a thalamocortical recurrent excitation that in turn integrates the control components of the feedback channel. Another might be that differencing and integration are interchanged and a subset of neurons integrate before differencing. Yet another possibility is suggested from noting that the feedforward and feedback controllers are identical, and the summation of the two control vectors can be viewed as a weighted integration (i.e. a memory function with non-exponential time dependency) in the feedback channel. That is, the basal ganglia – thalamocortical loops of higher order motor functions might be organized as feedback loops without any feedforward channel. In this case neurons with different 'integrate and fire' time constants could represent any weighted sum. Computations without making use of the feedforward channel show minor differences only at the start of the motion, indicating a viability even for the case of an exponential memory function. Considering that integration and differencing can be interchanged the concept of weighted memory function seems to accord well with the fairly recent findings that (1) activities in the supplementary motor area, the motor cortex and the putamen strongly overlap, and (2) the activities in the cortical regions precede the activity in the putamen [50,51,53]. The results concerning the robustness of the SDS scheme against inaccuracies in the differencing procedure, plausible connections with the major diseases of the basal ganglia as well as elaborations on other close similarities between the basal ganglia – thalamocortical loops and the SDS scheme can be found elsewhere [48,28].

6 Summary and further discussion

The 'Static and Dynamic' State (SDS) feedback control scheme [1,30] of ours was presented. Then slight modifications of the original control equations led to a global stability property which could be proved in a rigorous manner [1,28]. The main advantage of this new scheme is that within the confines of the stability theorem the parameter adaptation scheme can be chosen arbitrarily, allowing the use of non-linearly parametrized function approximators such as neural networks. One restriction of SDS Control is that the results hold only for bounded plants, although it was shown in computer trials that the SDS Control still seems quite suitable for non-bounded plants such as realistic robotic manipulators. One outstanding question is whether SDS Control can be used to control plants with non-trivial zero-dynamics. So far the results obtained were for only the case when there is no zero-dynamics at all, but it does seem to be able to extend the theory to cover plants which are globally minimum-phase.

Two sets of simulations were presented, that of the control of a bioreactor and of a robotic manipulator. The experiments bore out the predicted theoretical results, i.e. that the SDS Controller is capable of compensating for large perturbations. It was observed too that SDS Control is sensitive to the sampling rate of control, but it is less sensitive than the static (non-adaptive) computed torque method.

It was argued that since the SDS scheme allows one to choose and modify the functional form of the speed-field to be tracked, the estimation of the inverse-dynamics, and the particular adaption mechanism, one must have an attractive option for optimization according to definite external objectives like minimization of energy consumption and the mean square tracking error.

It has been argued elsewhere [48] that the architecture, the workings and the stability conditions have a lot in common with basal ganglia – thalamocortical loops. The higher order formulation of the SDS scheme allowed us to interpret the local approximators of the scheme as the models of the neuronal groups of the supplementary motor area, the motor cortex, and the putamen. The question arises of what extent one can view the SDS scheme as a viable functional model of these puzzling areas of the brain.

Acknowledgements

We are grateful to Prof. András Krámli for his invaluable comments and suggestions. This work was partially funded by OTKA grants T017110, T014330, T014566, F20132, and the US-Hungarian Joint Fund Grant 168/91-A 519/95-A.

Appendix

The exact forms of $\mathbf{M}$ and $\mathbf{V}$ used in the computer experiments are taken from those of a benchmark problem [37] and are detailed below with conventions and the notations closely following the cited work. (The numerical parameters are given in Section 3.2)

Let

$$\begin{aligned}
a_1 &= (M_1+M_2)L_1^2\cos^2(\theta_2) + M_2L_1L_2\cos(\theta_2)\cos(\theta_3) + M_2L_2^2\cos^2(\theta_3) + J \\
a_2 &= (M_1+M_2)L_1^2 \\
a_3 &= M_2L_1L_2\sin(\theta_2+\theta_3) \\
a_4 &= M_2L_2^2
\end{aligned}$$

and

$$H = (a_2a_4 - a_3^2).$$

Then

$$\mathbf{M}(\theta) = \begin{pmatrix} a_1 & 0 & 0 \\ 0 & H(a_4-a_3) & H(a_4-a_3) \\ 0 & H(a_4-a_3)\,a_3/a_2 & H/a_2\big((a_4-a_3)a_3+1\big) \end{pmatrix}.$$

Further, let

$$\begin{aligned}
b_1 &= \big(2(M_1+M_2)L_1^2 + M_2L_1L_2\big)\sin(\theta_2)\cos(\theta_2) \\
b_2 &= M_2L_1L_2\cos(\theta_2)\sin(\theta_3) + 2M_2L_2^2\sin(\theta_3)\cos(\theta_3) \\
b_3 &= M_2L_1L_2\sin(\theta_3-\theta_2) \\
b_4 &= (M_1-M_2)L_1^2\sin(\theta_2)\cos(\theta_3) + M_2L_1L_2\sin(\theta_2)\cos(\theta_3) \\
b_5 &= M_2L_2^2\sin(\theta_3)\cos(\theta_3) + M_2L_1L_2\cos(\theta_2)\sin(\theta_3).
\end{aligned}$$

Then

$$V(\theta,\dot\theta)=\begin{pmatrix} b_1\dot\theta_1\dot\theta_2+b_2\dot\theta_1\dot\theta_3 \\ -b_3\dot\theta_3^2-b_4\dot\theta_1^2-(M_1+M_2)L_1\cos(\theta_2)g \\ b_3\dot\theta_2^2-b_5\dot\theta_1^2-M_2L_2\cos(\theta_3)g \end{pmatrix}.$$

References

[1] Cs. Szepesvári, Sz. Cimmer, and A. Lőrincz. Dynamic state feedback neurocontroller for compensatory control. *Neural Networks*, 1997. (accepted).

[2] R. Berber. *Methods of Model Based Process Control.* Volume 293 of *NATO ASI Series E: Applied Sciences*, Kluwer Academic Publisher, 1995.

[3] D.Q. Mayne. Optimization in model predictive control. In R. Berber, editor, *Methods of Model Based Process Control*, pages 367–396, Kluwer Academic Publisher, 1995.

[4] D.W. Clarke, E. Mosca, and R. Scattaloni. Robustness of an adaptive predictive controller. In *Proc. of the 30th IEEE Conference on Decision and Control, Brighton*, pages 979–984, IEEE Press, 1991.

[5] T.H. Yang and E. Polak. Moving horizon control of nonlinear systems with input saturation, disturbances and plant unvertainty. *International Journal of Control*, 58:875–903, 1993.

[6] D.Q. Mayne and H. Michalska. Adaptive control of linear constained systems with inaccessible states. In *Proc. American Control Conference, Boston*, 1994.

[7] D.Q. Mayne and H. Michalska. Adaptive receding horizon control for constrained nonlinear systems. In *Proc. of the 32nd IEEE Conference on Decision and Control, San Antonio*, pages 1286–1291, IEEE Press, 1995.

[8] R. Bellman. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey, 1957.

[9] J. Walter, T. Martinetz, and K. Schulten. Industrial robot learns visuomotor coordination by means of "neural-gas" network. In *Proc. of ICANN*, pages 357–364, Elsevier, Amsterdam, 1991.

[10] P. van der Smagt, F. Groen, and F. van het Groenewoud. The locally linear nested network for robot manipulation. In *Proc. of the IEEE ICNN'94*, pages 2787–2792, IEEE Press, Orlando, Florida, May 1994.

[11] M.I. Jordan. Learning and the degrees of freedom problem. In *Attention and Performance, XIII.*, NJ: Erlbaum, Hillsdale, 1990.

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

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

[14] F.L. Lewis, C.T. Abdallah, and D.M. Dawson. *Control of Robot Manipulators.* MacMillan, New York, 1993.

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

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

[17] A. Isidori. *Nonlinear Control Systems.* Springer-Verlag, Berlin, 1989.

[18] S. Sastry and A. Isidori. Adaptive control of linearizable systems. *IEEE Trans. on Automatic Control*, 34(11):1123–1131, 1989.

[19] C. Byrnes and A. Isidori. Asymptotic stabilization of minimum phase nonlinear systems. *IEEE Trans. on Automatic Control*, 36(10):1122–1137, 1991.

[20] J. Craig, P. Hsu, and S. Sastry. Adaptive control of mechanical manipulators. *Int. J. of Robotic Research*, 6(2):16–28, 1987.

[21] H. Gomi and M. Kawato. Neural network control for a closed-loop system using Feedback-Error-Learning. *Neural Networks*, 6(7):933–946, 1993.

[22] T. Yabuta and T. Yamada. Learning control using neural networks. In *Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation*, pages 740–745, IEEE Press, Sacramento, California, 1991.

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

[24] S. Yoshida and N. Wakabayashi. A fuzzy logic controller for a rigid disk drive. *IEEE Control Systems Magazine*, 65–70, June 1992.

[25] V. Fomin, A. Fradkov, and V. Yakubovich, editors. *Adaptive control of dynamical systems.* Nauka, Moscow, 1981. (in Russian).

[26] A. Barron. Approximation and esimation bounds for artificial neural networks. In *4th Annual Workshop on Computational Learning Theory*, pages 243–249, 1991.

[27] B. Delyon, A. Juditsky, and A. Benveniste. Accuracy analysis for wavelet approximations. *IEEE Trans. on Neural Networks*, 6(2):332–348, 1995.

[28] Cs. Szepesvári, Sz. Cimmer, and A. Lőrincz. Static and dynamic state feedback control of higher order plants. *Biological Cybernetics*, 1997. (submitted).

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

[30] Cs. Szepesvári and A. Lőrincz. Neurocontrol II: High precision control achieved using approximate inverse dynamics models. *Neural Network World*, 897–920, 1996.

[31] A. Ben-Israel and T.N.E. Greville. *Generalized Inverses: Theory and Applications. Pure and Applied Mathematics, Wiley-Interscience*, J. Wiley & Sons, New York, 1974.

[32] S. Sastry and M. Bodson. *Adaptive Control - Stability, Convergence and Robustness.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[33] P. Hsu, M. Bodson, S. Sastry, and B. Paden. Adaptive identification and control of manipulators without using joint accelerations. In *Proc. of the IEEE Conf. on Robotics and Automation*, pages 1210–1215, IEEE Press, Relaigh, North Carolina, 1987.

[34] P. Agrawal, C. Lee, H.C. Lim, and D. Ramkrishna. Theoretical investigations of dynamic behavior of isothermal continuous stirred tank biological reactors. *Chemical Engineering Science*, 37, 1982.

[35] L.H. Ungar. A bioreactor benchmark for adaptive network-based process control. In *Neural Networks for Control*, pages 387–402, MIT Press, Cambridge, 1992.

[36] D.D. Brengel and W.D. Seider. A multi-step nonlinear predictive controller. *Ind. Eng. Chem. Res.*, 28:1812–1822, 1989.

[37] C.W. Anderson and W.T. Miller, III. Challenging control problems. In *Neural Networks for Control*, pages 475–510, MIT Press, Cambridge, 1992.

[38] W.T. III Miller, R.S. Sutton, and P.J. Werbos, editors. *Neural Networks for Control.* MIT Press, Cambridge, Massachusetts, 1990.

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

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

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

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

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

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

[45] R. Glasius, A. Komoda, and S. Gielen. Neural network dynamics for path planning and obstacle avoidance. *Neural Networks*, 1994.

[46] G.F. Marshall and L. Tarassenko. Robot path planning using vlsi resistive grids. In *IEEE Proc., Vision, Image and Signal Processing*, pages 267–272, 1994.

[47] Cs. Szepesvári and A. Lőrincz. Neurocontrol I: Self-organizing speed-field tracking. *Neural Network World*, 875–896, 1996.

[48] A. Lőrincz. Neurocontrol III: temporal differencing schemes of the basal ganglia-thalamocortical loops. *Neural Network World*, 1996. in press.

[49] G. E. Alexander. Basal ganglia. In *The Handbook of Brain Theory and Neural Networks*, pages 139–144, Bradford Books / MIT Press, Cambridge, MA, 1995.

[50] G. E. Alexander and M. D. Crutcher. Preparation for movement: neural representations of intended direction in three motor areas of the monkey. *Journal of Neurophysiology*, 64:133–150, 1990.

[51] M. D. Crutcher and G. E. Alexander. Movement-related neuronal activity selectively coding either direction or muscle pattern in three motor areas of the monkey. *Journal of Neurophysiology*, 64:151–163, 1990.

[52] J. E. Hoover and P. L. Strick. Multiple output channels in the basal ganglia. *Science*, 259:819–821, 1993.

[53] G. E. Alexander and M. D. Crutcher. Neural representations of the target (goal) of visually guided arm movements in three motor areas of the monkey. *Journal of Neurophysiology*, 64:164–178, 1990.