Optimal Control by Least Squares Support Vector Machines *

J.A.K. Suykens, J. Vandewalle and B. De Moor

Katholieke Universiteit Leuven

Department of Electrical Engineering, ESAT-SISTA

Kardinaal Mercierlaan 94, B-3001 Leuven (Heverlee), Belgium

Tel: 32/16/32 18 02 Fax: 32/16/32 19 70

Email: johan.suykens@esat.kuleuven.ac.be

Corresponding author: Johan Suykens

Running title: Optimal Control by LS-SVM's

Neural Networks 14, pp.23-35 (2001)

*This research work was carried out at the ESAT laboratory and the Interdisciplinary Center of Neural Networks ICNN of the Katholieke Universiteit Leuven, in the framework of the FWO project G.0262.97 *Learning and Optimization: an Interdisciplinary Approach*, the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture (IUAP P4-02 & IUAP P4-24) and the Concerted Action Project MIPS and MEFISTO-666 of the Flemish Community. Johan Suykens is a postdoctoral researcher with the National Fund for Scientific Research FWO - Flanders.

Abstract

Support vector machines have been very successful in pattern recognition and function estimation problems. In this paper we introduce the use of least squares support vector machines (LS-SVM's) for the optimal control of nonlinear systems. Linear and neural full static state feedback controllers are considered. The problem is formulated in such a way that it incorporates the N-stage optimal control problem as well as a least squares support vector machine approach for mapping the state space into the action space. The solution is characterized by a set of nonlinear equations. An alternative formulation as a constrained nonlinear optimization problem in less unknowns is given, together with a method for imposing local stability in the LS-SVM control scheme. The results are discussed for support vector machines with radial basis function kernel. Advantages of LS-SVM control are that no number of hidden units has to be determined for the controller and that no centers have to be specified for the Gaussian kernels when applying Mercer's condition. The curse of dimensionality is avoided in comparison with defining a regular grid for the centers in classical radial basis function networks. This is at the expense of taking the trajectory of state variables as additional unknowns in the optimization problem, while classical neural network approaches typically lead to parametric optimization problems. In the SVM methodology the number of unknowns equals the number of training data, while in the primal space the number of unknowns can be infinite dimensional. The method is illustrated both on stabilization and tracking problems including examples on swinging up an inverted pendulum with local stabilization at the endpoint and a tracking problem for a ball and beam system.

Keywords. Neural optimal control, support vector machines, radial basis functions.

1 Introduction

Since the introduction of backpropagation through time (Werbos, 1990) and dynamic backpropagation (Narendra & Parthasarathy, 1991), the area of neural optimal control has been focusing on gradient based approaches or backpropagation type solutions for dynamical systems involving neural networks. Among the many modelbased approaches, the N-stage optimal control problem (Bryson & Ho, 1969) has been studied in combination with several structures for the neural controller. For example, Nguyen & Widrow (1990) applied backpropagation to the problem of backing up a trailer-truck based on a neural network emulator for the plant. In Saerens *et al.* (1993) a full static state feedback controller has been studied in the context of optimal control, thereby relating the Lagrange multiplier sequence to the backpropagation algorithm. Parisini & Zoppoli (1994) assumed a linear structure preserving principle for the state tracking problem with application to control of a space robot. Plumer (1996) studied the case of optimal final time. In Suykens *et al.* (1994)(1996) the problem of swinging up of an inverted pendulum and double inverted pendulum with stabilization at the endpoint has been formulated as a parametric optimization problem in the unknown weights of feedforward or recurrent neural controllers.

In this paper we discuss N-stage optimal control by least squares support vector machines (LS-SVM's). Support vector machines have been recently introduced for solving pattern recognition and function estimation problems (Vapnik, 1995, 1998a; Schölkopf *et al.*, 1999). In this method one maps the data into a higher dimensional input space and one constructs an optimal separating hyperplane in this space. Hereby one exploits Mercer's theorem which avoids an explicit formulation of this nonlinear mapping. The solution is written as a weighted sum of the data points. In the original formulation one solves a quadratic programming problem, which yields many zero weights. The data points corresponding to the non-zero weights are called support vectors. Kernel function parameters can be chosen such that a bound on the generalization error is minimized, expressed in terms of the VC dimension. One has the possibility to use polynomials, splines, radial basis function networks or multilayer perceptrons as kernels. Although one preferably applies Mercer's condition, this is not absolutely necessary as is shown e.g. in Suykens &

Vandewalle (1999a), where standard multilayer perceptron classifiers with a fixed number of hidden units are trained using a SVM methodology. Being based on the structural risk minimization principle and capacity concept with pure combinatorial definitions, the quality and complexity of the SVM solution does not depend directly on the dimensionality of the input space.

In the SVM control method we make use of a least squares version of support vector machines. Originally, Vapnik's epsilon insensitive loss function has been employed for the function estimation problem. A least squares interpretation on the other hand has been given by Saunders *et al.* (1998) for function estimation problems, Suykens & Vandewalle (1999b) for classification, and Müller *et al.* (1997) for time-series prediction. In this case, the problem formulation involves equality instead of inequality constraints and the support values are proportional to the errors at the data points, which simplifies the problem. Sparseness gets lost when the function estimation corresponds to this form of ridge regression, but this can be imposed afterwards by doing pruning based upon the support value spectrum (Suykens *et al.*, 2000a, 2000b).

In the optimal control method by LS-SVM's, the N-stage optimal control problem and the optimization problem related to the LS-SVM controller are incorporated within one problem formulation. A model is assumed to be available for the plant, the controller is designed based upon this model and afterwards applied to the plant, assuming that the certainty equivalence principle holds (similar to Nguyen & Widrow's emulator approach). The solution is characterized by a set of nonlinear equations. However, the set of nonlinear equations will typically contain a large number of unknowns. Therefore an alternative formulation for LS-SVM control is given in less unknowns. This formulation also enables to incorporate local stability constraints. A main difference between standard neural network approaches (MLP, RBF) and LS-SVM control is that in the former one solves a parametric optimization problem in the unknown interconnection weights while in the latter also the state vector sequence is part of the unknown parameter vector in the optimization problem. However, standard methodologies suffer from problems like the choice of the number of hidden units needed in order to accomplish a given control task. More specifically, in the case of RBF networks one has a curse of dimensionality when one defines a regular grid for the centers (hidden units) in state space, as explained e.g. in the method of Sanner & Slotine (1992). In the LS-SVM control case the centers will follow from the optimal trajectory that one seeks. Furthermore, standard neural network approaches always work in a primal weight space, while in SVM methodologies the computations are done in a dual space such that the number of unknowns equals the number of training data points (and not the number of weights in the primal space, which can be infinite dimensional).

We illustrate the LS-SVM control method on a number of simulation examples including swinging up an inverted pendulum with local stabilization at the endpoint and a tracking problem for a ball and beam system. In Suykens *et al.* (1994) methods for realizing a transition between two states with local stabilization at the target point was discussed. This was illustrated on swinging up an inverted pendulum with local stabilization at the endpoint. For the full static state feedback case a Linear Quadratic Regulator (LQR) (Franklin *et al.*, 1990) was designed for balancing the pole in its upright position. This result was used in order to impose a set of constraints on the interconnection weights of a multilayer perceptron controller. A drawback of the approach was that the number of hidden units for the neural controller had to be chosen ad hoc. In this paper we present a solution by SVM control which takes into account an LQR design in the LS-SVM control with RBF kernel. The number of hidden units is determined here by the number N in the N-stage optimal control problem formulation. Instead of local linear LQR design one has also the possibility to apply robust linear control methods (Boyd & Barratt, 1991) such as H_∞ control and μ theory instead, in order to take into account uncertainties (parametric uncertainties, unmodelled dynamics) and noise.

For the example of a ball and beam system (Hauser *et al.*, 1992) a tracking problem with SVM control is discussed. Again a LQR design is incorporated in order to impose local stability of the origin for the autonomous closed-loop system. Imposing robust local stability for the closed-loop system has also been successful in the application of NLq neural control theory, introduced in Suykens *et al.* (1996)(1997), to the control of a real-life ball and beam system (Verrelst *et al.*, 1998). Within NLq theory sufficient conditions for global asymptotic stability and input/output stability with finite L_2-gain are available, which can also be employed in order to impose robust local stability of the origin for the closed-loop system. A similar approach is followed in the LS-SVM control for the ball and beam system, where robust local stability can be realized based upon the work of Sezer & Siljak (1992).

This paper is organized as follows. In Section 2 we formulate the N-stage optimal control problem. In Section 3 we review work on support vector machines. In Section 4 we discuss optimal control by least squares support vector machines. In Section 5 we discuss an alternative formulation together with stability constraints. In Section 6 we present examples.

2 The N-stage optimal control problem

In the N-stage optimal control problem one aims at solving the following problem (Bryson & Ho, 1969):

$$\min \mathcal{J}_N(x_k, u_k) = \rho(x_{N+1}) + \sum_{k=1}^{N} h(x_k, u_k) \tag{1}$$

subject to the system dynamics

$$x_{k+1} = f(x_k, u_k), \quad k = 1, ..., N \ (x_1 \text{ given}),$$

where $\rho(\cdot)$ and $h(\cdot, \cdot)$ are positive definite functions. A typical choice is the quadratic cost $h(x_k, u_k) = x_k^T Q x_k + u_k^T R u_k$, $\rho(x_{N+1}) = x_{N+1}^T Q x_{N+1}$ with $Q = Q^T > 0$, $R = R^T > 0$. The functions $\rho(\cdot)$, $h(\cdot, \cdot)$, $f(\cdot, \cdot)$ are assumed to be twice continuously differentiable. $x_k \in \mathbb{R}^n$ denotes the state vector, $u_k \in \mathbb{R}$ is the input of the system. The methods discussed in this paper are not restricted to single input systems.

In order to find the optimal control law, one constructs the Lagrangian

$$\mathcal{L}_N(x_k, u_k; \lambda_k) = \mathcal{J}_N(x_k, u_k) + \sum_{k=1}^{N} \lambda_k^T [x_{k+1} - f(x_k, u_k)] \tag{2}$$

with Lagrange multipliers $\lambda_k \in \mathbb{R}^n$. The conditions for optimality are given by (Fletcher, 1987)

$$
\left\{
\begin{array}{rcllr}
\frac{\partial \mathcal{L}_N}{\partial x_k} & = & \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k = 0, & k = 2, ..., N & \text{(adjoint equation)} \\
\frac{\partial \mathcal{L}_N}{\partial x_{N+1}} & = & \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 & & \text{(adjoint final condition)} \\
\frac{\partial \mathcal{L}_N}{\partial u_k} & = & \frac{\partial h}{\partial u_k} - \lambda_k^T \frac{\partial f}{\partial u_k} = 0, & k = 1, ..., N & \text{(variational condition)} \\
\frac{\partial \mathcal{L}_N}{\partial \lambda_k} & = & x_{k+1} - f(x_k, u_k) = 0, & k = 1, ..., N & \text{(system dynamics).}
\end{array}
\right. \tag{3}
$$

For the case of a quadratic cost function subject to linear system dynamics with infinite time horizon, the optimal control law can be represented by full static state feedback control. However, in general, the optimal control law cannot be represented by state feedback as the optimal control law may also depend on the co-state. Nevertheless, one is often interested in finding a suboptimal control strategy of this form. In the context of neural control, Saerens *et al.* (1993) considered, in addition to (1),

$$u_k = g(x_k) \tag{4}$$

where $g(\cdot)$ is parametrized by a neural network architecture, and discussed the link with the backpropagation algorithm. In this paper we will also consider a control law of the form (4), by relating (4) to support vector machines. Because SVM methodology is not a parametric modelling approach, this is less straightforward in comparison with standard neural networks such as MLP's and RBF's.

3 The support vector method of function estimation

In this Section we review basic ideas of the support vector method of function estimation, that are essential for its introduction within the neural control problem as will be considered in the sequel. For further details on SVM's we refer to Vapnik (1995)(1998a,b), Vapnik *et al.* (1997), Schölkopf *et al.* (1999), Smola *et al.* (1998)(1999), Cherkassky & Mulier (1998), Haykin (1994).

Consider regression in the following set of functions

$$\mathcal{F}(\mathcal{X}) = \mathcal{W}^T \varphi(\mathcal{X}) + \mathcal{B} \tag{5}$$

with given training data $\{\mathcal{X}_i, \mathcal{Y}_i\}_{i=1}^M$ where M denotes the number of training data, $\mathcal{X}_i \in \mathbb{R}^m$ are the input data and $\mathcal{Y}_i \in \mathbb{R}$ are the output data. The nonlinear mapping $\varphi : \mathbb{R}^m \rightarrow \mathbb{R}^{n_h}$ maps the input data into a so-called high dimensional feature space (which can be infinite dimensional) and $\mathcal{W} \in \mathbb{R}^{n_h}$, $\mathcal{B} \in \mathbb{R}$. In the support vector method one aims at minimizing the empirical risk

$$\mathcal{R}_{emp}(\mathcal{W}, \mathcal{B}) = \frac{1}{M} \sum_{i=1}^{M} |\, \mathcal{Y}_i - \mathcal{W}^T \varphi(\mathcal{X}_i) - \mathcal{B} \,|_\epsilon \tag{6}$$

subject to elements of the structure $\mathcal{S}_n$, defined by the inequality $\mathcal{W}^T\mathcal{W} \leq c_n$. The loss function employs Vapnik's ϵ-insensitive model:

$$|\, \mathcal{Y} - \mathcal{F}(\mathcal{X}) \,|_\epsilon = \begin{cases} 0, & \text{if } |\, \mathcal{Y} - \mathcal{F}(\mathcal{X}) \,| \leq \epsilon \\ |\, \mathcal{Y} - \mathcal{F}(\mathcal{X}) \,| - \epsilon, & \text{otherwise.} \end{cases} \tag{7}$$

The estimation problem is formulated then as the optimization problem:

$$\min_{\mathcal{W}, \mathcal{B}, \xi^*, \xi} \mathcal{J}_\epsilon(\mathcal{W}, \xi^*, \xi) = \frac{1}{2} \mathcal{W}^T \mathcal{W} + \gamma \,\{\sum_{i=1}^{M} \xi_i^* + \sum_{i=1}^{M} \xi_i\} \tag{8}$$

subject to the constraints

$$\begin{cases} \mathcal{Y}_i - \mathcal{W}^T \varphi(\mathcal{X}_i) - \mathcal{B} & \leq & \epsilon + \xi_i^*, & i = 1, .., M \\ -\mathcal{Y}_i + \mathcal{W}^T \varphi(\mathcal{X}_i) + \mathcal{B} & \leq & \epsilon + \xi_i, & i = 1, .., M \\ \xi_i^* & \geq & 0, & i = 1, .., M \\ \xi_i & \geq & 0, & i = 1, .., M \end{cases}$$

where ξ_i, ξ_i^* are slack variables and γ is a positive real constant. One obtains $\mathcal{W} = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) \varphi(\mathcal{X}_i)$ where α_i^*, α_i are obtained by solving a quadratic program and are the Lagrange multipliers related to the first and second set of constraints. The data points corresponding to nonzero values for $(\alpha_i^* - \alpha_i)$ are called support vectors. Typically, many of these values are equal to zero. Finally, one obtains the following model in the dual space

$$\mathcal{F}(\mathcal{X}) = \sum_{i=1}^{M} (\alpha_i^* - \alpha_i) K(\mathcal{X}_i, \mathcal{X}) \tag{9}$$

where the kernel function K corresponds to

$$K(\mathcal{X}_i, \mathcal{X}) = \varphi(\mathcal{X}_i)^T \varphi(\mathcal{X}) \tag{10}$$

according to Mercer's condition. One has several possibilities for the choice of this kernel function, including linear, polynomial, splines, RBF. In the sequel of this paper we will focus on RBF kernels.

The influence of the number of support vectors on the generalization performance has been theoretically studied in Vapnik (1995). An extension studied in the context of the ϵ-insensitive loss function is

$$\mathcal{J}_{\epsilon,p}(\mathcal{W}, \xi^*, \xi) = \frac{1}{2}\mathcal{W}^T\mathcal{W} + \gamma\,\{\sum_{i=1}^{M}(\xi_i^*)^p + \sum_{i=1}^{M}(\xi_i)^p\} \tag{11}$$

where $p = 1$ corresponds to (8). General convex cost functions have been investigated in Smola (1999). Furthermore, links between these loss functions and regularization theory have been studied in Smola *et al.* (1998) (1999).

For the sequel we employ a least squares version of the support vector method, which has been investigated by Saunders *et al.* (1998) and Suykens & Vandewalle (1998b), for function estimation and classification problems, respectively. It corresponds to $p = 2$ and the following form of ridge regression (Golub & Van Loan, 1989)

$$\min_{\mathcal{W},\mathcal{B},\xi} \mathcal{J}_{LS}(\mathcal{W}, \mathcal{B}, \xi) = \frac{1}{2}\mathcal{W}^T\mathcal{W} + \gamma\,\frac{1}{2}\sum_{i=1}^{M}\xi_i^2 \tag{12}$$

subject to the equality constraints

$$\mathcal{Y}_i = \mathcal{W}^T\varphi(\mathcal{X}_i) + \mathcal{B} + \xi_i,\ \ i = 1, ..., M.$$

One defines the Lagrangian

$$\mathcal{L}_{LS}(\mathcal{W}, \mathcal{B}, \xi; \alpha) = \mathcal{J}_{LS}(\mathcal{W}, \mathcal{B}, \xi) - \sum_{i=1}^{M}\alpha_i\,(\,\mathcal{W}^T\varphi(\mathcal{X}_i) + \mathcal{B} + \xi_i - \mathcal{Y}_i\,) \tag{13}$$

where α_i are Lagrange multipliers (which can be either positive or negative due to the equality constraints as follows from the Kuhn-Tucker conditions (Fletcher, 1987)). The conditions for optimality

$$\begin{cases} \frac{\partial \mathcal{L}_{LS}}{\partial \mathcal{W}} = 0 & \rightarrow \quad \mathcal{W} = \sum_{i=1}^{M} \alpha_i \varphi(\mathcal{X}_i) \\ \\ \frac{\partial \mathcal{L}_{LS}}{\partial \mathcal{B}} = 0 & \rightarrow \quad \sum_{i=1}^{M} \alpha_i = 0 \\ \\ \frac{\partial \mathcal{L}_{LS}}{\partial \xi_i} = 0 & \rightarrow \quad \alpha_i = \gamma \xi_i, \qquad\qquad\qquad\qquad i = 1, ..., M \\ \\ \frac{\partial \mathcal{L}_{LS}}{\partial \alpha_i} = 0 & \rightarrow \quad \mathcal{W}^T \varphi(\mathcal{X}_i) + \mathcal{B} + \xi_i - \mathcal{Y}_i = 0, \quad i = 1, ..., M \end{cases} \tag{14}$$

can be written as the solution to the following set of linear equations after elimination of $\mathcal{W}$ and ξ_i

$$\left[\begin{array}{c|c} 0 & \vec{1}^T \\ \hline \vec{1} & \Omega + \gamma^{-1} I \end{array} \right] \left[\begin{array}{c} \mathcal{B} \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline \mathcal{Y} \end{array} \right] \tag{15}$$

with $\mathcal{X} = [\mathcal{X}_1 \ldots \mathcal{X}_M]$, $\mathcal{Y} = [\mathcal{Y}_1; \ldots; \mathcal{Y}_M]$, $\vec{1} = [1; \ldots; 1]$, $\alpha = [\alpha_1; \ldots; \alpha_M]$ and

$$\Omega_{ij} = K(\mathcal{X}_i, \mathcal{X}_j) = \varphi(\mathcal{X}_i)^T \varphi(\mathcal{X}_j), \;\; i, j = 1, ..., N. \tag{16}$$

The support values α_i are proportional to the errors at the data points (14), while in the case of (8) most values are equal to zero. Hence sparseness is lost in the least squares case but it can be imposed by pruning the support value spectrum (Suykens *et al.*, 2000a, 2000b). Although other norms such as Huber's loss function are more appropriate concerning robustness (Smola, 1999), least squares norms are used in many applications in identification and control theory such as in the context of prediction error algorithms (Ljung, 1987). In this paper we restrict to the least squares case, because of the equality constraints in the problem formulation (which is more tractable than a SVM version with inequality constraints) in order to introduce SVM's within the context of optimal control.

4 Optimal control by support vector machines

Now, we relate the training data $\{\mathcal{X}_i, \mathcal{Y}_i\}_{i=1}^M$ in (5) to the state space and action space in (1), i.e. $\{x_k, u_k\}_{k=1}^N$. We state the following N-stage optimal control problem:

$$\min \mathcal{J}(x_k, u_k, w, e_k) = \mathcal{J}_N(x_k, u_k) + \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 \tag{17}$$

subject to the system dynamics

$$x_{k+1} = f(x_k, u_k), \quad k = 1, ..., N \ (x_1 \text{ given})$$

and the control law

$$u_k = w^T \varphi(x_k) + e_k, \quad k = 1, ..., N \tag{18}$$

where $\mathcal{J}_N$ is defined in (1), $w \in \mathbb{R}^{n_h}$, $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ with n_h the number of hidden units (which can be infinite dimensional)[1] of $\varphi(\cdot)$. The actual control signal applied to the plant is assumed to be $w^T \varphi(x_k)$. In the linear control case one has $\varphi(x_k) = x_k$ with $n_h = n$. In the sequel we will employ radial basis function (RBF) kernels. The support vector method, as it has been originally introduced, exploits Mercer's condition such that one doesn't have to construct $\varphi(\cdot)$. Although the use of this kernel function is preferred, one could also evaluate $\varphi(\cdot)$ explicitly, as it has been demonstrated in Suykens & Vandewalle (1999a) for multilayer perceptron classifiers. For the RBF case, (18) becomes then

$$u_k = \sum_{i=1}^{n_h} w_i \exp(-\frac{1}{\sigma^2} \|x_k - c_i\|_2^2) + e_k \tag{19}$$

where $c_i \in \mathbb{R}^n$ are chosen centers and σ is a chosen constant. One can take for example the set $\{c_i\}_{i=1}^{n_h}$ equal to $\{x_k\}_{k=1}^N$ in order to avoid additional unknown parameters for the centers which is also motivated by the work on regularization networks by Poggio & Girosi (1990). In standard SVM theory for static function estimation problems σ can be selected so as to minimize an upper bound on the generalization error. These bounds are not applicable in the context of SVM control due to the fact that the input patterns to the activation function are not independent from each other. Hence σ should be chosen as hoc

[1]The number of hidden units of the LS-SVM with kernel function K is not n_h but N.

(typically avoid values which are too small) or could be taken as an additional unknown within the cost function J.

In order to find the optimal control law we construct the Lagrangian

$$\begin{array}{c}\mathcal{L}(x_k, u_k, w, e_k; \lambda_k, \alpha_k) = \mathcal{J}_N(x_k, u_k) + \frac{1}{2}w^T w + \gamma \frac{1}{2}\sum_{k=1}^{N} e_k^2 + \\ \sum_{k=1}^{N} \lambda_k^T[x_{k+1} - f(x_k, u_k)] + \sum_{k=1}^{N} \alpha_k[u_k - w^T\varphi(x_k) - e_k].\end{array} \tag{20}$$

The conditions for optimality are given by

$$\left\{\begin{array}{rcll}
\frac{\partial \mathcal{L}}{\partial x_k} & = & \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k - & \\
 & & \alpha_k \frac{\partial}{\partial x_k}[w^T\varphi(x_k)] = 0, \qquad k = 2, ..., N & \text{(adjoint equation)} \\
\frac{\partial \mathcal{L}}{\partial x_{N+1}} & = & \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 & \text{(adjoint final condition)} \\
\frac{\partial \mathcal{L}}{\partial u_k} & = & \frac{\partial h}{\partial u_k} - \lambda_k^T \frac{\partial f}{\partial u_k} + \alpha_k = 0, \quad k = 1, ..., N & \text{(variational condition)} \\
\frac{\partial \mathcal{L}}{\partial w} & = & w - \sum_{k=1}^{N} \alpha_k \varphi(x_k) = 0 & \text{(support vectors)} \\
\frac{\partial \mathcal{L}}{\partial e_k} & = & \gamma e_k - \alpha_k = 0 \qquad k = 1, ..., N & \text{(support values)} \\
\frac{\partial \mathcal{L}}{\partial \lambda_k} & = & x_{k+1} - f(x_k, u_k) = 0, \quad k = 1, ..., N & \text{(system dynamics)} \\
\frac{\partial \mathcal{L}}{\partial \alpha_k} & = & u_k - w^T\varphi(x_k) - e_k = 0, \quad k = 1, ..., N & \text{(SVM control).}
\end{array}\right. \tag{21}$$

For RBF kernels one has

$$\frac{\partial}{\partial x_k}[w^T\varphi(x_k)] = -2\sum_i w_i \exp(-\frac{1}{\sigma^2}\|x_k - c_i\|_2^2)\ (x_k - c_i)/\sigma^2. \tag{22}$$

The set of nonlinear equations (21) is of the form

$$F_1(x_k, x_{N+1}, u_k, w, e_k, \lambda_k, \alpha_k) = 0 \tag{23}$$

for $k = 1, ..., N$ with x_1 given and can be numerically solved in the unknown variables. In Appendix A a formulation without e_k variables is given. In comparison with function estimation or classification problems, one looses the advantage of solving a quadratic program or a linear least squares problem. Nevertheless, one is still able to exploit some of the interesting SVM features.

As in SVM theory, Mercer's condition can be applied within (21), by replacing $w = \sum_k \alpha_k \varphi(x_k)$ in the equations. For kernels satisfying Mercer's condition one can impose

$$K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l). \tag{24}$$

For RBF kernels one takes (Vapnik, 1995)

$$K(x_k, x_l) = \exp(-\eta \, \|x_k - x_l\|_2^2) \tag{25}$$

with η a positive real constant. One sees that (25) doesn't contain the centers c_i as in (19). After this elimination of w, a set of nonlinear equations of the form

$$F_2(x_k, x_{N+1}, u_k, \lambda_k, \alpha_k) = 0 \tag{26}$$

for $k = 1, ..., N$ with x_1 given is obtained. More specifically, after exploiting Mercer's condition, one has

$$\begin{cases} \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k - \alpha_k \sum_{l=1}^N \alpha_l \frac{\partial K(x_k, x_l)}{\partial x_k} = 0, & k = 2, ..., N \\ \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 & \\ \frac{\partial h}{\partial u_k} - \lambda_k^T \frac{\partial f}{\partial u_k} + \alpha_k = 0, & k = 1, ..., N \\ x_{k+1} - f(x_k, u_k) = 0, & k = 1, ..., N \\ u_k - \sum_{l=1}^N \alpha_l K(x_l, x_k) - \alpha_k/\gamma = 0, & k = 1, ..., N. \end{cases} \tag{27}$$

For RBF kernels one has

$$\frac{\partial K(x_k, x_l)}{\partial x_k} = -2\eta \, (x_k - x_l) \, \exp(-\eta \, \|x_k - x_l\|_2^2). \tag{28}$$

The actual control signal applied to the plant becomes

$$u_k = \sum_{l=1}^{N} \alpha_l K(x_l, x_k) \tag{29}$$

where $\{x_l\}_{l=1}^N, \{\alpha_l\}_{l=1}^N$ are obtained from solving the set of nonlinear equations (27) and x_k is the actual state vector at time k. The data $\{x_l\}_{l=1}^N$ are used as support vector data for the control signal. Furthermore, note that e_k has been taken equal to zero in (29).

5 Alternative formulation and local stability

In this Section we give an alternative formulation to the LS-SVM control approach in less unknowns, together with a method for imposing local stability at the origin for the autonomous closed-loop system.

Based upon (21)(27) we consider the optimization problem

$$\min_{x_k, \alpha_k} J_N(x_k, x_k^r, u_k) + \lambda \sum_{k=1}^{N} \alpha_k^2 \tag{30}$$

subject to

$$\begin{cases} x_{k+1} &= f(x_k, u_k), \quad x_1 \text{ given} \\ u_k &= \sum_{l=1}^{N} \alpha_l \, K([x_l; x_l^r], [x_k; x_k^r]). \end{cases}$$

In this formulation a reference state vector x_k^r is considered and $\gamma \to \infty$ is taken with respect to (27). A regularization by the term $\sum_k \alpha_k^2$ is included (see also Smola (1999)) with λ is a positive real constant. Instead of solving a set of nonlinear equations one solves then a constrained nonlinear optimization problem with less unknowns than in the approaches of the previous Section. From the formulation (30) the difference between standard neural network controllers and LS-SVM control is clear. In the former one solves a parametric optimization problem in the unknown interconnection weights while in the latter also the state vector sequence is part of the unknown parameter vector in the optimization problem. Furthermore, standard neural network approaches always work in the primal weight space, while in SVM methodology the computations are done in the dual space such that the number of unknowns equals the number of training data points (and not the number of weights in the primal space, which can be infinite dimensional).

For RBF kernels the parameter η in (25) can be taken as an additional unknown to the cost function (30). When applying an optimization method the constraints in (30) will only hold with a certain tolerance. These small numerical errors may lead to differences between the state vectors as solution to (30) and the simulation of the closed-loop system with LS-SVM control, especially in the control of unstable systems. For control of stable systems this problem will be less critical. A simulation of the closed-loop system with LS-SVM control

$$\hat{x}_{k+1} = f(\hat{x}_k, \sum_{l=1}^{N} \alpha_l \, K([x_l; x_l^r], [\hat{x}_k; x_k^r])) \quad \hat{x}_1 = x_1 \text{ given} \tag{31}$$

is always needed in order to validate the results, where $\{x_l\}_{l=1}^N$ and $\{\alpha_l\}_{l=1}^N$ are obtained as a solution to (30).

In order to impose local stability at a fixed equilibrium point x^{eq}, one locally linearizes the autonomous closed-loop simulation model (31) by evaluating $\frac{\partial f}{\partial \hat{x}_k}$ at x^{eq}. A LS-SVM control design with local stability constraint can then be formulated as:

$$\min_{x_k, \alpha_k, Q} \; J_N(x_k, x^{eq}, u_k) + \lambda \sum_{k=1}^{N} \alpha_k^2 \tag{32}$$

subject to

$$\begin{cases} x_{k+1} = f(x_k, \sum_{l=1}^N \alpha_l K([x_l; x^{eq}], [x_k; x^{eq}])), \;\; x_1 \text{ given} \\ \\ A^T P A - P < 0 \end{cases}$$

where $P = Q^T Q$, $A = \frac{\partial f}{\partial \hat{x}_k} \mid_{\hat{x}_k = x^{eq}}$ and $Z < 0$ denotes a negative definite symmetric matrix.

Imposing robust local stability for the closed-loop system has been successfully applied within the context of NLq neural control theory (Suykens *et al.*; 1996, 1997). Within NLq theory sufficient conditions for global asymptotic stability and input/output stability with finite L_2-gain are available, which can also be employed in order to impose robust local stability of the origin for the closed-loop system. Based upon NLq stability criteria, dynamic backpropagation can be modified by imposing matrix inequality constraints as in (32). For (32) also robust local stability can be imposed, e.g. based upon (Sezer & Siljak, 1992).

6 Simulation examples

6.1 Example 1

Here we illustrate the LS-SVM optimal control method of Section 4 on an example reported in Narendra & Mukhopadhyay (1997). Given the nonlinear system

$$\left\{ \begin{array}{lcl} x_{1,k+1} & = & 0.1 x_{1,k} + 2 \frac{u_k + x_{2,k}}{1+(u_k+x_{2,k})^2} \\ \\ x_{2,k+1} & = & 0.1 x_{2,k} + u_k (2 + \frac{u_k^2}{1+x_{1,k}^2 + x_{2,k}^2}) \end{array} \right. \tag{33}$$

we consider a state vector tracking problem with $h(x_k, u_k) = (x_k - x_k^r)^T Q (x_k - x_k^r) + u_k^T R u_k$, $\rho(x_{N+1}) = (x_{N+1} - x_{N+1}^r)^T Q (x_{N+1} - x_{N+1}^r)$ where x_k^r is the reference trajectory to be tracked. We aim at tracking the first state variable and choose $Q = \mathrm{diag}\{1, 0.001\}$, $R = 1$ for $x_k^r = [\sin(2\pi k/20); \cos(2\pi k/20)]$ with $k = 1, N$ and $N = 20$. The given initial state is $x_1 = [0; 0]$. As control law is taken $w^T \varphi([x_k; x_k^r])$, which results only into a slight modification of (21) and (27). In Fig.1 simulation results are shown for the method (27) that makes use of Mercer's condition. The set of nonlinear equations was solved using Matlab's optimization toolbox (function *leastsq*) with unknowns $x_k, u_k, \lambda_k, \alpha_k, \sigma$ (variables e_k eliminated) for $\gamma = 100$. The unknowns were randomly initialized with zero mean and standard deviation 0.3. The plots show the simulation results for the closed-loop system $\hat{x}_{k+1} = f(\hat{x}_k, \sum_{l=1}^N \alpha_l K(x_l, \hat{x}_k))$ with RBF kernel. The controller is generalizing well with respect to other initial conditions than the origin (for which it has been trained) and beyond the time horizon of $N = 20$. The method (22) without the use of Mercer's condition gave similar results by taking the centers $\{c_i\}$ equal to $\{x_k\}$.

6.2 Example 2: inverted pendulum

In this example we illustrate the LS-SVM control method of Section 5 on the problem of swinging up an inverted pendulum with local stabilization at the endpoint, which has been investigated in Suykens *et al.* (1994).

A nonlinear state space model for the inverted pendulum (Fig.2) system is given by

$$\dot{x} = F(x) + G(x)u \tag{34}$$

with

$$F(x) = \begin{bmatrix} x_2 \\ \frac{\frac{4}{3}mlx_4^2 sinx_3 - \frac{mg}{2} sin(2x_3)}{\frac{4}{3}m_t - mcos^2 x_3} \\ x_4 \\ \frac{m_t g sinx_3 - \frac{ml}{2} x_4^2 sin(2x_3)}{l(\frac{4}{3}m_t - mcos^2 x_3)} \end{bmatrix}, \; G(x) = \begin{bmatrix} 0 \\ \frac{4}{3} \frac{1}{\frac{4}{3}m_t - mcos^2 x_3} \\ 0 \\ -\frac{cosx_3}{l(\frac{4}{3}m_t - mcos^2 x_3)} \end{bmatrix}.$$

The state variables x_1, x_2, x_3, x_4 are respectively position and velocity of the cart, angle of the pole with the vertical and rate of change of the angle. The input signal u is the force applied to the cart's center of mass. The symbols m, m_t, l, g denote respectively mass of the pole, total mass of cart and pole, half pole length and the acceleration due to gravity. We take $m = 0.1$, $m_t = 1.1$, $l = 0.5$. Remark that in the autonomous case $x = [0; 0; 0; 0]$ (pole up) and $x = [0; 0; \pi; 0]$ (pole down) are equilibrium points. The linearized system around the target equilibrium point (the origin) is given by

$$\dot{x} = Ax + Bu \tag{35}$$

with

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{\frac{4}{3}m_t - m} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{m_t g}{l(\frac{4}{3}m_t - m)} & 0 \end{bmatrix}, \; B = \begin{bmatrix} 0 \\ \frac{4}{3} \frac{1}{\frac{4}{3}m_t - m} \\ 0 \\ -\frac{1}{l(\frac{4}{3}m_t - m)} \end{bmatrix}. \tag{36}$$

In Suykens *et al.* (1994) for the neural controller, a Linear Quadratic Regulator (LQR) controller (Franklin *et al.*, 1990; Boyd & Barratt, 1991) was designed first based upon the linearized model (eventually, also robust linear controllers could be designed, such as based upon H_∞ control and μ theory, if additional robustness with respect to noise and uncertainties would be needed). This was done in order to impose a set of constraints on the choice of the interconnection weights of a multilayer perceptron controller. As a result a swinging up neural control has been obtained which ensures that the pole remains locally stabilized in its upright position.

In order to solve the swinging up problem using LS-SVM control, we follow a somewhat similar approach for the LS-SVM control case now. Based on the continuous time model (34) we design a LQR controller for $Q = I$, $R = 0.01$ in the LQR cost function $\int_0^\infty (x^T Q x + u^T R u) dt$ (Matlab function lqr). Then the continuous time model (34) is discretized by using a fourth order Runge-Kutta integration rule with constant step $h = 0.05$ into a discrete time model of the form

$$x_{k+1} = \mathcal{F}(x_k, u_k) \tag{37}$$

where u_k is assumed to be constant in the time intervals $[kh, (k+1)h)$ (zero order hold). The control law is taken as

$$u_k = (L_{lqr} - L_\Delta) x_k + u_k^{svm} \tag{38}$$

with

$$u_k^{svm} = \sum_{l=1}^{N} \alpha_l K(x_l, x_k) \tag{39}$$

where L_{lqr} is the resulting feedback matrix from LQR design and

$$L_\Delta = \frac{\partial u_k^{svm}}{\partial x_k} \, |_{x_k=0} \tag{40}$$

is a modification to the LS-SVM control law such that locally at the origin the overall controller u_k is acting as a LQR controller, in a continuous time sense. The mixture between a continuous time LQR result and the discrete time SVM control law may look surprising at first sight, but here it is a most convenient way to design an overall LS-SVM controller for the given continuous time model. An approach according to (32) would be much more complicated here due to the Runge-Kutta integration rule. An RBF kernel function is used and expression (22) is applied in order to compute (40).

The resulting closed-loop simulation model is given by

$$\hat{x}_{k+1} = \mathcal{F}(\hat{x}_k, (L_{lqr} - L_\Delta)\hat{x}_k + \sum_{l=1}^{N} \alpha_l K(x_l, \hat{x}_k)) \tag{41}$$

with $\hat{x}_1 = x_1 = 0$ given. In (41) $\{x_l\}_{l=1}^N$ and $\{\alpha_l\}_{l=1}^N$ are the solution to the constrained optimization problem (30). As cost function in (30) has been defined

$$J_N = \sum_{k=N-5}^{N} \|x_k\|_2^2 \tag{42}$$

with $x_k^r = 0$ and $\lambda = 0$. As time horizon has been taken $N = 100$ (5 seconds). The constrained optimization problem has been solved using Matlab's optimization toolbox by SQP (sequential quadratic programming) (function constr). The values x_k and α_k were initialized by taking small random values. The parameter η of the RBF kernel was taken as additional unknown in the optimization problem with starting point 0.1. In order to emphasize the equations of the constraints, they have been multiplied with a factor 1000. This is needed due to fact that the system to be controlled is unstable and small differences between x_k (as solution to (30)) and $\hat{x}_k$ (in the simulation model (31)) may cause large differences otherwise. Simulation results of the closed-loop simulation model are shown in Fig.3-4.

6.3 Example 3: ball and beam

Here we discuss the LS-SVM control method of Section 5 on a ball and beam system which has been described in Hauser *et al.* (1992).

The continuous time system description of the ball and beam system (Fig.5) is given by (34) with

$$F(x) = \begin{bmatrix} x_2 \\ B(x_1 x_4^2 - G \sin x_3) \\ x_4 \\ 0 \end{bmatrix}, \; G(x) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{43}$$

where $x = [x_1; x_2; x_3; x_4] = [r; \dot{r}; \theta; \dot{\theta}]$ with r the ball position and θ the beam angle and $B = M/(J_b/R_b^2 + M)$ where M, J_b, R_b are the mass, moment of inertia and radius of the ball, respectively. For the control input one has $\tau = 2Mr\dot{r}\dot{\theta} + MGr\cos\theta + (Mr^2 + J + J_b)u$ where τ, G, J denote the torque applied to the beam, the acceleration of gravity and the moment of inertia of the beam, respectively. As control objective we consider tracking of the ball position for a reference input $d(t) = 2\cos(\pi t/5)$.

As for the inverted pendulum example, a LQR design is incorporated in the LS-SVM control law. Here we impose local stability for the autonomous closed-loop system, which is motivated by the application of NLq neural control theory (Suykens *et al.*; 1996, 1997), to the control of a real-life ball and beam system (Verrelst *et al.*, 1998). We proceed in a similar way as for the inverted pendulum system. The continuous time state space description (43) is linearized around the origin, for which a LQR controller is designed with $Q = I$, $R = 1$ in the LQR cost function. Then the continuous time model is discretized by using a fourth order Runge-Kutta integration rule with constant step $h = 0.3$ into a discrete time model of the form (37). The first component of the reference state vector $x^r_{1,k}$ is derived from $d(t)$ according to this step size. The control law is taken as

$$u_k = (L_{lqr} - L_\Delta) \begin{bmatrix} x_k \\ x^r_{1,k} \end{bmatrix} + u_k^{svm} \tag{44}$$

with

$$u_k^{svm} = \sum_{l=1}^{N} \alpha_l K(\begin{bmatrix} x_l \\ x^r_{1,l} \end{bmatrix}, \begin{bmatrix} x_k \\ x^r_{1,k} \end{bmatrix}) \tag{45}$$

where L_{lqr} is the resulting feedback matrix from LQR (for the autonomous closed-loop system) and

$$L_\Delta = \frac{\partial u_k^{svm}}{\partial x_k} |_{x_k=0} \ . \tag{46}$$

Note that (46) also depends on the reference $x^r_{1,k}$. A zero intial state has been taken. SQP was applied for constrained nonlinear optimization of (30) with a similarly chosen initial unknown parameter vector as for the inverted pendulum example. Simulation results of the closed-loop simulation model are shown in Fig.5.

7 Conclusions

In this paper we introduced the use of support vector machines for solving optimal control problems. The N-stage optimal control problem has been formulated for a nonlinear state feedback controller consisting of support vector machines. Therefore a least squares version of SVM's has been considered, leading to equality constraints instead of inequality constraints that would result for example from Vapnik's epsilon-insensitive loss function. Sparseness is lost in the least squares case but other SVM features are still applicable, such as Mercer's condition and the fact that no centers have to be determined for RBF kernels. However, in order to keep the optimal control problem formulation tractable, an equality constraint based formulation of SVM's has been preferred, motivating the use of LS-SVM's. The cost function for the control problem and for the least squares SVM have been formulated within one objective function. The solution is characterized then by a set of nonlinear equations. A drawback of this approach is that the problem contains many unknowns, even in the equality constraint case of LS-SVM's. Therefore an alternative formulation of LS-SVM optimal control has been given in less unknowns. Imposing local stability for the control law has been discussed. A main difference of LS-SVM control with other neural and more specifically RBF controllers is that in the former case the state vector sequence of the considered time horizon is supporting the control signal and has to be obtained as solution to the constrained nonlinear optimization problem, while for the latter one often solves a parametric optimization problem in the unknown interconnection weights. The number of hidden units of the LS-SVM with RBF kernel in the control law equals the considered number of points in the discrete time evolution, while in primal weight space the number of parameters can be infinite. Special attention has to be paid for the closed-loop simulation models of the LS-SVM controllers, whose result could differ from the solution to the constrained optimization problem, especially in the control of unstable systems. Examples of LS-SVM control are given on swinging up an inverted pendulum with local stabilization in its upright position and tracking for a ball and beam system.

References

Bishop C.M. (1995). *Neural networks for pattern recognition*, Oxford University Press.

Boyd S., Barratt C. (1991). *Linear controller design, limits of performance*, Prentice-Hall.

Bryson A.E. & Ho Y.C. (1969). *Applied optimal control*, Waltham, MA: Blaisdel.

Cherkassky V., Mulier F. (1998). *Learning from data: concepts, theory and methods*, John Wiley and Sons.

Fletcher R. (1987). *Practical methods of optimization*, Chichester and New York: John Wiley and Sons.

Franklin G.F., Powell J.D., Workman M.L. (1990). *Digital control of dynamic systems*, Reading MA: Addison-Wesley.

Gill P.E., Murray W. & Wright M.H. (1981). *Practical Optimization*, London: Academic Press.

Golub G.H. & Van Loan C.F. (1989). *Matrix Computations*, Baltimore MD: Johns Hopkins University Press.

Hauser J., Sastry S. & Kokotovic P. (1992). Nonlinear control via approximate input-output linearization: the ball and beam example, *IEEE Transactions on Automatic Control*, Vol.37, No.3, pp.392-398.

Haykin S. (1994). *Neural Networks: a Comprehensive Foundation*, Macmillan College Publishing Company: Englewood Cliffs, 2nd edition.

Ljung L. (1987). *System Identification: Theory for the User*, Prentice-Hall.

Müller K.R., Smola A.J., Rätsch G., Schölkopf B., Kohlmorgen J., Vapnik V. (1997) Predicting time series with support vector machines, *ICANN'97 Proc. of the Int. Conf. on Artificial Neural Networks* (Eds. W. Gerstner, A. Germond, M. Hasler, J.-D. Nicoud), LNCS 1327, Spinger Berlin, pp.999-1004.

Narendra K.S. & Parthasarathy K. (1991). Gradient methods for the optimization of dynamical systems containing neural networks, *IEEE Transactions on Neural Networks*, Vol.2, No.2, pp.252-262.

Narendra K.S. & Mukhopadhyay S. (1997). Adaptive control using neural networks and approximate models, *IEEE Transactions on Neural Networks*, Vol.8, No.3, pp.475-485.

Nguyen D. & Widrow B. (1990). Neural networks for self-learning control systems, *IEEE Control Systems Magazine*, 10(3), pp.18-23.

Parisini T. & Zoppoli R. (1994). Neural networks for feedback feedforward nonlinear control systems, *IEEE Transactions on Neural Networks*, Vol.5, No.3, pp.436-449.

Plumer E.S. (1996). Optimal control of terminal processes using neural networks, *IEEE Transactions on Neural Networks*, Vol.7, No.2, pp.408-418.

Poggio T., Girosi F. (1990). Networks for approximation and learning, *Proceedings of the IEEE*, Vol.78, No.9, pp.1481-1497.

Saerens M., Renders J.-M. & Bersini H. (1993). Neural controllers based on backpropagation algorithm. In *IEEE Press Book on Intelligent Control: Theory and Practice*, M. M. Gupta, N. K. Sinha (Eds.), IEEE Press.

Sanner R.M. & Slotine J.-J. E. (1992). Gaussian networks for direct adaptive control, *IEEE Transactions on Neural Networks*, Vol.3, No.6, pp.837-863.

Saunders C., Gammerman A. & Vovk V. (1998). Ridge Regression Learning Algorithm in Dual Variables, *Proceedings of the 15th International Conference on Machine Learning ICML-98*, Madison-Wisconsin.

Schölkopf B., Sung K.-K., Burges C., Girosi F., Niyogi P., Poggio T. & Vapnik V. (1997). Comparing support vector machines with Gaussian kernels to radial basis function classifiers, *IEEE Transactions on Signal Processing,* Vol.45, No.11, pp.2758-2765.

Schölkopf B., Burges C.J.C., Smola A.J. (Eds.) (1999) *Advances in kernel methods - Support vector learning*, MIT Press, Cambridge MA.

Sezer M.E. & Siljak D.D. (1988). Robust stability of discrete systems, *International Journal of Control*, Vol.48, No.5, pp.2055-2063.

Smola A., Schölkopf B. & Müller K.-R. (1998). The connection between regularization operators and support vector kernels, *Neural Networks*, Vol.11, No.4.

Smola, A., & Schölkopf, B. (1998). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, 22, pp.211-231.

Smola A. (1999). *Learning with Kernels*, PhD Thesis, published by: GMD, Birlinghoven.

Suykens J.A.K., De Moor B. & Vandewalle J. (1994). Static and dynamic stabilizing neural controllers, applicable to transition between equilibrium points, *Neural Networks*, Vol.7, No.5, pp.819-831.

Suykens J.A.K., Vandewalle J. & De Moor B. (1996). *Artificial neural networks for modelling and control of non-linear systems*, Kluwer Academic Publishers: Boston.

Suykens J.A.K., De Moor B. & Vandewalle J. (1997). NL_q theory: a neural control framework with global asymptotic stability criteria, *Neural Networks*, Vol.10, No.4, pp.615-637.

Suykens J.A.K. & Vandewalle J. (1999a). Training multilayer perceptron classifiers based on a modified support vector method, *IEEE Transactions on Neural Networks*, Vol.10, No.4, pp.907-912.

Suykens J.A.K. & Vandewalle J. (1999b). Least squares support vector machine classifiers, *Neural Processing Letters*, Vol.9, No.3, pp.293-300.

Suykens J.A.K., Lukas L., Vandewalle J. (2000a) "Sparse approximation using least squares support vector machines," *IEEE International Symposium on Circuits and Systems ISCAS 2000*, Geneva, Switzerland, May 28-31, pp.II-757-760.

Suykens J.A.K., Lukas L., Vandewalle J. (2000b) "Sparse Least Squares Support Vector Machine Classifiers," *8th European Symposium on Artificial Neural Networks ESANN 2000*, Bruges Belgium, pp.37-42.

Vapnik V. (1995). *The nature of statistical learning theory*, Springer-Verlag, New-York.

Vapnik V., Golowich S. & Smola A. (1997). Support vector method for function approximation, regression estimation and signal processing, *In Advances in Neural Information Processing Systems, 9*, MIT Press.

Vapnik V. (1998a). *Statistical learning theory*, John Wiley, New-York.

Vapnik V. (1998b). The support vector method of function estimation, In *Nonlinear Modeling: advanced black-box techniques*, Suykens J.A.K., Vandewalle J. (Eds.), Kluwer Academic Publishers, Boston, pp.55-85.

Verrelst H., Van Acker K., Suykens J., Motmans B., De Moor B. & Vandewalle J. (1998). Application of NLq neural control theory to a ball and beam system, *European Journal of Control*, 4, pp.148-157.

Werbos P. (1990). Backpropagation through time: what it does and how to do it, *Proceedings of the IEEE*, 78 (10), pp.1150-1560.

A Formulation without error variables

A variation on the problem (17)(18) without the use of the variables e_k is

$$\min \mathcal{J}_0(x_k, w) = \mathcal{J}_N(x_k, u_k(w, x_k)) + \frac{1}{2} w^T w \tag{47}$$

subject to

$$\begin{cases} x_{k+1} & = & f(x_k, u_k(w, x_k)), \quad k = 1, ..., N \ (x_1 \text{ given}) \\ \\ u_k & = & w^T \varphi(x_k) \quad k = 1, ..., N. \end{cases}$$

This gives the Lagrangian

$$\mathcal{L}_0(x_k, w; \lambda_k) = \mathcal{J}_N(x_k, u_k(w, x_k)) + \frac{1}{2} w^T w + \sum_{k=1}^{N} \lambda_k^T [x_{k+1} - f(x_k, w^T \varphi(x_k))]. \tag{48}$$

The conditions for optimality are given by

$$\begin{cases} \frac{\partial \mathcal{L}_0}{\partial x_k} & = & \frac{\partial h}{\partial x_k} + \lambda_{k-1} - (\frac{\partial f}{\partial x_k})^T \lambda_k - (\frac{\partial f}{\partial u_k} \frac{\partial u_k}{\partial x_k})^T \lambda_k = 0, \quad k = 2, ..., N \\ \\ \frac{\partial \mathcal{L}_0}{\partial x_{N+1}} & = & \frac{\partial \rho}{\partial x_{N+1}} + \lambda_N = 0 \\ \\ \frac{\partial \mathcal{L}_0}{\partial w} & = & w - \sum_{k=1}^{N} \lambda_k^T \frac{\partial f}{\partial u_k} \varphi(x_k) = 0 \\ \\ \frac{\partial \mathcal{L}_0}{\partial \lambda_k} & = & x_{k+1} - f(x_k, w^T \varphi(x_k)) = 0, \quad k = 1, ..., N. \end{cases} \tag{49}$$

Note that the support values are directly related here to the Lagrange multipliers λ_k. The set of nonlinear equations (49) is of the form

$$F_3(x_k, x_{N+1}, w, \lambda_k) = 0. \tag{50}$$

Mercer condition can be applied as to (49)(50), yielding a set of equations of the form

$$F_4(x_k, x_{N+1}, \lambda_k) = 0 \tag{51}$$

by elimination of w.

Captions of Figures

Fig.1: (Top) Optimal control by a least squares support vector machine with RBF kernel and application of Mercer's condition. (Full line) first state variable to be tracked; (Dashed line) actual state variable by closed-loop simulation of the system. The data $k = 1, ..., 20$ were used for training of the controller with the origin as initial state. (Bottom) simulation result for another randomly chosen initial state x_1. The LS-SVM controller shows a good generalization performance with respect to other initial states and beyond the time horizon.

Fig.2: Swinging up an inverted pendulum by a LS-SVM controller with local stabilization in its upright position. Around the target point the controller is behaving as a LQR controller. (Top) inverted pendulum system; (Bottom) simulation result which visualizes the several pole positions in time.

Fig.3: (Continued) (Top) state variables with respect to time of the closed-loop simulation model with LS-SVM controller: $x_1(t)$ (-); $x_2(t)$ (- -); x_3 (-.); $x_4(t)$ (:). The state vector data before the vertical line were used in the training process as support vector data with an RBF kernel; (Bottom) control signal u_k.

Fig.4: (Continued) (Top) $(x_1(t), x_2(t))$; (Middle) $(x_1(t), x_3(t))$; (Bottom) $(x_3(t), x_4(t))$. The initial state is marked by a square and the target state by o.

Fig.5: Tracking control of a ball and beam system by a LS-SVM controller: (Top) ball and beam system; (Bottom) reference input (-) and position of the ball (- -). The state vector data before the vertical line were used in the training process as support vector data with an RBF kernel. Shown is the closed-loop simulation result.

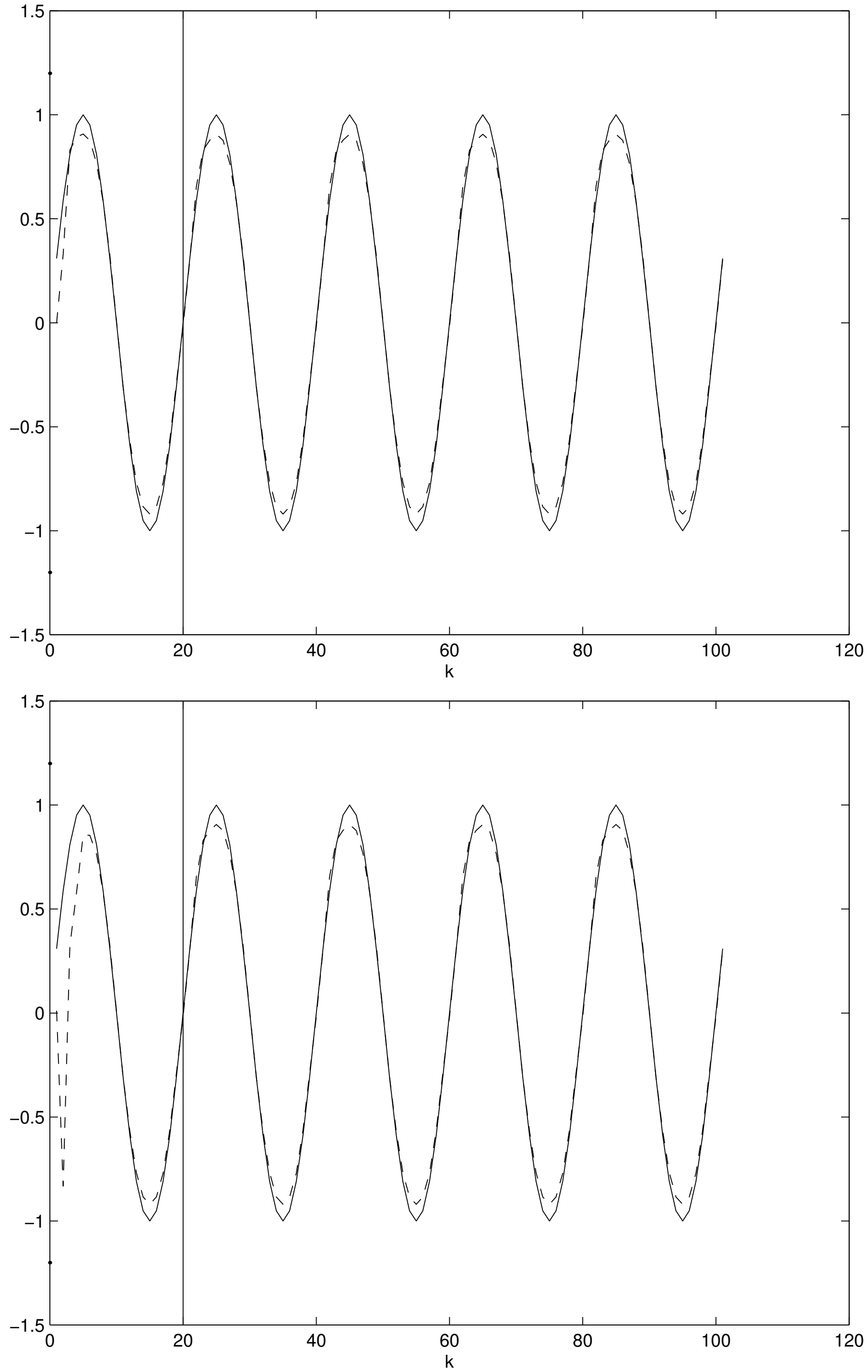
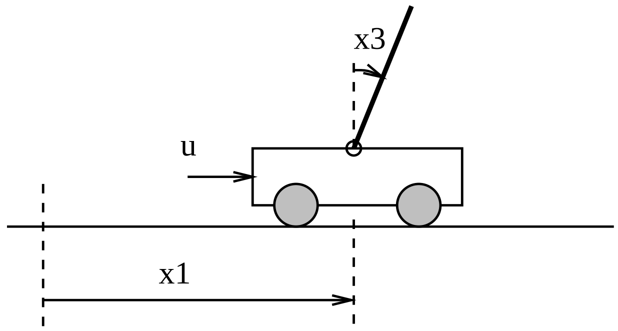

1.5
1
0.5
0
−0.5
−1
−1.5
0
20
40
60
80
100
120
k
1.5
1
0.5
0
−0.5
−1
−1.5
0
20
40
60
80
100
120
k

Fig. 1.



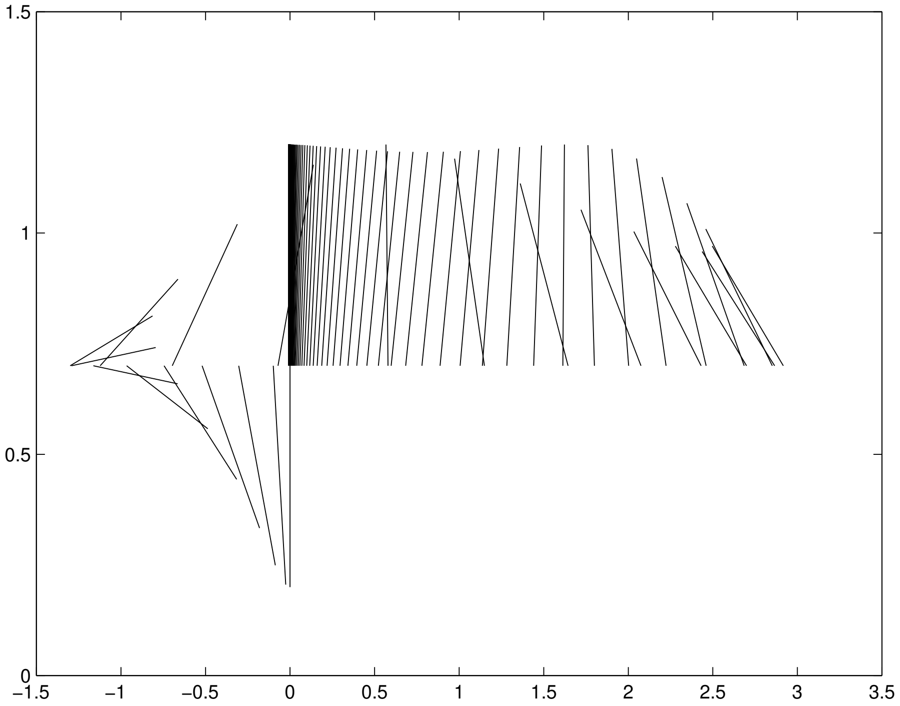


Fig. 2.

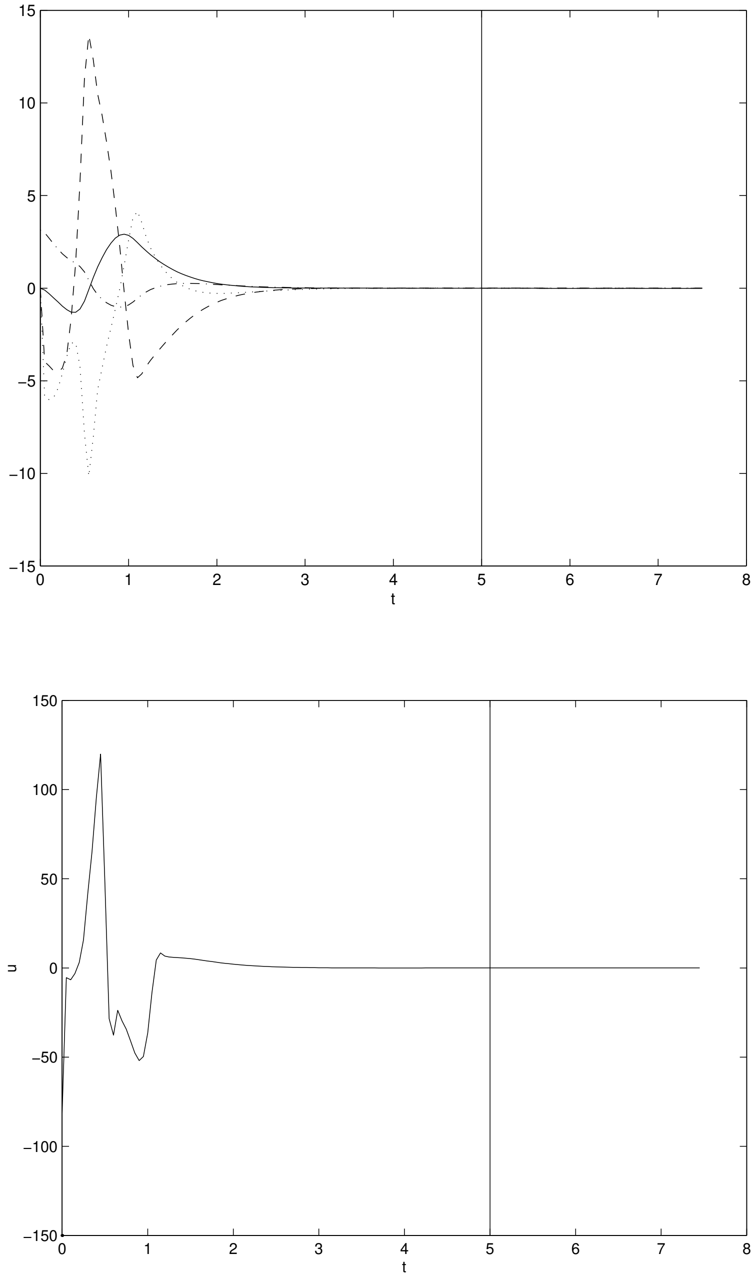


Fig. 3.

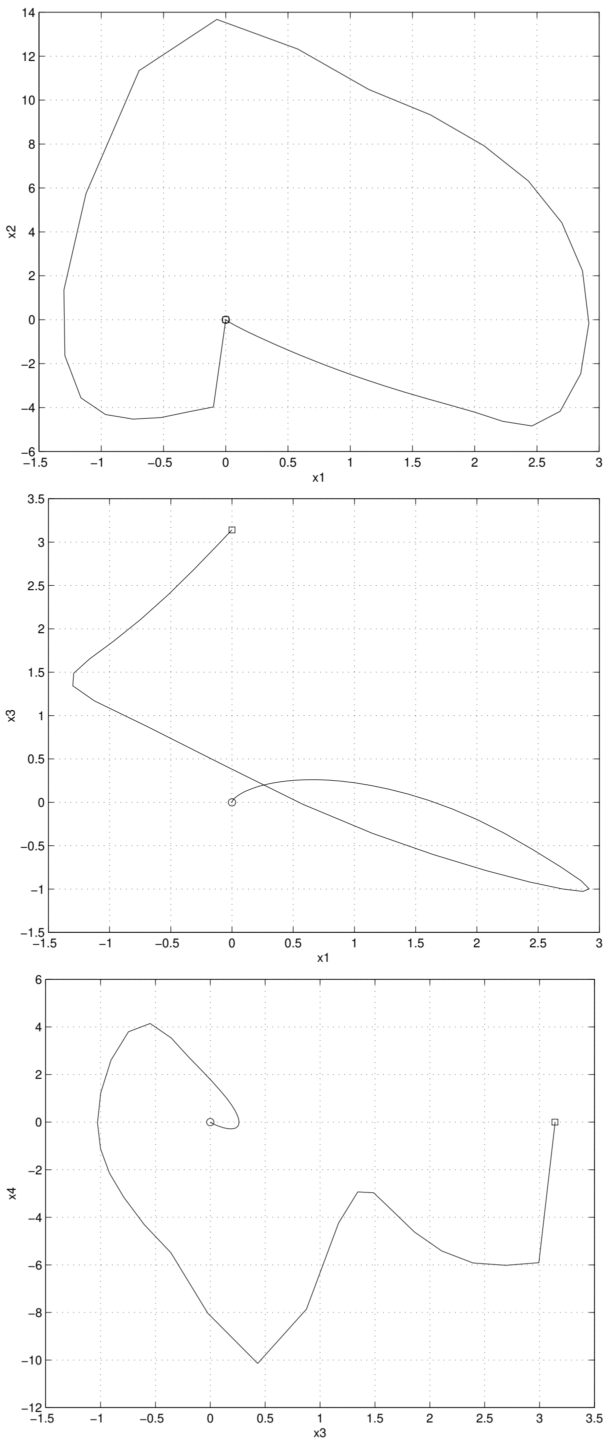


Fig. 4.

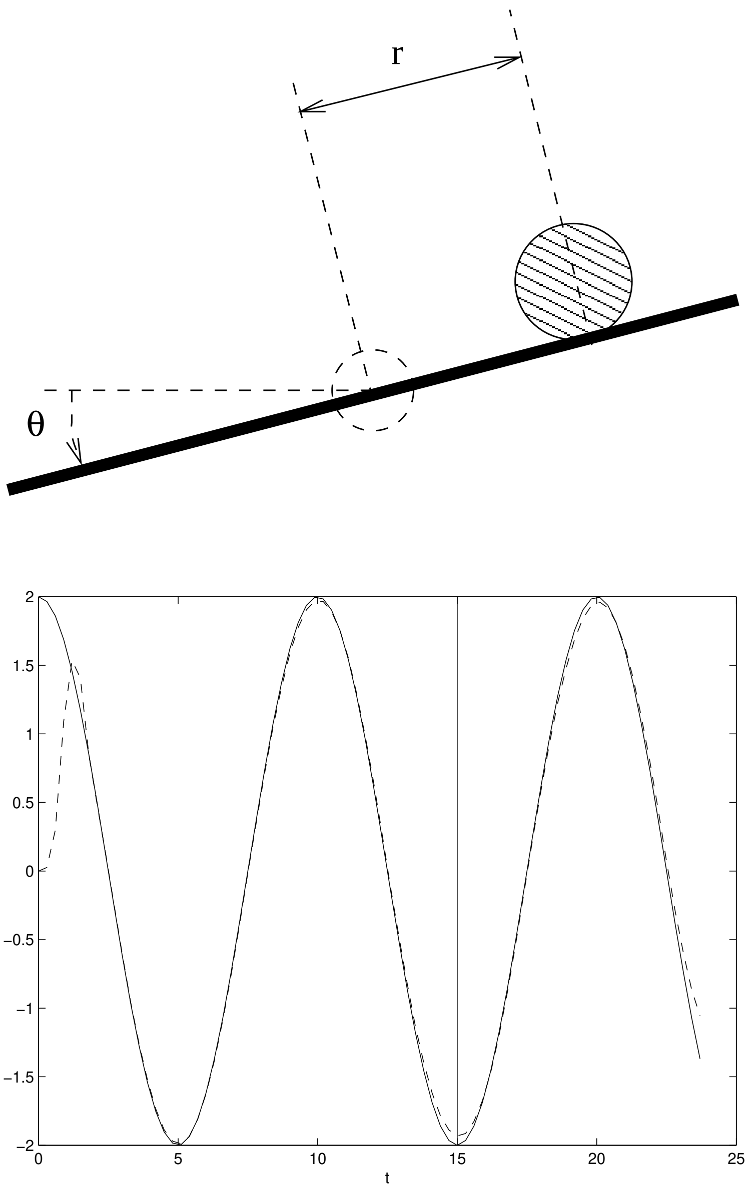


Fig. 5.