IDENTIFICATION OF A HYDRAULIC SERVO-AXIS USING SUPPORT VECTOR MACHINES

Jochen Schaab [*] **Marco Muenchhof** [**] **Michael Vogt** [**]
Rolf Isermann [**]

[*] *Darmstadt University of Technology, Institute of Flight Systems and Automatic Control, Petersenstrasse 30, 64287 Darmstadt, Germany, E-mail: schaab@fsr.tu-darmstadt.de*
[**] *Darmstadt University of Technology, Institute of Automatic Control, Landgraf-Georg-Strasse 4, 64283 Darmstadt, Germany,*
E-mail: {mmuenchhof,mvogt,risermann}@iat.tu-darmstadt.de

Abstract: In this paper, different models of the pressure buildup inside a hydraulic servo-axis are compared. These models are obtained using RBF networks, local linear models and support vector machines (SVMs), with a particular focus on the latter. For SVMs, a reduction method is derived, which allows to reduce the number of support vectors without losing the generalization abilities of the SVM. Experimental results obtained at a hydraulic servo-axis and a comparison of the different modelling techniques conclude this paper. *Copyright© 2005 IFAC.*

Keywords: hydraulic actuators, system identification, support vector machines, neural networks, nonlinear modelling

1. INTRODUCTION

At the beginning of the 20th century, hydraulic systems became a feasible alternative to other actuation principles, which could mainly be attributed to the introduction of mineral oil as the energy transmitting medium. After the end of the second world war, hydraulics were employed on a large scale in aeronautical applications and became subject to intense research, especially in the area of servo-hydraulics, i.e. hydraulic control.

Today, hydraulics are used in manifold applications, ranging from industrial processes, road vehicles, shipbuilding, aviation to construction machinery and so forth. As for example in the area of aircrafts and ships, hydraulics are often employed in safety critical applications. Due to this, it is necessary to reduce malfunctions of the hydraulic system as much as possible and to detect the onset of remaining unavoidable malfunctions as early as possible so to allow for time to initiate countermeasures or shut down the system in a safe way.

The ongoing trend towards mechatronic systems, which integrate mechanic, electronic and information processing components (Isermann, 2003), allows to implement many new functions into these intelligent components, such as more sophisticated control algorithms, fault detection and fault management systems.

Model-based fault detection employs analytical redundancy between different standard sensor signals to detect faults without the need for additional sensors and is currently under intense research in the area of hydraulic systems. Muenchhof and Isermann (2004) show, how the geometry of the control edges can be monitored, which allows to detect damages of the valve spool such as control edge wear and grooving.

In (Ramdén, 1998), the predominant topic is fault detection of hydraulic pumps. However, fault detection of an entire hydraulic servo axis comprising of a pressure supply, three valves and three cylinders was also described. Here, a neural net was used to detect valves, which were stuck close or open.

As part of the Eurofighter project, a fault tolerant rudder actuator was developed (Kubbat *et al.*, 2000). For this rudder actuator, different model-based fault detection methods, mainly based on Kalman filtering, have been developed, see e.g. (Kreß, 2002).

As is evident, model-based fault detection approaches make use of mathematical process models. There are two main requirements that these models have to meet. First, they have to mimic the behavior of the process precisely and secondly they may not be computationally expensive, since they have to be evaluated in real-time. The latter requires that these models have a lean structure.

In Section 2, the overall setup of the hydraulic linear servo axis will be described. A physics-based theoretical model of the pressure buildup inside the cylinder chamber will be presented. The process will then be modelled by three different methods: RBF networks, local linear models and support vector machines (SVMs). These different regression methods will be highlighted in Section 3. Next, the steps for solving the SVM regression problem are explained. SVMs tend to be more complex than models computed by other methods. In order to compensate this disadvantage, a reduction method for SVMs is proposed in Section 4.2. Experimental results conclude this paper and compare the different modelling techniques.

2. THE HYDRAULIC SERVO-AXIS

The system scrutineered consists of a swash-plate piston pump, a proportional valve and a differential piston. A photo of the testbed is shown in Fig. 1.

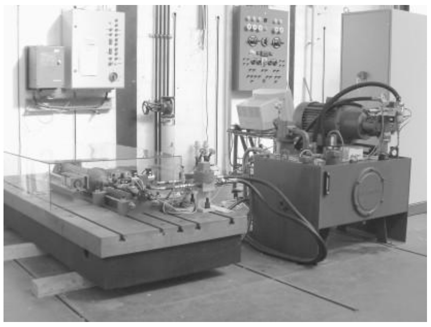

Fig. 1. Photo of the testbed

The behavior of the system is highly nonlinear. The modelling equations can e.g. be found in (Muenchhof and Isermann, 2004) and will be shortly summarized in the following: The pressure buildup in chamber A of the hydraulic cylinder is given by

$$\dot{p}_\mathrm{A} = \frac{E(p_\mathrm{A})\,(\dot{V}_\mathrm{A} - A_\mathrm{A}\,\dot{x} - G_\mathrm{AB}(p_\mathrm{A} - p_\mathrm{B}))}{V_\mathrm{0A} + A_\mathrm{A}\,x}\,. \qquad (1)$$

In this equation, E denotes the pressure dependent bulk modulus. $\dot{V}_\mathrm{A}$ is the flow into chamber A, A_A the cross-sectional area and V_0A the residual volume, i.e. the volume at displacement $x = 0$. p_A and p_B denote the corresponding pressure levels and G_AB is the coefficient of laminar leakage flow between chamber A and B, $\dot{V}_{AB}$.

The flow into chamber A is determined by the displacement of the valve spool and the subsequent opening of the control edges. The flow into chamber A is given by

$$\begin{aligned}\dot{V}_\mathrm{A} =& B_\mathrm{V1}(x_\mathrm{V})\sqrt{|p_\mathrm{P} - p_\mathrm{A}|}\,\mathrm{sign}(p_\mathrm{P} - p_\mathrm{A}) \\ &- B_\mathrm{V2}(x_\mathrm{V})\sqrt{|p_\mathrm{A} - p_\mathrm{T}|}\,\mathrm{sign}(p_\mathrm{A} - p_\mathrm{T})\,,\end{aligned} \qquad (2)$$

where B_V1 and B_V2 are the spool-displacement dependent coefficients of turbulent flow across the control edges. The spool displacement is called x_V, p_P is the pressure supplied by the pump and p_T is the pressure in the return line, whose influence is neglected ($p_\mathrm{T} = 0$). A cross-sectional drawing of the cylinder is depicted in Fig. 2.

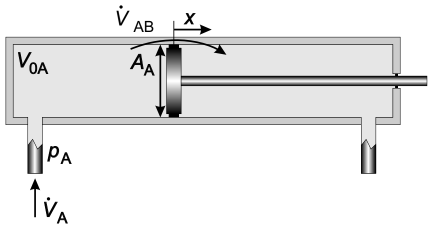

Fig. 2. Schematic view of a double acting cylinder

Equations (1) and (2) show which input signals must be supplied to a model. It is

$$\dot{p}_\mathrm{A} = f(p_\mathrm{A}, \dot{V}_\mathrm{A}, x, \dot{x}, p_\mathrm{B}) \qquad (3)$$

with

$$\dot{V}_\mathrm{A} = f(p_\mathrm{A}, p_\mathrm{P}, x_\mathrm{V}) \qquad (4)$$

3. REGRESSION METHODS

The process described in Sec. 2 will be modelled by three different methods: RBF networks, local linear models and support vector machines (SVMs), with particular interest on SVMs. This *regression* problem can be considered as a data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where the goal is to estimate the output values y_i from the *regressors* $\mathbf{x}_i$ respecting a given optimality criterion.

RBF networks are well-established model structures having an input/output behavior of the form

$$f(\mathbf{x}) = \sum_{i=1}^{q} a_i K(\mathbf{c}_i, \mathbf{x}) \qquad (5)$$

where

$$K(\mathbf{c}_i, \mathbf{x}) = \exp\Big(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma^2}\Big) \qquad (6)$$

is the Gaussian transfer function of a single neuron. The coefficients a_i are determined by least-squares

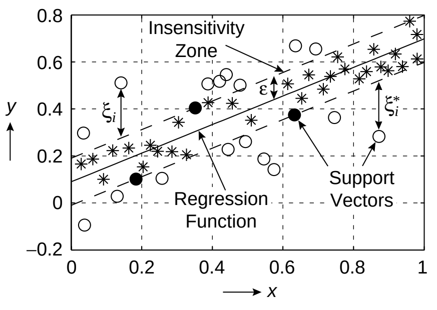

Fig. 3. Linear support vector machine regression

estimation. To allow a direct comparison with SVMs, the Gaussians' width σ is kept fixed and their centers $\mathbf{c}_i$ are located on data points $\mathbf{x}_i$. This network can also be seen as the optimal *regularization network* for a certain smoothness measure (Nelles, 2000). To ensure a good generalization performance, the number of neurons has to be limited. To find the optimal set of neurons for a given model complexity, the *orthogonal least squares* (OLS) method selects those neurons that maximally reduce the output error (Chen *et al.*, 1991).

Based on the RBF network, two enhancements can be made:

(1) to use local linear models as a more appropriate model structure while retaining the least-squares optimality criterion.
(2) to use SVMs having a different optimality criterion while keeping the model structure fixed.

As local linear structure the *local linear model tree* (LOLIMOT) is employed (Nelles, 2000) having the input/output behavior

$$f(\mathbf{x}) = \sum_{i=1}^{q} (w_{i0} + w_{i1}x_1 + \ldots + w_{in}x_n)\varphi_i(\mathbf{x})\,. \tag{7}$$

Compared to RBF networks, the coefficients a_i are augmented to linear submodels, and the Gaussians $K(\mathbf{c}_i, \mathbf{x})$ have been normalized:

$$\varphi_i(\mathbf{x}) = K(\mathbf{c}_i, \mathbf{x}) \Big/ \sum_{j=1}^{q} K(\mathbf{c}_j, \mathbf{x}) \tag{8}$$

LOLIMOT does not use the data points as the Gaussians' centers. Instead, it iteratively determines an axis-orthogonal partitioning of the input space and places a (normalized) Gaussian in each hyperrectangle. The weights w_{ij} are determined by local least-squares estimation.

Support vector machines improve the generalization performance by an alternative optimality criterion resulting from the *statistical learning theory* (Vapnik, 1995): Linear regression SVMs try to find a *flat* function

$$f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b\,, \tag{9}$$

so that all data lie within an *insensitivity zone* of size ε around the function, see Fig. 3. Outliers are treated by two sets of slack variables ξ_i and ξ_i^* measuring the distance above and below the insensitivity zone, respectively. This results in the following *primal* optimization problem (Schölkopf and Smola, 2002):

$$\min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad J_{\mathrm{p}}(\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*) = \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{10a}$$

$$\text{s.t.} \quad y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i - b \le \varepsilon + \xi_i \tag{10b}$$

$$\mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b - y_i \le \varepsilon + \xi_i^* \tag{10c}$$

$$\xi_i, \xi_i^* \ge 0, \quad i = 1, \ldots, N. \tag{10d}$$

To solve this problem, its (primal) Lagrangian

$$\begin{aligned} L_{\mathrm{p}}(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \boldsymbol{\alpha}, \boldsymbol{\alpha}^*, \boldsymbol{\beta}, \boldsymbol{\beta}^*) &= \\ \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) &- \sum_{i=1}^{N}(\beta_i\xi_i + \beta_i^*\xi_i^*) \\ - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - y_i + \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + b)& \\ - \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i - b)& \end{aligned} \tag{11}$$

is needed. The *dual* variables $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^*$, $\boldsymbol{\beta}$ and $\boldsymbol{\beta}^*$ are the Lagrange multipliers of the primal constraints. According to the Karush-Kuhn-Tucker (KKT) conditions, the derivatives with respect to the *primal* variables must vanish in the optimum:

$$\frac{\partial L_{\mathrm{p}}}{\partial \mathbf{w}} = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)\mathbf{x}_i \tag{12a}$$

$$\frac{\partial L_{\mathrm{p}}}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) = 0 \tag{12b}$$

$$\frac{\partial L_{\mathrm{p}}}{\partial \xi_i} = 0 \quad \Rightarrow \quad \alpha_i + \beta_i = C, \quad i = 1, \ldots, N \tag{12c}$$

$$\frac{\partial L_{\mathrm{p}}}{\partial \xi_i^*} = 0 \quad \Rightarrow \quad \alpha_i^* + \beta_i^* = C, \quad i = 1, \ldots, N \tag{12d}$$

This yields a dual objective function that is solely dependent on the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$:

$$\begin{aligned} J_{\mathrm{d}}(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*) = &\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j \\ &- \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)y_i + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) \end{aligned} \tag{13}$$

To solve *nonlinear* regression problems, the input space is mapped into a *feature space* by a nonlinear mapping $\boldsymbol{\Phi}$. The linear SVM is then applied to the features $\boldsymbol{\Phi}(\mathbf{x})$ instead of the original regressors $\mathbf{x}$. Since the regressors occur in Eq. (13) only as scalar product $\mathbf{x}_i^{\mathrm{T}}\mathbf{x}_j$, we define a *kernel function*

$$K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{x})\boldsymbol{\Phi}(\mathbf{x}')\,. \tag{14}$$

Consequently, the only difference between linear and nonlinear SVMs is the choice of kernel functions instead of scalar products. Among the variety of possible kernel function (Schölkopf and Smola, 2002), only the *Gauss kernel* is considered in the following since it is identical to an RBF neuron's activation function (6). Furthermore, the dual variables $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ are substituted by the SVM *coefficients* $\mathbf{a}$ because they are not independent (due to $\alpha_i\alpha_i^* = 0$):

$$a_i = \alpha_i - \alpha_i^* \tag{15a}$$
$$|a_i| = \alpha_i + \alpha_i^* \tag{15b}$$

This modification reduces the number of variables from $2N$ to N, accelerates the optimization method described in Sec. 4 and leads to an optimization problem very similar to that of SVM *classification* (Vapnik, 1995):

$$\min_{\mathbf{a}} \quad J_\mathrm{d} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_j K_{ij} - \sum_{i=1}^{N} a_i y_i + \varepsilon \sum_{i=1}^{N} |a_i| \tag{16a}$$
$$\text{s.t.} \quad -C \le a_i \le C\,, \quad i = 1,\dots,N \tag{16b}$$
$$\sum_{i=1}^{N} a_i = 0 \tag{16c}$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. The SVM output is computed as

$$f(\mathbf{x}) = \sum_{a_i \neq 0} a_i K(\mathbf{x}_i, \mathbf{x}) + b \tag{17}$$

with the *bias term b* which provides an constant offset to f and is computed from the KKT conditions, see (Vapnik, 1995) and Fig. 3. Vectors $\mathbf{x}_i$ with $a_i \neq 0$ are called *support vectors* (SVs); Eq. (17) is the *support vector expansion* and is (except for the bias term) identical to Eq. (5). Usually only a small fraction of the data set are support vectors, typically about 10%.

For *positive definite* kernel functions, the bias term b can be kept fixed or even omitted (Poggio *et al.*, 2002). The missing equality constraint (12b) leads to an alternative formulation of the dual problem:

$$\begin{aligned}\min_{\mathbf{a}} \quad J_\mathrm{d}(\mathbf{a}) =& \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} a_i a_i K_{ij} - \sum_{i=1}^{N} a_i y_i \\ &+ \varepsilon \sum_{i=1}^{N} |a_i| + b \sum_{i=1}^{N} a_i\end{aligned} \tag{18a}$$
$$\text{s.t.} \quad -C \le a_i \le C\,, \quad i = 1,\dots,N \tag{18b}$$

Since Eq. (18) contains only *box constraints*, it is one of the most convenient QP cases leading to a simpler optimization algorithm. If the bias term is omitted ("no-bias SVM", $b = 0$), the last term in Eq. (18a) vanishes.

4. SOLVING THE SVM REGRESSION PROBLEM

4.1 Optimization algorithm

To solve the optimization problem (16) a method based on the *Sequential Minimal Optimization* (SMO) algorithm (Platt, 1999) is used. According to Osuna's theorem (Osuna *et al.*, 1997) the sequential solution of subproblems leads to the minimization of the original problem. Using this to an extreme, SMO reduces the optimization problem to smallest possible subproblems in order to perform an analytical update step. In case of a variable bias term b two variables are needed. A detailed derivation of the update step for regression problems can be found in (Flake and Lawrence, 2002). Without loss of generality the following assumes that a_1 and a_2 are updated. All variables not involved in the current step remain unchanged

$$a_i = a_i^\mathrm{old}\,,\; i = 3,\dots,N\,. \tag{19}$$

The two variables needed to perform the step are isolated in the cost function (16a) from variables $a_3,\dots,a_N$ combined in the constant c.

$$\begin{aligned} J_\mathrm{d} =& \frac{1}{2}a_1^2 K_{11} + a_1 a_2 K_{12} + \frac{1}{2}a_2^2 K_{22} \\ &+ \varepsilon |a_1| + \varepsilon |a_2| \\ &+ a_1(E_1 - b - a_1^\mathrm{old}K_{11} - a_2^\mathrm{old}K_{21}) \\ &+ a_2(E_2 - b - a_1^\mathrm{old}K_{12} - a_2^\mathrm{old}K_{22}) + c\,, \end{aligned} \tag{20}$$

where E_j is the prediction error of the SVM in the previous step

$$E_j = f(\mathbf{x}_j) - y_j = \sum_{i=1}^{N} a_i^\mathrm{old} K_{ij} + b - y_j\,. \tag{21}$$

The equality constraint (16c) forces the updated variables to fulfill

$$a_1 + a_2 = a_1^\mathrm{old} + a_2^\mathrm{old} = -\sum_{i=3}^{N} a_i = \gamma\,. \tag{22}$$

Using this equation, a_2 can be eliminated from Eq. (20) which leads to:

$$\begin{aligned} J_\mathrm{d} =& \frac{1}{2}\eta a_1^2 + (E_1 - E_2 - \eta a_1^\mathrm{old})a_1 \\ &+ \varepsilon(|a_1| + |\gamma - a_1|) + c \end{aligned} \tag{23}$$

with

$$\eta = K_{11} - 2K_{12} + K_{22}\,. \tag{24}$$

The derivative of Eq. (23) with respect to a_1 has to be 0 at the optimum. The update rule for a_1 can be found to

$$a_1 = a_1^\mathrm{old} - \frac{1}{\eta}(E_1 - E_2 - \varepsilon(\mathrm{sign}(a_1) + \mathrm{sign}(\gamma - a_1)))\,. \tag{25}$$

The a_1 found with this equation further has to fulfill the inequality constraint which is easily achieved using a proper clipping step. The variable a_2 is found using Eq. (22)

For a *fixed* bias term only a single variable a_1 is affected in each update step. The belonging update rule is received in an analogous way:

$$a_1 = a_1^\mathrm{old} - \frac{E_1 + \varepsilon\,\mathrm{sign}(a_1)}{K_{11}}\,. \tag{26}$$

To perform an update step, SMO needs to chose appropriate variables. Therefore a set of heuristics is used, as described in (Platt, 1999). This leads to an easy to implement algorithm.

However, a major source of inefficiency is included in the original algorithm concerning the calculation of the bias term. This is described in (Keerthi *et al.*, 2001) and (Shevade *et al.*, 1999). To overcome the inefficiencies of SMO, the Lagrangian L_d of the dual problem is used in the algorithm together with two bounds for the bias term. In the optimum, these two bounds equal the bias term b, not during the iteration process. But, it is not necessary to know the exact bias b to perform an update step as can be seen from Eq. (25) together with Eq. (21). Some of the heuristics used in the original SMO are replaced by a proper method of choosing the variables. These changes lead to a more reliable, exact and significantly faster version of the SMO. These modifications are used together with the formulation of SVMs in coefficient form (16), (17) and (18).

A kernel cache can be used to decrease the runtime of the algorithm as described in (Flake and Lawrence, 2002), where a *least recently used* policy is applied to update the cache. This strategy is used as SMO tends to access the kernel matrix in an unstructured manner. However, here a different strategy is used as it is possible to increase the regularity of the kernel matrix using an index vector. A cache based on this structured kernel matrix can focus on regions with a higher repetition rate of kernel accesses. This index vector $\mathbf{i}$ is structured concerning the type of a variable a_{i_j}

$$i_1,\dots,i_p \quad \text{with} \quad 0 < |a_{i_j}| < C \tag{27a}$$
$$i_{p+1},\dots,i_q \quad \text{with} \quad |a_{i_j}| = C \tag{27b}$$
$$i_{q+1},\dots,i_N \quad \text{with} \quad a_{i_j} = 0\,. \tag{27c}$$

After rearranging the kernel matrix according to $\mathbf{i}$ the first q columns represent a kernel computation between a SV and any other vector. Experiments have shown that this is the case for up to 99% of all kernel computations. As in normal cases the fraction of SVs is about 10% it is now possible, while storing just 10% of the kernel matrix, providing space for up to 99% of needed kernels. As the number of SVs is not known a-priori, space is reserved for the first m columns of the kernel matrix. This method leads to a very simple but efficient caching structure. Saving of time is achieved due to the avoidance of kernel recalculations.

4.2 A reduction method for SVMs

SVMs tend to be more complex than models computed by other methods, e.g., the OLS algorithm described in Sec. 3. This applies in particular to large data sets since the number q of support vectors loosely depends on the size of the data set. The following describes a possible strategy to reduce q.

A method is needed to reduce the model complexity without losing the generalization abilities of the SVM. Therefore an additional reduction step is introduced to shrink the size of a pre-computed SVM, which is obtained out of the original data set $\mathcal{D}$ with

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\},\ i = 1,\dots,N,\ \mathbf{x}_i \in \mathbb{R}^n,\ y_i \in \mathbb{R}, \tag{28}$$

a kernel function K, a trade-off parameter C and the width ε of the insensitive loss function. The solution of the appropriate quadratic program leads to Eq. (17). Now the elements of Eq. (28) can be divided in three subsets.

$$\mathcal{A}_0 = \{(\mathbf{x}_i, y_i) \mid a_i = 0\} \tag{29a}$$
$$\mathcal{F} = \{(\mathbf{x}_i, y_i) \mid 0 < |a_i| < C\} \tag{29b}$$
$$\mathcal{A}_C = \{(\mathbf{x}_i, y_i) \mid |a_i| = C\}\,. \tag{29c}$$

Subset $\mathcal{A}_0$ contains all points inside the insensitive zone, depicted as stars in Fig. 3. SVs exactly on the margin, depicted as filled circles, are contained in subset $\mathcal{F}$. The third subset $\mathcal{A}_C$ consists of the SVs lying outside the insensitive zone, depicted as circles. Dependent on the chosen ε most SVs belong to this third subset. At this point starts the additional reduction step. A condensed data set $\mathcal{D}_{\text{con}}$ is created which contains only the subsets $\mathcal{A}_0$ and $\mathcal{F}$

$$\mathcal{D}_{\text{con}} = \mathcal{A}_0 \cup \mathcal{F}\,, \tag{30}$$

leaving out all data point lying outside the insensitive zone. For $\mathcal{D}_{\text{con}}$ it is possible to find a solution with all data points lying inside the ε-insensitive zone or exactly on the margin. To find this reduced solution $C = \infty$ has to be chosen, this is reasonable as all outliers are eliminated. With the original kernel function and ε this leads to an approximation of the original solution. In the solution obtained all SVs of subset $\mathcal{A}_C$ are eliminated, the size of subset $\mathcal{F}$ may slightly increase, but the overall number of SVs decreases significantly.

The additional reduction step leading to a reduced SVM does not change the overall behavior of the original SVM computed on the original data set. This may surprise as the C is changed. But, leaving out the subset $\mathcal{A}_C$ of the original data set the form of the original solution is conserved. Using $C = \infty$ in the additional reduction step assures to approximate the behavior of the original SVM.

5. EXPERIMENTAL RESULTS

Based on (3) and (4) a first order discrete-time dynamical model has been selected

$$p_{\text{A}}(k) = f(p_{\text{A}}(k-1), p_{\text{B}}(k), p_{\text{P}}(k), x(k), \dot{x}(k), x_{\text{V}}(k))\,, \tag{31}$$

where f will be modelled by the three model structures described in Sec. 3. For that, the process has been dynamically excited for different loads; the piston velocity $\dot{x}$ is computed numerically from the displacement x. The resulting signals are then split into a training data set comprising 2500 samples, and a validation data set. Table 1 shows the best results for different sampling frequencies. The *root-mean-square errors*

Table 1. Validation RMSE for different sampling frequencies

f_s	RBF	SVM	LOLIMOT
500 Hz	4.33%	3.93%	2.94%
1000 Hz	5.03%	4.81%	3.51%
2000 Hz	5.48%	5.35%	3.37%

Table 2. Models for $f_s = 500\,\text{Hz}$

	RBF	SVM	LOLIMOT
RMSE	4.33%	3.93%	2.94%
Width σ	0.6	0.5	-
Size q	101	749	35

Table 3. Different SVM types

	standard	std./red.	fixed	fixed/red.
RMSE	4.01%	4.07%	3,93%	3,96%
Size q	896	261	749	227

have been determined by simulating the models on the validation data set. Obviously, all models have the best generalization performance for $f_s = 500\,\text{Hz}$, so the detailed results for this choice can be found in Tab. 2. Whereas LOLIMOT has the smallest validation error, it can be observed that the performance of the RBF network can be improved if it is replaced by a SVM with the identical structure. However, SVMs usually lead to much more complex models than RBF networks. Regarding LOLIMOT's size, it should be kept in mind that each of its "neurons" contains a linear submodel and is therefore *not* equivalent to a RBF neuron (or a support vector, respectively). The parameters of the SVM have been chosen as $\varepsilon = 0.008$ and $C = 4.6$.

To reduce the SVM's size, the reduction method described in Sec. 4.2 is applied, see Tab. 3. The results show that both standard SVMs with variable bias and SVMs with fixed bias ($b = 0$) can be cut down to 30% of the model complexity, without losing generalization abilities.

6. CONCLUSIONS

Three model types have been studied for their applicability as nonlinear dynamical models of a hydraulic cylinder: RBF networks, support vector machines and local-linear model trees (LOLIMOT). In general, all models were able to grasp the cylinder's behavior.

The best results are obtained by LOLIMOT since its local submodels are an adequate structure for this type of process. However, by the use of SVMs the generalization abilities of the RBF network can be improved without changing its structure. For the computation of SVMs a novel implementation of the SMO regression algorithm has been proposed as well as a reduction method that significantly reduces the SVM's complexity, i.e., the number of support vectors.

The areas of future work comprise all topics addressed in this study: Improved optimization algorithms, SVM reduction techniques as well as the application of the SVM principle to model structures other than the basis function network considered so far.

REFERENCES

Chen, Sheng, C. F. N. Cowan and P. M. Grant (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks* **2**(2), 302–309.

Flake, Gary William and Steve Lawrence (2002). Efficient SVM regression training with SMO. *Machine Learning* **46**(1), 271–290.

Isermann, Rolf (2003). *Mechatronic Systems: Fundamentals*. Springer Verlag. UK.

Keerthi, S. Sathiya et al. (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation* **13**, 637–649.

Kreß, Richard (2002). Robuste Fehlerdiagnoseverfahren zur Wartung und Serienabnahme elektrohydraulischer Aktuatoren. PhD thesis. TU Darmstadt, Fachbereich Maschinenbau. Darmstadt.

Kubbat, Wolfgang, Richard Kreß and Volker van Lier (2000). Neue Konzepte für elektrohydraulische Aktoren der primären Flugsteuerung. *Thema Forschung* **1**, 140–149.

Muenchhof, Marco and Rolf Isermann (2004). Zustandsüberwachung für hydraulische Proportionalwegeventile. In: *Proceedings of the 2. Paderborner Workshop Intelligente mechatronische Systeme*. Paderborn (Germany).

Nelles, Oliver (2000). *Nonlinear System Identification*. Springer-Verlag. Berlin.

Osuna, Edgar, Robert Freund and Federico Girosi (1997). An improved training algorithm for support vector machines. In: *Neural Networks for Signal Processing VII — Proceedings of the 1997 IEEE Workshop* (J. Principe, L. Gile, N. Morgan and E. Wilson, Eds.). IEEE. New York. pp. 276 – 285.

Platt, John C. (1999). Fast training of support vector machines using sequential minimal optimization. In: *Advances in Kernel Methods – Support Vector Learning* (Bernhard Schölkopf, Christopher J. C. Burges and Alexander Smola, Eds.). Chap. 12. MIT Press. Cambridge, MA.

Poggio, Tomaso et al. (2002). b.. In: *Uncertainty in Geometric Computations* (Joab Winkler and Mahesan Niranjan, Eds.). Chap. 11, pp. 131–141. Kluwer Academic Publishers. Boston.

Ramdén, Teresia (1998). Condition Monitoring and Fault Diagnosis of Fluid Power Systems : Dissertation No 514. PhD thesis. Linköping University, Sweden. Linköping.

Schölkopf, Berhhard and Alexander J. Smola (2002). *Lerning with Kernels*. Adaptive Computation and Machine Lerning. The MIT Press. Cambridge, MA.

Shevade, Shirish Krishnaj et al. (1999). Improvement to SMO algorithm for SVM regression. Technical Report CD-99-16. Control Devision, National University of Singapore. Singapore.

Vapnik, Vladimir N. (1995). *The Nature of Statistical Learning Theory*. Information Science and Statistics. Springer-Verlag. New York.