# EXPLICIT MODEL PREDICTIVE CONTROL OF A HYBRID SYSTEM USING SUPPORT VECTOR MACHINES

**Bernt M. Åkesson** [1] **Mats J. Nikus** [2] **Hannu T. Toivonen**

*Process Control Laboratory*
*Faculty of Technology*
*Åbo Akademi University*
*FI-20500 Turku, Finland*

Abstract: The objective of this work is to study the capabilities of support vector machines for approximating the complex control law arising from model predictive control of a hybrid MIMO-system. By approximating the control law, an explicit formulation can be obtained, which is computationally less intensive for on-line use. The explicit model predictive control approach is applied to a simulated hybrid system consisting of two tanks in series. The system has real-valued, integer-valued, and binary-valued control inputs. A model predictive controller is first designed to control the entire system. This controller is then approximated using support vector machines, with separate approximators for each control input. Reasonable control results were achieved with the approximators. *Copyright © 2006 IFAC*

Keywords: model predictive control, hybrid systems, support vector machines, approximations

## 1. INTRODUCTION

The continuous increase of the computational power of computers has turned the attention of the control community to problems that previously were intractable. Among these problems, the control of hybrid systems can be found. Classically these problems were tackled by dividing the system into integer and continuous parts, and designing separate controllers for the two parts. Naturally the resulting controller cannot be optimal.

Optimal control of these systems is, still today, not possible but a model predictive controller (MPC) can give a near-optimal solution. However, many problems of this kind remain impossible to solve on-line with MPC. To overcome this problem approximation techniques can be used. In this paper we study the use of explicit model predictive controllers, which is a technique that has previously been used for some non-hybrid systems (Parisini and Zoppoli, 1995; Åkesson *et al.*, 2005).

## 2. MODEL PREDICTIVE CONTROL OF HYBRID SYSTEMS

Hybrid systems describe processes whose behavior can be characterized both by continuous- and/or discrete-time dynamics as well as discrete events governed by logic rules. The state of a hybrid system consists of both real-valued variables and discrete-valued, or logical variables. Due to the mixed continuous/discrete and logic dynamics, it is very hard to design optimal controllers for hybrid systems.

[1] Supported by the Finnish Graduate School in Chemical Engineering (GSCE)

[2] Supported by the Academy of Finland, grant 206750. Present address: Helsinki University of Technology, Laboratory of Process Control and Automation, 02015 Espoo, Finland.

Hybrid systems with linear dynamics can however be transformed to a piecewise affine (PWA) system (Bemporad, 2004) for which optimal controller synthesis techniques can be applied. Application of the popular MPC strategy to hybrid systems gives a useful suboptimal controller (Bemporad and Morari, 1999). In MPC the predicted cost function over a future control horizon is minimized at each sampling instant with respect to the control sequence. Model predictive control of nonlinear hybrid systems requires the on-line solution of a mixed integer nonlinear programming (MINLP) problem at each sampling period. For many realistic processes this may, however, result in a prohibitively high computational burden, and the MPC strategy cannot be solved in real time.

The MPC strategy for a hybrid system is defined by the following general optimization problem

$$\min_{U_{n_u}(k|k)} J(x(k+j|k), u(k+j|k), \Delta u(k+j|k)) \tag{1}$$

subject to

$$x(k+j+1|k) = f(x(k+j|k), u(k+j|k)) \tag{2a}$$

$$x_{\min} \le x(k+j+1|k) \le x_{\max}, \quad j = 0, \ldots, n_p - 1 \tag{2b}$$

$$u(k+j|k) \in [u_{c,\min}, u_{c,\max}] \times \{u_{d,\min}, \ldots, u_{d,\max}\}, \quad j = 0, \ldots, n_u - 1 \tag{2c}$$

where $U_{n_u}(k|k)$ is a sequence of control inputs, $u(k|k), \ldots, u(k+n_u-1|k)$. Here $u(k+j|k)$ denotes the input $u(k+j)$ calculated from information available at time instant $k$. Furthermore, $\Delta u(k+j|k) = u(k+j|k) - u(k+j-1|k)$. The cost function in (1) typically includes setpoint deviations and input variance. The process behavior is predicted by using the process model (2a). Possible hard constraints on state variables and inputs are included in (2b) and (2c), respectively. The optimization problem is solved at each sampling instant and the first input in the sequence is implemented.

## 3. EXPLICIT MPC

One approach for reducing the on-line computations of MPC is to describe the strategy defined by the MPC using a function approximator, such as an artificial neural network (Parisini and Zoppoli, 1995). The MPC optimizations can then be performed off line to generate the training data, whereas only the approximator is applied in real-time operation.

### 3.1 Mathematical formulation

The MPC strategy can be characterized in the form:

$$u_i^c(k) = g_{i,MPC}^c(x(k)) \tag{3}$$

$$u_j^d(k) = \mathrm{dsc}(g_{j,MPC}^d(x(k))) \tag{4}$$

where the functions $g_{i,MPC}^c(x(k))$ and $g_{j,MPC}^d(x(k)))$, implicitly defined by the MPC optimization, describe the functional relationship of the continuous- and discrete-valued control signals, respectively. The dsc operator gives only discrete values to the output. The approximated MPC can be described as:

$$u_i^c(k) = g_{i,APP}^c(x(k); w^c) \tag{5}$$

$$u_j^d(k) = \mathrm{dsc}(g_{j,APP}^d(x(k); w^d)) \tag{6}$$

where $w^c$ and $w^d$ denote vectors of parameters that need to be determined. The approximation errors are defined as:

$$\Delta_i^c(x) = g_{i,APP}^c(x; w^c) - g_{i,MPC}^c(x) \tag{7}$$

$$\Delta_j^d(x) = \mathrm{dsc}(g_{j,APP}^d(x; w^d)) - \mathrm{dsc}(g_{j,MPC}^d(x)) \tag{8}$$

The objective is to minimize the approximation errors for a set of training data, generated with the MPC controller, by adjusting the parameters and structure of the functions $g_{i,APP}^c(x; w^c)$ and $g_{j,APP}^d(x; w^d)$.

### 3.2 The approximators

It was decided to to use support vector machines (Vapnik, 1998) for the approximation task instead of neural networks in this study. Support vector machines (SVM) were introduced by Vapnik in the 1990s and can be used for solving classification and regression tasks. They are based on the principle of structural risk minimization, which ensures that the model complexity is not too high by minimizing the so called VC-dimension (Vapnik, 1998). Within the SVM framework radial basis networks, feedforward neural networks as well as other kinds of models can be set up. A nice property of the SVM is that it yields a unique optimal solution of the resulting optimization problem.

When solving the classification problems, a decision surface of the form $w^T \phi(x) + b = 0$ is sought. The basis function $\phi(x)$ maps the inputs to a high dimensional feature space. In the support vector formulation the cost function

$$\min_{w,\, b,\, \xi} \{\frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i\} \tag{9}$$

is minimized subject to the constraints

$$y_i(w^T \phi(x_i) + b) \ge 1 - \xi_i \tag{10a}$$

$$\xi_i \ge 0, \quad i = 1, \ldots, l \tag{10b}$$

where $x_i$ represents an input vector in the data set and $y_i$ $(= \pm 1)$ the corresponding scalar output. In practice the solution of the SVM optimization problem is solved by introducing a dual problem that arises after the inclusion of Lagrange multipliers. For the classification case this yields the decision function

$$f_d(x) = \mathrm{sgn}(\sum_{i=1}^{l} y_i \alpha_i \phi(x_i)^T \phi(x) + b) \tag{11}$$

where the $\alpha_i$ are the Lagrange multipliers of the dual problem and $l$ is the number of support vectors. The support vectors are data vectors selected from the training set to form the basis of the model. When solving the dual problem it turns out that the basis function $\phi(x_i)$ is only present as an inner product in the solution

$$K(x_i, x) = \phi(x_i)^T \phi(x) \tag{12}$$

Hence only the kernel function $K(x_i, x)$ needs to be known and the basis functions $\phi(x_i)$ are not used explicitly (the “kernel trick”).

Several kernels have been proposed and the following are just examples sampled from the plethora.

$$K(x_i, x) = \exp(-\gamma \|x - x_i\|^2) \tag{13a}$$
$$K(x_i, x) = \tanh(\beta_0 x^T x_i + \beta_1) \tag{13b}$$
$$K(x_i, x) = (\beta_0 x^T x_i + \beta_1)^p \tag{13c}$$
$$K(x_i, x) = x^T x_i \tag{13d}$$

The gaussian kernel (13a) gives rise to a radial basis network. With a sigmoidal kernel (13b) perceptron networks much like like feedforward 2-layer neural networks can be designed within the SVM framework. Also polynomial and linear kernels (13c, 13d) can be used.

Sometimes classification problems include multiple classes whereas the standard SVM formulation only handles two. One way to solve the multi-class problem is to use the so-called one-against-one scheme with voting (Hsu and Lin, 2002). In general this means that

$$n_{\mathrm{m}} = \begin{pmatrix} n_{cl} \\ 2 \end{pmatrix} = \frac{n_{cl}(n_{cl} - 1)}{2} \tag{14}$$

machines have to be trained, where $n_{cl}$ is the number of classes. Out of the $n_{cl}$ classes, the class that gets the most votes will be output from the classifier.

Regression problems, arising from the approximation of the continuous control variables, have the primal cost function

$$\min_{w,b,\xi} \{\frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i + C \sum_{i=1}^{l} \xi_i^*\} \tag{15}$$

subject to the constraints

$$w^T \phi(x_i) + b - y_i \leq \varepsilon + \xi_i \tag{16a}$$
$$y_i - w^T \phi(x_i) + b \leq \varepsilon + \xi_i^* \tag{16b}$$
$$\xi_i, \xi_i^* \geq 0, \ \ i = 1, \ldots, l \tag{16c}$$

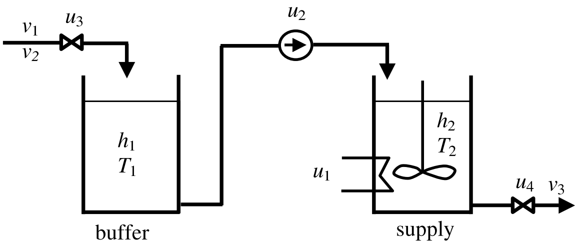

Fig. 1. Schematic of the hybrid example process.

For these cases the approximation function becomes

$$f(x) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) K(x_i, x) + b \tag{17}$$

## 4. EXAMPLE

In this work the explicit MPC approach for hybrid systems is applied to a two-tank configuration depicted in Fig. 1. The control objective is to keep the temperature in the second tank at its setpoint while keeping the levels of the tanks within preset limits. The system, which was first studied by Slupphaug *et al.* (1998), is controlled by two on/off valves, a heater and a pump. The pump has three operational levels (off, medium, high). The temperature in the second tank of the studied simulated system is to be setpoint controlled while the two levels are allowed to vary between preset limits (hard constraints). The temperature in the first tank is not controlled. The system is described by the mass and energy balances (Slupphaug *et al.*, 1998)

$$\dot{h}_1 = \frac{1}{A_b}(v_1 u_3^d - \alpha u_2^d) \tag{18a}$$
$$\dot{T}_1 = \frac{1}{A_b h_1}(v_2 - T_1) v_1 u_3^d \tag{18b}$$
$$\dot{h}_2 = \frac{1}{A_s}(\alpha u_2^d - v_3 u_4^d) \tag{18c}$$
$$\dot{T}_2 = \frac{1}{A_s h_2}((T_1 - T_2)\alpha u_2^d + \frac{u_1^c}{c_l \rho_l}) \tag{18d}$$

where the variables are defined in Table 1 and the model parameters are given in Table 2.

Table 1. States, inputs and disturbances

| | |
|---|---|
| $h_1, h_2$ | Buffer and supply levels |
| $T_1, T_2$ | Buffer and supply temperatures |
| $u_1^c$ | Heater (superscript $c$ for continuous) |
| $u_2^d$ | Pump (superscript $d$ for discrete) |
| $u_3^d, u_4^d$ | Inlet- and outlet-valves |
| $v_1$ | Inflow |
| $v_2$ | Temperature of inflow |
| $v_3$ | Outflow |

By defining a state vector $x = [h_1, T_1, h_2, T_2]^T$, along with inputs $u$ and disturbances $v$, the process model (18) can be written compactly as

$$\dot{x} = f(x, u, v) \tag{19}$$

Table 2. Model parameters

| | | |
|---|---|---|
| $A_b$ | 3.5 m$^2$ | Buffer area |
| $A_s$ | 2 m$^2$ | Supply area |
| $c_l$ | 4.2 kJ/kgK | Specific liquid heat capacity |
| $\rho_l$ | 1000 kg/m$^3$ | Liquid density |
| $\alpha$ | 1 m$^3$/min | Pump capacity factor |

which can be discretized by using Euler's approximation,

$$x(k+1) = x(k) + \sigma f(x(k), u(k), v(k)) \qquad (20)$$

where $\sigma = 15$ s is the sampling time. In this study it is assumed that the disturbaces $v(k)$ are unmeasured and at their nominal values, $v_0$.

The MPC strategy can be defined similarly to (1). Here we have the optimization problem

$$\min_{u(k|k),\ldots,u(k+n_u-1|k)} \{J_1 + J_2 + J_3\} \qquad (21)$$

subject to

$$h_{\min} \leq h_1(k+j|k) \leq h_{\max}, \quad j = 1,\ldots,n_p \quad (22a)$$

$$h_{\min} \leq h_2(k+j|k) \leq h_{\max}, \quad j = 1,\ldots,n_p \quad (22b)$$

$$u(k+j|k) \in [0, u_{1,\max}] \times \{0,1,2\} \times \{0,1\}^2, \quad j = 0,\ldots,n_u-1 \qquad (22c)$$

$$u(k+j|k) = u(k+j-1|k), \quad j = n_u,\ldots,n_p-1 \qquad (22d)$$

The individual terms in the cost function are defined as

$$J_1 = \sum_{j=0}^{n_p-1} (T_2(k+1+j|k) - T_{\text{set}}(k))^2 \qquad (23a)$$

$$J_2 = \sum_{j=0}^{n_u-1} (u^d(k+j|k) - u_0^d)^T R (u^d(k+j|k) - u_0^d) (23b)$$

$$J_3 = \sum_{j=0}^{n_u-1} \Delta u(k+j|k)^T Q \Delta u(k+j|k) \qquad (23c)$$

$J_1$ and $J_3$ denote the costs for setpoint deviations and variations in the inputs, respectively. An additional penalty function $J_2$ is included in order to avoid the trivial solution $u^d(k) = 0$, i.e. valves closed and pump turned off. This cost penalizes deviations from nominal input values, defined as $u_0^d = 1$.

The weighting matrices $Q$ and $R$ were chosen as diag($4/u_{1,\max}^2$, 1, 1, 1) and diag(0.1, 2, 2), respectively. The prediction horizon $n_p = 10$ and control horizon $n_u = 4$ were used. The level constraints were set to $h_{\min} = 1$ m and $h_{\max} = 9$ m and the input constraint to $u_{1,\max} = 560$ kW.

The process was simulated in MATLAB/Simulink. The MPC used an MINLP optimizer (Kuipers, 2000) based on the branch and bound algorithm.

Figs 2 and 3 illustrate the behavior of the system when controlled by the MPC controller. The system starts at the initial state $x(0) = [7, 18, 1.5, 22]^T$, with the setpoint $T_{\text{set}} = 20$ °C and the disturbances $v = [1.5, 18, 1]^T$. At $t = 25$ min the setpoint changes from 20 °C to 22 °C and at $t = 75$ min it changes back to 20 °C. A step change in the feed flow $v_1$ occurs at $t = 50$ min from 1.5 m$^3$/min to 0.5 m$^3$/min. Finally, at $t = 125$ min, the feed temperature $v_2$ decreases from 18 °C to 16 °C.

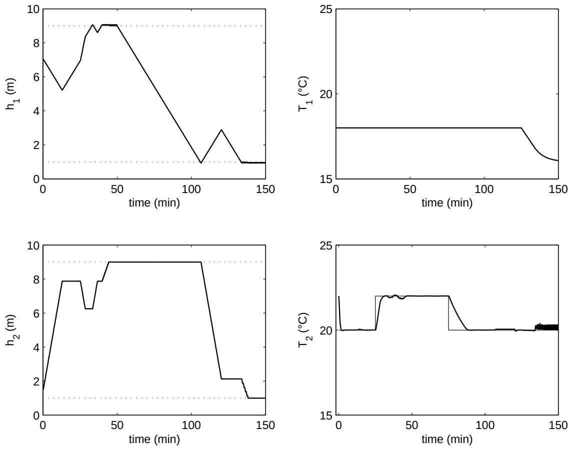

Fig. 2. Simulation results with MPC: states.

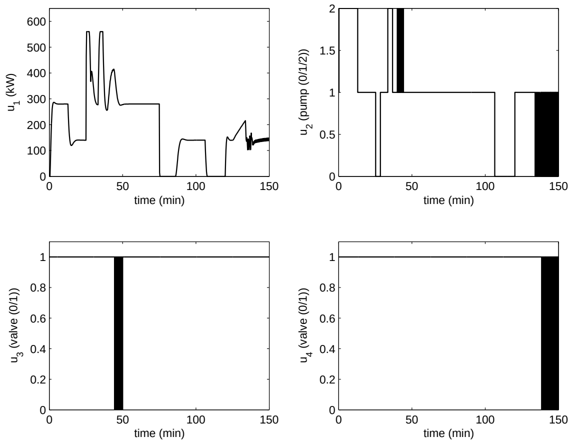

Fig. 3. Simulation results with MPC: control inputs.

Using the model predictive controller, a set of training data consisting of 5400 patterns was generated. Each control input was approximated separately. From inspection of the training data for the inlet valve $u_3^d$, it was seen that the underlying control law could be described exactly as a simple on/off controller with feedback from the level $h_1$ of the buffer tank. A similar approach was used by Slupphaug *et al.* (1998). However, in our case a small dead-band of 32 cm was introduced in order to avoid excessive valve movement. The controller is implemented as

```
if $h_1(k) > 8.92$
    $u_3(k) = 0$
else if $h_1(k) < 8.6$
    $u_3(k) = 1$                                  (24)
else
    $u_3(k) = u_3(k-1)$
end
```

Table 3. Number of parameters in the approximations.

| $u_1^c$ | $u_2^d$ | $u_3^d$ | $u_4^d$ |
|---|---|---|---|
| 1745 | 1836 | 2 | 81 |

The SVM that were to compute $u_2(k)$ was given a total of nine inputs: the state vector, the setpoint and previous input values. In addition to these inputs, the support vector machines controlling the heater and the outlet valve were given the present pumping level $u_2^d(k)$, as this information clearly is relevant for the temperature and level control of the second tank.

For the binary and integer inputs, support vector classifiers were used, and the continuous control input was approximated by support vector regression. The training of the SVMs was performed with the LIBSVM software (Chang and Lin, 2001), which uses a numerically powerful decomposition method much like the method by Platt (1999). The decomposition method for the SVMs made it possible to use quite large data sets.

There are some open parameters in the SVMs that need to be determined outside of the main optimization. They are the width parameter of the gaussians $\gamma$, the weighting factor between model complexity and performance $C$, and for the regression case also $\varepsilon$, i.e. the threshold value for the residuals to contribute to the cost function (cf. equations (16a) and (16b)). The best values for the first two parameters were determined by performing a grid search on test data. For the regression model the threshold $\varepsilon = 0.1$ was used, and the optimum $\gamma$ was 0.03 (C = 1000). For the pump $\gamma = 0.01$ ($C = 10$) was used and for the outlet valve a linear kernel was used ($C = 100$).

The number of parameters in the SVM controllers are given in Table 3. The SVMs tend to give rather large models with the number of support vectors as high as one third of the number of patterns. These large models are somewhat computationally intensive but the SVM computations are still fast compared to the MPC computations.

The approximators were tested on the process by simulating the same setpoint changes and disturbances that were tested on the MPC. As can be seen in Fig. 4 the SVM controller has succeeded in controlling the temperature to its setpoint while keeping the tank levels within the allowed limits. The corresponding control inputs are shown in Fig. 5.

The total and individual costs incurred during the simulation sequence are summarized in Table 4. The total costs are comparable for the two controllers, but interestingly the SVM controller actually has a lower cost than the MPC controller in this simulation. This is largely attributed to the high input variance cost $J_3$ of the MPC. The SVM controller, on the other hand, has a higher cost for the deviation from the nominal state of the inputs, $J_2$. Since the MPC has a limited prediction horizon it is entirely possible for another controller to outperform it on an actual simulation sequence. However, in this case the other controller was trained to mimic the behavior of the MPC and in that respect the results are a little surprising. The fact that the SVM model has approximation errors can explain these result even though it seems more likely that an approximation error would cause a deterioration of the control results — not an improvement.

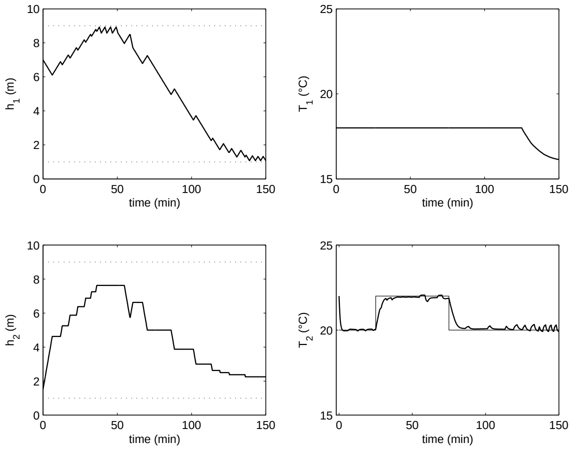

Fig. 4. Simulation results with SVM control: states.

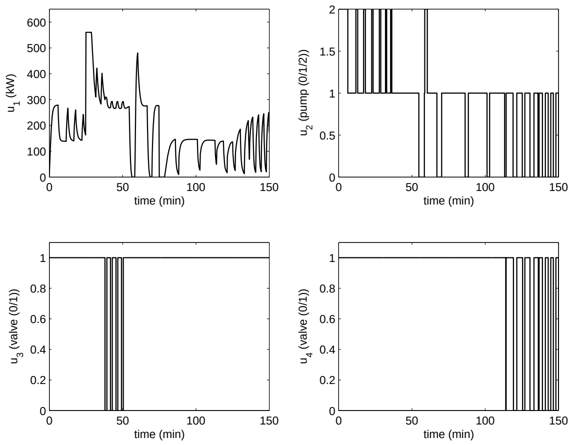

Fig. 5. Simulation results with SVM control: control inputs.

Table 4. Costs using the two controllers.

| | MPC | SVM |
|---|---|---|
| $J_1$ | 100.77 | 70.32 |
| $J_2$ | 81.40 | 159.80 |
| $J_3$ | 149.58 | 71.64 |
| $J_{\text{tot}}$ | 331.75 | 301.76 |

Another way to compare the SVM controller with the MPC is to calculate the minimum of the cost function in (21) at each sampling instant. This is illustrated in Fig. 6. It is seen that the MPC cost is lower for the MPC than for the SVM controller, which is to be expected since the MPC optimizes this particular cost. For some samples at the end of the run, however, the cost for the SVM is lower than that of the MPC. The explanation for this is that the MPC does not allow any constraint violations during the prediction horizon

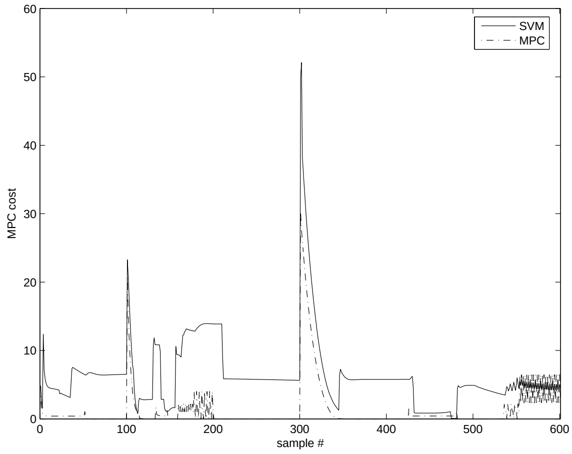

Fig. 6. Minimum of the MPC cost function at each time step in the simulation sequence. The dash-dotted line represents the MPC and the solid line the SVM controller.

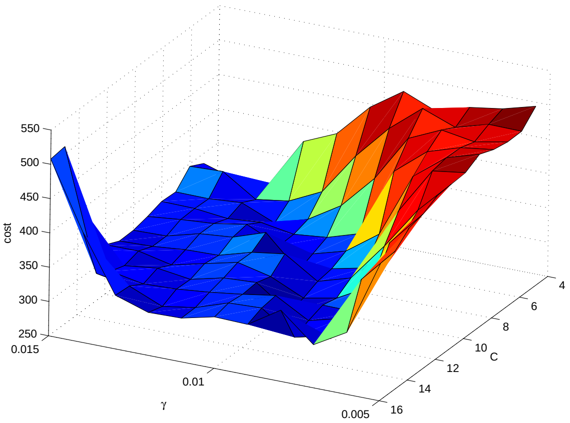

Fig. 7. Cost for the simulation sequence for different $\gamma$ and C values when approximating $u_2(k)$.

while there is no guarantee that this is the case for the SVM controller. This reduces the freedom of the MPC compared to the SVM controller and may result in cases where the SVM controller has lower cost than the MPC.

The sensitivity of the parameter values $\gamma$ and $C$ of the approximator of the pump controller on the overall cost for the simulation sequence was also investigated. This is illustrated in Fig. 7. As can be seen the surface is rather flat around the optimal values, indicating that the control results are not excessively sensitive to changes in these parameters.

The main reason for approximating the control strategy is to reduce the on-line computational burden of the controller. The simulations were performed on a PC with a 2.4 GHz Pentium IV processor and the operations were timed. The simulation time was greatly reduced with the approximator. The total computation time for the MPC was 7500 s while it was only 4.9 s for the SVM. The computational requirements of the MPC strategy vary from one sampling instant to the next and may therefore occasionally be too slow for real-time application.

## 5. CONCLUSIONS

In this work approximation techniques have been used to make an explicit MPC formulation for a hybrid system. Reasonable results compared to the original MPC controller were obtained, while achieving a significant reduction in on-line computations. It was observed that the approximation problem was rather complex and a potential drawback of the SVM approach is that it resulted in large models.The possibility to prune the SVM models remains a topic for further studies.

## REFERENCES

Bemporad, A. (2004). Efficient conversion of mixed logical dynamical systems into an equivalent piecewise affine form. *IEEE transactions on Automatic Control* **49**, 832–838.

Bemporad, A. and M. Morari (1999). Control of systems integrating logic, dynamic and constraints. *Automatica* **35**, 407–427.

Chang, C.-C. and C.-J. Lin (2001). Libsvm: a library for support vector machines. SVM software available on the internet. Available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Hsu, C.-W. and C.-J. Lin (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks* **13**, 415–425.

Åkesson, B. M., H. T. Toivonen, J. B. Waller and R. H. Nyström (2005). Neural network approximation of a nonlinear model predictive controller applied to a pH neutralization process. *Computers & Chemical Engineering* **29**, 323–335.

Kuipers, K. (2000). Bnb20. Shareware on Mathworks website. Available at `http://www.mathworks.com/matlabcentral/fileexchange>optimization`.

Parisini, T. and R. Zoppoli (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica* **31**, 1443–1451.

Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In: *Advances in Kernel Methods – Support Vector Learning* (B. Schölkopf, C. J. C. Burges and A. J. Smola, Eds.). pp. 185–208. MIT Press. Cambridge, MA.

Slupphaug, O., J. Vada and B. Foss (1998). MPC in systems with continuous and discrete control inputs. *Modelling, Identification and Control* **19**, 175–184.

Vapnik, V. N. (1998). *Statistical learning theory*. John Wiley & Sons. New York, NY.