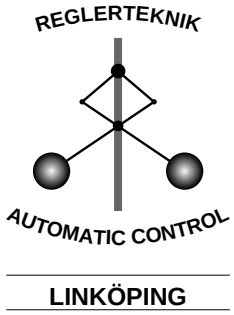
Model-free Predictive Control

Anders Stenman

Department of Electrical Engineering
Linköping University, SE-581 83 Linköping, Sweden
URL: http://www.control.isy.liu.se
Email: stenman@isy.liu.se

August 23, 1999



Report no.: LiTH-ISY-R-2180
To be presented at the 38th IEEE Conference on Decision and Control,
Phoenix, Arizona, USA, December 7–10, 1999.

Model-free Predictive Control

Anders Stenman

Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden.
Phone: +46 13 28 40 79, Fax: +46 13 28 26 22
Email: stenman@isy.liu.se

Abstract

Model predictive control, MPC, is a model-based control philosophy that select control actions by on-line optimization of objective functions. Design methods based on MPC have found wide acceptance in industrial process control applications, and have been thoroughly studied by the academia. Most of the work so far have relied on linear models of different sophistication because of their advantage of providing simple and straightforward implementations. However, when turning to the nonlinear domain, problems often arise as a consequence of the difficulties in obtaining good nonlinear models, and the computational burden associated with the control optimization. In this paper we present a new approach to the nonlinear MPC problem using the recently proposed concept of model-on-demand. The idea is to estimate the process dynamics locally and on-line using process data stored in a database. By treating the local model obtained at each sample time as a local linearization, it is thus possible to reuse tools and concepts from the linear MPC framework. Three different variants of the idea, based on local linearization, linearization along a trajectory and nonlinear optimization respectively, are studied. They are all illustrated in numerical simulations.

1 Introduction

Model predictive control, MPC, is a family of optimal-control related methods that selects control actions by on-line minimization of objective functions. As the name indicates, the methods assume a model of the controlled process and utilize its predictive power to optimize the control. There exist many variants of model predictive control, for instance, *dynamic matrix control* and *extended horizon control*. See García et al. [5] for a survey. If the model is estimated on-line in an adaptive control manner, one usually talks about *generalized predictive control*, GPC, [3].

Early formulations of MPC used linear models of various sophistication such as step response, impulse response or state-space models, and turned out to be successful in controlling linear and mildly nonlinear processes. The performance degradation and instability noticed in the presence of strong nonlinearities, though, soon motivated the extension to nonlinear models. However, both the difficulties in obtaining a good model of the nonlinear process and the excessive computational burden associated with the control optimization have been serious obstacles to widespread industrial implementations. Various simplifications and approximations based on linearization of the nonlinear process have therefore been proposed, see, for instance, Gattu and Zafiriou [6]. In this paper we instead present the new approach of *model-free predictive control*, which incorporates the concept of *model-on-demand* into the GPC framework. This provides an estimator that is better suitable than traditional recursive estimators to cope with nonlinear system dynamics.

The organization of the paper is as follows: Section 2 describes the basics behind modeling-on-demand, Section 3 derives a variant of the GPC setup suitable for the local models. Section 4 illustrates the proposed methods in numerical simulations, and Section 5, finally, provides some concluding remarks.

2 Model on Demand

The model-on-demand concept is a "data mining" technology that has been brought up as an alternative to more traditional modeling approaches like neural nets, radial basis networks and wavelets [11, 9, 10]. Rather than estimating a complex global model covering the entire input-output domain, the idea is instead to model the data belonging to a small neighborhood around the current operating point. This gives the advantage of providing a good fit using a significantly simpler model than the one needed for an acceptable global approximation.

For simplicity and ease of notation we will in this paper restrict ourselves to single-input, single-output processes of NARX type. That is, nonlinear systems governed by the input-output description

$$y(t) = m(\varphi(t)) + e(t), \qquad t = 1, \dots, M, \qquad (1)$$

where $\varphi(t) = \varphi(Z^{t-1})$ denotes a regression vector constructed from lagged input-output data,

$$Z^M \triangleq \{(y(1), u(1)), \ldots, (y(M), u(M))\},$$

$m(\cdot)$ is an unknown nonlinear mapping and $e(t)$ is a noise term modeled as i.i.d. random variables with zero means and variances σ_t^2.

2.1 Basics

The model-on-demand philosophy takes advantage of the increasing ability of computers to collect and manipulate large data sets. The basic idea is to store all observed data from the system in a database as pairs $\{(y(k), \varphi(k))\}_{k=1}^M$, and compute models "on demand". When there is need for a model at (or around) a certain operating point $\varphi(t)$, a subset of relevant data is retrieved from the database and a proper modeling operation is applied to it.

For the modeling part we have adopted a weighted regression approach. That is, at each time instant, a local model is formed via the localized fitting procedure

$$\hat{\beta} = \arg\min_{\beta} \sum_k \ell(y(k) - m(\varphi(k), \beta)) \times W\left(\frac{\|\varphi(k) - \varphi(t)\|_{\mathbf{M}}}{h}\right), \quad (2)$$

where $\|u\|_{\mathbf{M}} \triangleq \sqrt{u^T \mathbf{M} u}$ is a scaled distance function (vector norm) on the regressor space, $\ell(\cdot)$ is a scalar-valued and positive norm function, h is a *bandwidth* parameter controlling the size of the neighborhood around $\varphi(t)$, and $W(\cdot)$ is a window function (usually referred to as the *kernel* [4]) assigning weights to each remote data point according to its distance from $\varphi(t)$. The window is typically a bell-shaped function with support on the interval $[0, 1]$. See Figure 1, where some commonly used windows are depicted.

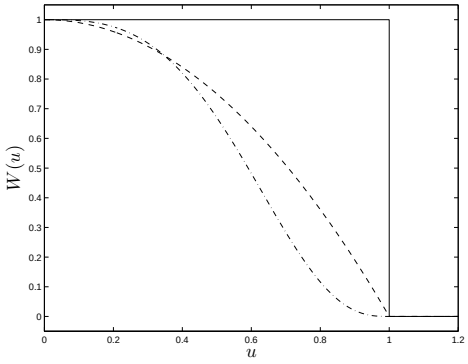


Figure 1: Some commonly used window functions: Uniform (solid), Epanechnikov, $W(u) = (1 - u^2)_+$ (dashed), and tricube, $W(u) = (1 - u^3)_+^3$ (dash-dotted).

Note the distinction between this approach and the traditional adaptive (i.e., recursive) identification approach used in GPC. Adaptive estimators are *recent-data* (i.e., local-in-time) approaches whereas the model-on-demand method represents a *relevant-data* approach. In the formulation (2), relevance is synonymous with closeness in regressor space, but it is of course also possible to incorporate other quality measures like "smallest variance" or "most recent data". Note also that no global model is estimated. All recorded data are kept unchanged in the database, and at each time instant the local estimator returns the best available local model given data. This will increase the flexibility and reduce the bias problem associated with global parametric fitting. The price that has to be paid for these improvements, however, is the need for a computationally expensive data set searching procedure. This of course limits the rate at which predictions can be formed.

In principle, it is possible to use any nonlinear model structure as local model in (2). However, if the quadratic L_2 norm, $\ell(\varepsilon) = \frac{1}{2}\varepsilon^2$, is used and the model is linear in the parameters, the parameter estimate can be explicitly computed using simple and powerful weighted least squares. We thus assume the local linear model structure,

$$m(\varphi(k), \beta) = \beta_0 + \beta_1^T(\varphi(k) - \varphi(t)) \quad (3)$$

as the default choice in the sequel of the paper. If $\hat{\beta}_0$ and $\hat{\beta}_1$ denote the minimizers of (2) using the model (3), it is easily realized that an estimate of $y(t)$ (i.e., a one-step-ahead prediction), is given by

$$\hat{y}(t) = m(\varphi(t), \hat{\beta}) = \hat{\beta}_0. \quad (4)$$

However, it has been shown that it also is possible to enhance the estimate by first estimating higher order derivatives of $m(\cdot)$ from data, and plugging them into an additional optimization step. See Stenman [9] and Stenman et al. [12] for more details around this.

In earlier contributions we have almost exclusively utilized the constant term in the local linear expansion (3) for prediction and simulation. However, it is clear that in a neighborhood of $\varphi(t)$, this model provides an input-output linearization of the form

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \alpha, \quad (5)$$

where $A(q^{-1})$ and $B(q^{-1})$ are polynomials in the backward time-shift operator q^{-1} obtained from the components of $\hat{\beta}_1$, and

$$\alpha = \hat{\beta}_0 - \hat{\beta}_1^T \varphi(t). \quad (6)$$

is an offset term. Some efforts have already been made in the literature to use local models of the form (5) in a control context by using LQ techniques [2]. In Section 3 we will present a new approach based on GPC.

2.2 Model Tuning

It is well known that the bandwidth parameter h has a critical impact on the quality of the prediction (4), since it governs a trade-off between the bias and variance errors. Methods that use the available data to produce good bandwidths are usually referred to as *bandwidth selectors*, and have been thoroughly studied within the statistical literature, see, e.g., Fan and Gijbels [4]. They can roughly be divided into "classical" methods which are based on cross-validation ideas, and "plug-in" methods which rely on minimizing asymptotic MSE expressions [7]. The majority of the bandwidth selectors proposed so far, have been of global type, i.e., they produce a single value that is used over the entire regressor space. However, adaptive (i.e., local) methods, that select bandwidths on-line for each estimation point $\varphi(t)$, have gained a significant interest in recent years, although the development of them still seems to be an open and active research area. The bandwidth selection problem is beyond the scope of this paper, though. Consult Stenman [10] for possibilities.

The scaling matrix $\mathbf{M}$ that controls the distance function (and hence the shape of the neighborhood) can be optimized in a similar way, although we here only have considered fixed choices. Selecting the scaling matrix is very important, though, especially when the regressor components have very different magnitudes. An obvious default choice, which has been adopted here, is to make it proportional to the inverse covariance matrix of the regressors.

3 Model Predictive Control

Model predictive control relies upon the predictive power of the process model and can be formulated as follows [8]: Given a model description of the form (1) and knowledge of the current system state, seek a control that minimizes the performance index (or cost function)

$$J = \sum_{k=0}^{N-1} Q_e(k) \left(r(t+k+1) - \hat{y}(t+k+1)\right)^2 + Q_u(k)u^2(t+k) + Q_{\Delta u}(k) \left(\Delta u(t+k)\right)^2, \quad (7)$$

where $Q_e(k)$, $Q_u(k)$ and $Q_{\Delta u}(k)$ represent penalties on the control error, control signal and control increment magnitudes respectively. Of the N future control actions that minimize J, only the first one is applied to the controlled process. When new measurements become available, a new optimization problem is formulated whose solution provides the next control action. This is usually referred to as the *receding horizon* principle. Another special feature of the formulation (7) is the presence of the control increment,

$$\Delta u(t+k) = u(t+k) - u(t+k-1), \quad (8)$$

in the cost function. In some examples, for instance in process control applications, the change rate of the control action may be restricted. Rather than including the actuator dynamics in the model, it is instead a common practice to include penalties on the control increment. An additional advantage with MPC is that it is straightforward to include hard bounds on the control signal magnitude and the control increment, i.e.,

$$u_{\min} \le u(t+k) \le u_{\max}, \quad (9a)$$

$$\Delta u_{\min} \le \Delta u(t+k) \le \Delta u_{\max}. \quad (9b)$$

Optimization of (7) can be quite demanding for large prediction horizons. To decrease the computational complexity it is thus very common to introduce constraints on the future controls. An often used approach is to assume that the control increments are equal to zero after $N_u \le N$ steps, i.e.,

$$\Delta u(t+k-1) = 0, \qquad k > N_u. \quad (10)$$

It is well known that this also has the effect of producing less aggressive controllers [8]. The quantity N_u is usually referred to as the *control horizon*.

3.1 Optimization Based on Local Linearizations

The most obvious way of incorporating the model-on-demand approach into the MPC formulation is to optimize the objective (7) based on the local model delivered by the estimator at time t. A similar idea was explored by Gattu and Zafiriou [6] using state-space models, but here we choose to remain in the input-output domain.

From the model (3) we obtain an input-output linearization of the form (5). The basic problem is now to express the output prediction at time $t+k$ as a function of future controls. This can be done similar to the usual GPC derivation which is included here for the sake of completeness. The key trick [1] is to introduce the identity

$$1 = A(q^{-1})F_k(q^{-1}) + q^{-k}G_k(q^{-1}) \quad (11)$$

where $F_k(q^{-1})$ and $G_k(q^{-1})$ denote polynomials of degrees $k-1$ and $n_a - 1$ respectively. Substituting (11) into (5) yields

$$\hat{y}(t+k) = B(q^{-1})F_k(q^{-1})u(t+k-1) + G_k(q^{-1})y(t) + F_k(1)\alpha. \quad (12)$$

By partitioning $B(q^{-1})F_k(q^{-1})$ as

$$B(q^{-1})F_k(q^{-1}) = S_k(q^{-1}) + q^{-k}\tilde{S}_k(q^{-1}), \quad (13)$$

where $\deg S_k(q^{-1}) = k-1$ and $\deg \tilde{S}_k(q^{-1}) = n_b - 2$, the output prediction (12) can be rewritten as

$$\hat{y}(t+k) = S_k(q^{-1})u(t+k-1) + \bar{y}(t+k). \quad (14)$$

Here the first term depends on future control actions $u(t),\dots,u(t+k-1)$ whereas the remaining terms (collected into $\bar{y}(t+k)$) depend on known quantities only. By introducing the notations

$$\begin{aligned}
\hat{\mathbf{y}} &\triangleq \begin{pmatrix}\hat{y}(t+1) & \dots & \hat{y}(t+N)\end{pmatrix}^T,\\
\tilde{\mathbf{u}} &\triangleq \begin{pmatrix}u(t) & \dots & u(t+N-1)\end{pmatrix}^T,\\
\bar{\mathbf{y}} &\triangleq \begin{pmatrix}\bar{y}(t+1) & \dots & \bar{y}(t+k)\end{pmatrix}^T,\\
\tilde{\mathbf{S}} &\triangleq \begin{pmatrix} s_0 & 0 & \dots & 0\\ s_1 & s_0 & \dots & 0\\ \vdots & & \ddots & \vdots\\ s_{N-1} & s_{N-2} & \dots & s_0\end{pmatrix},
\end{aligned}$$

where s_i are the coefficients of $S_k(q^{-1})$ we have that

$$\hat{\mathbf{y}} = \bar{\mathbf{y}} + \tilde{\mathbf{S}}\tilde{\mathbf{u}}. \tag{15}$$

However, taking into account that the control horizon N_u typically is shorter than the prediction horizon N, i.e., that (10) holds, this can be rewritten as

$$\hat{\mathbf{y}} = \bar{\mathbf{y}} + \mathbf{S}\mathbf{u}, \tag{16}$$

where

$$\mathbf{u} \triangleq \begin{pmatrix}u(t) & \dots & u(t+N_u-1)\end{pmatrix}^T, \tag{17}$$

and

$$\mathbf{S} \triangleq \tilde{\mathbf{S}}\mathbf{\Lambda} \quad \text{with} \quad \mathbf{\Lambda} \triangleq \begin{pmatrix}1 & 0 & \dots & 0 & \dots & 0\\ 0 & 1 & \dots & 0 & \dots & 0\\ \vdots & \vdots & \ddots & \vdots & & \vdots\\ 0 & 0 & \dots & 1 & \dots & 1\end{pmatrix}^T.$$

The control increments (8) can also be expressed in vector form

$$\mathbf{\Delta u} = \mathbf{D}\mathbf{u} - \bar{\mathbf{u}} \tag{18}$$

by introducing the auxiliary quantities

$$\mathbf{D} \triangleq \begin{pmatrix}1 & 0 & \dots & 0\\ -1 & 1 & & \vdots\\ 0 & \ddots & \ddots & 0\\ 0 & \dots & -1 & 1\end{pmatrix} \quad \text{and} \quad \bar{\mathbf{u}} \triangleq \begin{pmatrix}u(t-1)\\ 0\\ \vdots\\ 0\end{pmatrix}.$$

The objective (7) can thus be simplified as

$$\begin{aligned}
J(\mathbf{u}) &= \|\mathbf{r}-\hat{\mathbf{y}}\|^2_{\mathbf{Q_e}} + \|\mathbf{u}\|^2_{\mathbf{Q_u}} + \|\mathbf{\Delta u}\|^2_{\mathbf{Q_{\Delta u}}}\\
&= \|\mathbf{r}-\bar{\mathbf{y}}-\mathbf{S}\mathbf{u}\|^2_{\mathbf{Q_e}} + \|\mathbf{u}\|^2_{\mathbf{Q_u}} + \|\mathbf{D}\mathbf{u}-\bar{\mathbf{u}}\|^2_{\mathbf{Q_{\Delta u}}},
\end{aligned} \tag{19}$$

where

$$\mathbf{r} \triangleq \begin{pmatrix}r(t+1) & \dots & r(t+N)\end{pmatrix} \tag{20}$$

denotes the desired (and possibly smoothed) reference trajectory, and $\mathbf{Q_e}$, $\mathbf{Q_u}$ and $\mathbf{Q_{\Delta u}}$ are diagonal matrices with entries $Q_e(k)$, $Q_u(k)$ and $Q_{\Delta u}(k)$ respectively.

For the unconstrained case, the minimizing control sequence $\mathbf{u}^*$ is thus obtained explicitly by means of ordinary weighted least squares.

For the constrained case, the constraints (9) can be compactly re-formulated as

$$\mathbf{C}\mathbf{u} \le \mathbf{c} \tag{21}$$

where

$$\mathbf{C} \triangleq \begin{pmatrix}I\\ -I\\ \mathbf{D}\\ -\mathbf{D}\end{pmatrix} \quad \text{and} \quad \mathbf{c} \triangleq \begin{pmatrix}u_{\max}\cdot\mathbf{1}\\ -u_{\min}\cdot\mathbf{1}\\ \Delta u_{\max}\cdot\mathbf{1}+\bar{\mathbf{u}}\\ -\Delta u_{\min}\cdot\mathbf{1}-\bar{\mathbf{u}}\end{pmatrix}.$$

We thus obtain a quadratic programming problem which can be efficiently solved using standard numerical optimization software.

3.2 Linearization along Trajectories

A drawback with the approach described in the preceeding subsection, is that the predicted future behavior of the process is determined on basis of the input-output linearization delivered by the estimator at time t. In this way we have implicitly assumed that the linearization is a valid model along the trajectory we predict. An obvious solution to this problem is to feed the previously optimized control sequence $\mathbf{u}^*$ to the local estimator and compute the corresponding N-step-ahead prediction. Then we will obtain an approximate time-varying local linear model over the future N samples, and the expressions derived in the end of Section 3.1 can be utilized when optimizing the objective function.

At each time instant $\tau = t+1,\dots,t+N$ along the hypothesized future trajectory, the local estimator will return a input-output linearization of the form

$$A_\tau(q^{-1})y(\tau) = B_\tau(q^{-1})u(\tau) + \alpha_\tau$$

An output predictor associated with this time-varying model can be derived similar to the time-invariant case in Section 3.1. Introduce the identity

$$\begin{aligned}
1 = {} & A_{t+k}(q^{-1}) + f_1 q^{-1} A_{t+k-1}(q^{-1}) + \dots\\
& + f_{k-1}q^{-k+1}A_{t+1}(q^{-1}) + q^{-k}G_{t,k}(q^{-1})
\end{aligned} \tag{22}$$

which can be interpreted as the time-variant counterpart of (11). The coefficients of $F_{t,k}$ and $G_{t,k}$ can be determined using repeated polynomial division. For instance, for f_1 we have

$$\frac{q\left(1-A_{t+k}(q^{-1})\right)}{A_{t+k-1}(q^{-1})} = f_1 + \frac{R_1(q^{-1})}{A_{t+k-1}(q^{-1})}. \tag{23}$$

By applying the same procedure on the remainder terms $R_i(q^{-1})$, the rest of the coefficients of $F_{t,k}(q^{-1})$

and $G_{t,k}(q^{-1})$ can be determined. The corresponding output predictor is given by

$$\hat{y}(t+k) = \big(B_{t+k}(q^{-1}) + f_1 q^{-1} B_{t+k-1}(q^{-1}) + \dots + f_{k-1} q^{-k+1} B_{t+1}(q^{-1})\big) u(t+k-1) + G_{t,k}(q^{-1}) y(t) + F_{t,k}(q^{-1}) \alpha_{t+k}. \quad (24)$$

As in (14) this expression can be partitioned in one part that depends on future control moves and one part that depends on measured data only;

$$\hat{y}(t+k) \;=\; S_{t,k}(q^{-1}) u(t+k-1) + \bar{y}(t+k). \quad (25)$$

Hence the output predictions can be expressed similar to (16), and the control sequence can be optimized analogously to the derivation in Section 3.1.

An obvious question is of course which input sequence that should be used when determining the time-varying model. A straightforward approach here, though, is to take advantage of the result obtained from the optimization performed at the previous sampling instant. Another alternative is to use the result from Section 3.1 as an initial value.

3.3 Numerical Optimization

The most general solution to the nonlinear MPC problem is perhaps direct optimization of the criterion (7) using nonlinear programming methods. That is, the optimal control sequence is determined using a constrained numerical optimization routine, that at each iteration simulates the system given the current value of $\mathbf{u}$ (that is, computes an N-step-ahead prediction), and then updates it in a direction such that the value of the performance index J decreases.

It is known that more accurate results could be obtained if the optimization routine is provided with gradient information. From (19) we have that

$$\nabla J = 2\mathbf{S}^T \mathbf{Q_e}(\hat{\mathbf{y}} - \mathbf{r}) + 2\mathbf{Q_u}\mathbf{u} + 2\mathbf{D}^T \mathbf{Q_{\Delta u}} \mathbf{\Delta u}. \quad (26)$$

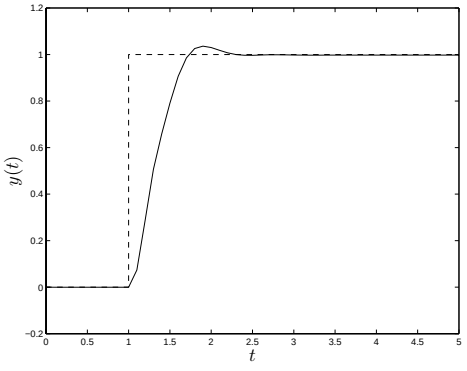
The gradient of J with respect to $\mathbf{u}$ can thus be estimated from the local models obtained along the simulated trajectory.

A major problem with the general optimization formulation, though, is that the optimization typically will be very complex and time-consuming for large control horizons N_u. We will most likely also get the problem with local minima.

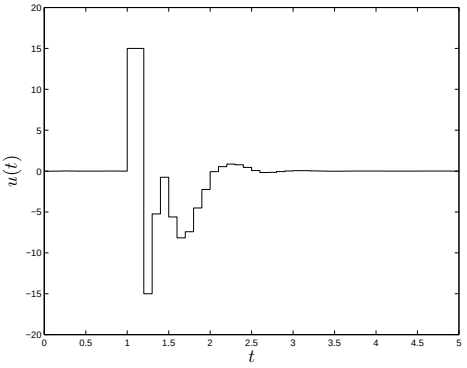
4 A Simple Example

To illustrate the proposed methods in simulations we will consider the nonlinear system

$$\ddot{y}(t)\,(1 + |y(t)|) = u(t). \quad (27)$$



(a)



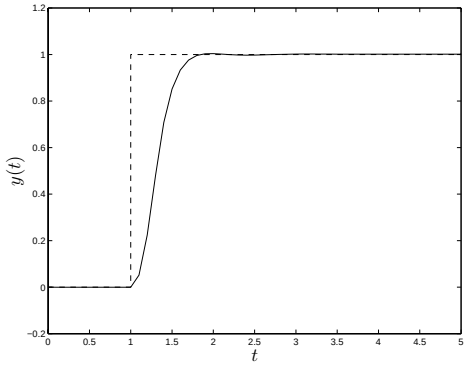
(b)

Figure 2: Step response experiment using the predictive controller of Section 3.1. (a) Reference signal (dashed) and system output (solid). (b) Control signal.

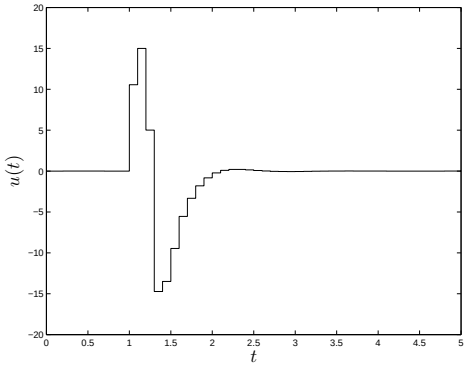
It has the property that the gain of system decreases as the magnitude of the output signal increases. The open-loop system is unstable, so a database consisting of 2000 output and regressor pairs $(y(k), \varphi(k))$ was built-up during a closed-loop experiment using a proportional controller, $u(t) = K(y(t) - r(t))$, and a white noise reference signal $r(t)$. The sampling interval was selected as $T_s = 0.1$ seconds. A simulation using this database and the predictive controller of Section 3.1 with parameter values $N = 10$, $N_u = 7$, $Q_e = 1$, $Q_u = 0$ and $Q_{\Delta u} = 0.001$, and assuming that the control magnitude is limited according to

$$|u(t+k)| \le 15, \qquad k = 0, \dots, N_u - 1$$

is shown in Figure 2. A simulation using the same parameter values but the controller of Section 3.2 is shown in Figure 3. We see that the second controller gives a much better result. Since this controller uses linear models of the plant along the predicted future trajectory, it is aware of that the gain of the system will decrease as the output magnitude increases. Therefore it is more restrictive in its use of control energy as the output approaches the setpoint.



(a)



(b)

Figure 3: Step response experiment using the predictive controller of Section 3.2. (a) Reference signal (dashed) and system output (solid). (b) Control signal.

5 Conclusions

This paper has presented the promising concept of *model-free predictive control*, which combines the idea of model-on-demand with established and well-known MPC techniques. The proposed method is model-free in the sense that no global model of the process dynamics is needed. Instead it uses a local estimator that depends upon process data stored in a database.

Under assumption that the estimator delivers local linear models, an advantage with the approach is that it is possible to use standard GPC techniques for the controller part. This also implies that standard tuning guidelines apply. Moreover, since the estimator at each sampling instant returns the best available linearization given data, the method has the ability of controlling some classes of nonlinear systems. This has been demonstrated in several numerical simulations [10]. However, the concepts based on linearizations (sections 3.1 and 3.2) will probably not work for controlling strongly nonlinear systems where the local models are valid only in a very narrow neighborhood of the operating point. In such situations one is probably better off with the numerical approach of Section 3.3. A drawback with the approach is that the performance of the controller is critically depending on the quality of the database contents. It is therefore very important to validate the database through simulations before using it for control. Another potential problem is that the controller normally requires large computational resources due to the nature of the underlying estimation procedure.

References

[1] K.J. Åström and B. Wittenmark. *Adaptive Control*. Addison-Wesley, second edition, 1995.

[2] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11(1-5):11–73, 1997.

[3] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized predictive control. *Automatica*, 23(2):137–160, 1987.

[4] J. Fan and I. Gijbels. *Local Polynomial Modelling and Its Applications*. Chapman & Hall, 1996.

[5] C.G. García, D.M. Prett, and M. Morari. Model predictive control: Theory and practice – a survey. *Automatica*, 25(3):335–348, 1989.

[6] G. Gattu and E. Zafiriou. Nonlinear quadratic dynamic matrix control with state estimation. *Ind. Eng. Chem. Res.*, 31(4):1096–1104, 1992.

[7] C.R. Loader. Old faithful erupts: Bandwidth selection reviewed. Technical report, AT&T Bell Laboratories, 1995.

[8] E.S. Meadows and J.B. Rawlings. Model predictive control. In M.A. Henson and D.E. Seborg, editors, *Nonlinear Process Control*, chapter 5. Prentice Hall, 1997.

[9] A. Stenman. *Just-in-Time Models with Applications to Dynamical Systems*. Licentiate thesis LIU-TEK-LIC-1997:02, Dept of EE, Linköping University, SE-581 83 Linköping, Sweden, March 1997.

[10] A. Stenman. *Model on Demand: Algorithms, Analysis and Applications*. PhD thesis, Dept of EE, Linköping University, SE-581 83, Linköping, Sweden, May 1999.

[11] A. Stenman, F. Gustafsson, and L. Ljung. Just in time models for dynamical systems. In *Proceedings of the 35th IEEE Conference on Decision and Control, Kobe, Japan*, pages 1115–1120, 1996.

[12] A. Stenman, A.V. Nazin, and F. Gustafsson. Asymptotic properties of Just-in-Time models. In Y. Sawaragi and S. Sagara, editors, *Preprints of the 11th IFAC Symposium on System Identification, Kitakyushu, Japan*, pages 1249–1254, 1997.