Predictive Control – An Introduction

M. Fikar
Department of Process Control
Faculty of Chemical Technology
Slovak University of Technology in Bratislava
Radlinského 9, 812 37 Bratislava
Slovakia

Technical Report KAMF9801

url: www.esr.ruhr-uni-bochum.de/~fikar/research/gpc/98r_pred.zip

Version 1.4: 27.08.1999

Abstract

This report considers an introduction to predictive control methology and methods. Basic principles of Model Based Predictive Control (MBPC) are discussed. Some of the best known approaches are given. Generalised Predictive Control (GPC) is derived in unconstrained and constrained cases. Some stability results are presented. Last part of the report deals with tuning of the algorithm and with simulation and real-time examples.

Keywords: Model Based Predictive Control, Review, Constraints

Contents

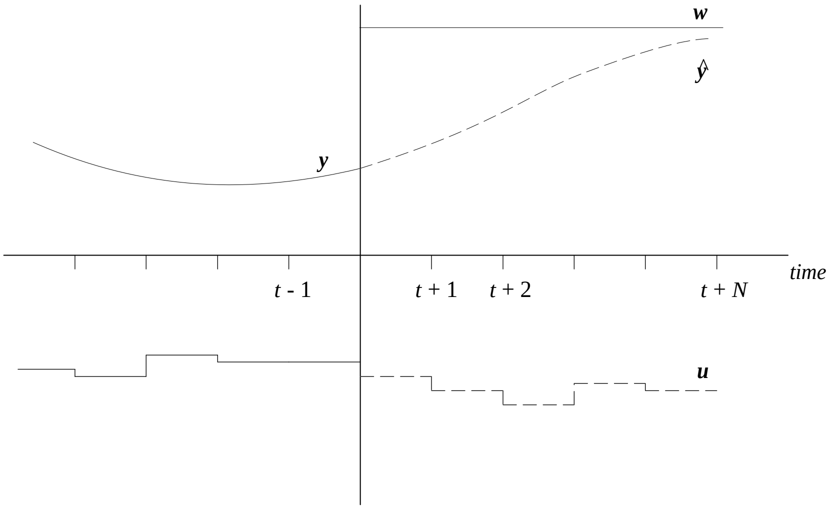

Figure 1: Principle of MBPC

1 Introduction

Model Based Predictive Control (MBPC) or only Predictive Control is a broad variety of control methods that comprise certain common ideas:

- a process model that is explicitly used to predicts the process output for a fixed number of steps into future,
- a known future reference trajectory,
- calculation of a future control sequence minimising a certain objective function (usually quadratic, that involves future process output errors and control increments),
- receding strategy: at each sampling period only the first control signal of the sequence calculated is applied to a process controlled.

Among many useful features of MBPC, there is one that has created extensive industrial interest: the process constraints can easily be incorporated into the method at the design stage.

MBPC algorithms are reported to be very versatile and robust in process control applications. They usually outperform PID controllers and are applicable to non-minimum phase, open-loop unstable, time delay, and multivariable processes.

The principle of MBPC is shown in Fig. 1 and is as follows:

1. The process model is used to predict the future outputs $\hat{y}$ over some horizon N. The predictions are calculated based on information up to time t and on the future control actions that are to be determined.
2. The future control trajectory is calculated as a solution of an optimisation problem consisting of a cost function and possibly some constraints. The cost function comprises future output predictions, future reference trajectory, and future control actions.
3. Although the whole future control trajectory was calculated in the previous step, only its first element $u(t)$ is actually applied to the process. At the next sampling time the procedure is repeated. This is known as *Receding Horizon* concept.

1.1 Some Important References

Begin: Richalet (MAC) [40], Cutler and Ramaker (DMC) [10].

GPC: [6, 7, 47].

Survey papers: [19].

Useful books: [1, 4, 9, 38, 51].

Stable predictive control: CRHPC [8], SIORHC [35], SGPC [20, 24, 25, 43, 44, 45], YKPC [15, 16, 17].

Kwon et. al.: [27, 29].

Morari et. al.: [5, 28, 55, 56].

Rawlings et. al.: [32, 36, 37, 39, 48].

Zafiriou et. al.: [52, 53, 54].

Tuning: [31, 49, 50].

Others [12, 18, 22, 23, 30, 34].

1.2 Abbreviations

AGPC	Adaptive Generalised Predictive Control
ANN	Artificial Neural Network
ARE	Algebraic Riccati Equation
ARIMAX	AutoRegressive Integrated Moving Average eXogenous
ARMA	AutoRegressive Moving Average
ARX	AutoRegressive eXogenous
CARIMA	Controlled AutoRegressive Integrated Moving Average
CRHPC	Constrained Receding Horizon Predictive Control
DMC	Dynamic Matrix Control
EHAC	Extended Horizon Adaptive Control
EPSAC	Extended Prediction Self-Adaptive Control
FIR	Finite Impulse Response
GMV	Generalised Minimum Variance
GPC	Generalised Predictive Control
GPCNN	Generalised Predictive Control + Neural Networks
LQ	Linear Quadratic
MAC	Model Algorithmic Control
MBPC	Model Based Predictive Control
MIMO	Multi Input Multi Output
MV	Minimum Variance
QP	Quadratic Programming
RLS	Recursive Least Squares
SIORHC	Stabilising Input/Output Receding Horizon Control
SISO	Single Input Single Output
SGPC	Stable Generalised Predictive Control
YKPC	Youla-Kučera Predictive Control

2 Ingredients of MBPC

2.1 Models

MBPC enables to plug-in directly any type of the process model. Of course, linear models are most often used. This is caused by the possibility of an analytic solution for the future control trajectory in unconstrained case.

The model should capture the process dynamics and to permit theoretical analysis. The process model is required to calculate predicted future output trajectory. Some of the models incorporate directly disturbance model, in others it is simply assumed that disturbance is constant.

2.1.1 Impulse Response

The theoretical impulse sequence is usually truncated for practical reasons. The output is related to the input by the equation

$$y(t) = \sum_{i=1}^{N} h_i u(t-i) = H(q^{-1})u(t) \tag{1}$$

where $H(q^{-1}) = h_1 q^{-1} + h_2 q^{-2} + \cdots h_N q^{-N}$ and q^{-1} is the backward shift operator defined as $y(t)q^{-1} = y(t-1)$.

The drawbacks of this model are:

- high value of N needed ≈ 50,
- only stable processes can be represented.

2.1.2 Step Response

The step response model is very similar to the FIR model with the same drawbacks. Again, truncated step response is used for stable systems

$$y(t) = \sum_{i=1}^{N} g_i \Delta u(t-i) = G(q^{-1})(1-q^{-1})u(t) \tag{2}$$

As the step and impulse responses are easily collected, the methods based on them gained large popularity in the industry. The step model is for example used in DMC.

2.1.3 Transfer Function

This model is used in GPC, EHAC, EPSAC, and others. The output is modelled by the equation

$$A(q^{-1})y(t) = B(q^{-1})u(t) \tag{3}$$

The advantage of this representation is that is also valid for unstable models. On the other side, order of the A, B polynomials is needed.

2.1.4 State Space

The representation of the state-space model is as follows:

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \qquad (4) \\ y(t) &= Cx(t) \qquad (5) \end{aligned}$$

It advantage is uncomplicated way of dealing with multivariable processes. However, the state observer is often needed [3].

2.1.5 Others

As it was stated before, any other process model is acceptable. Continuous nonlinear model in the form of ordinary differential equations are often used. Their drawbacks are large simulation times. The area of dynamic optimisation usually covers them.

Recently, neural and fuzzy models have gained its popularity. Two approaches have emerged. The model is either directly used, or it only generates some process characteristics: step or impulse responses.

2.1.6 Disturbances

The most general disturbance model is an ARMA process given by

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})}\xi(t) \qquad (6)$$

with $\xi(t)$ being white noise. Within the MBPC framework, the $D = \Delta A$ polynomial includes the integrator $\Delta = 1 - q^{-1}$ to cover random-walk disturbances. Another pleasing property of the integrator is set-point tracking (integral action).

The overall model is then called CARIMA (or ARIMAX) and is given as

$$\Delta A y(t) = B\Delta u(t) + C\xi(t) \qquad (7)$$

2.2 Objective Function

The different MBPC algorithm propose various cost functions that lead to optimal future control trajectory. All approaches deal with a finite horizon quadratic objective function that usually contains penalisation of (possibly filtered) future control errors and future control increments. Thus, a general form that can include all variants may be given as [51]

$$J = \sum_{i=N_1}^{N_2} [P\hat{y}(k+i) - Rw(k+i)]^2 + \rho \sum_{i=1}^{N_u} \left[\frac{Q_n}{Q_d} u(k+i-1)\right]^2 \qquad (8)$$

where $\hat{y}(k+i)$ is the predicted output i steps into the future and based upon information available at time t, $w(k+i)$ is the reference signal, and $u(k+i-1)$ is the sequence of future control actions that is to be determined.

However, this cost function might often be regarded as too complicated. The standard cost function can be written as

$$J = \sum_{i=N_1}^{N_2} [P\hat{y}(k+i) - w(k+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(k+i-1)]^2 \qquad (9)$$

Here, the sequence of control increments $\Delta u(k+i) = u(k+i) = u(k+i-1)$ is to be determined rather than the sequence of $u(k+i)$. In both cases, implicit constraints on Δu are placed between N_u and N_2 as

$$\Delta u(k+i-1) = 0, \quad N_u < i \leq N_2 \tag{10}$$

The cost function parameters are following:

- Horizons N_1, N_2, and N_u called minimum, maximum, and control horizon, respectively. The horizons N_1 and N_2 mark the future time interval where it is desirable to follow the reference trajectory. N_1 should be at least equal to $d+1$ where d is the assumed value of process time delay. Also, the non-minimum phase behaviour of the process can be eliminated from the cost by letting N_1 to be sufficiently large. The value of N_2 should cover the important part of the step response curve, usually is chosen to be about the settling time of the plant. The use of the control horizon N_u reduces computational load of the methods.

- Reference trajectory $w(k+i)$ is assumed to be known beforehand. Several approaches are possible. The most simple is to assume that the future reference is constant and equal to the desired setpoint w^∞. The preferred approach is to use smooth reference trajectory that begins from the actual output value and approaches asymptotically via the first order filter the desired setpoint w^∞. It is thus given as

$$\begin{aligned} w(k) &= y(k) & (11)\\ w(k+i) &= \alpha w(k+i-1) + (1-\alpha)w^\infty & (12) \end{aligned}$$

  The parameter α determines smoothness of the trajectory with $\alpha \to 0$ being the fastest and $\alpha \to 1$ being the slowest trajectory.

  The same effect can be achieved with the use of the filter polynomial $P(q^{-1})$. The output y follows the model trajectory $\frac{1}{P}w$. The corresponding filter to the previous first order trajectory is given as

$$P(q^{-1}) = \frac{1-\alpha q^{-1}}{1-\alpha} \tag{13}$$

3 Overview of Existing Algorithms

In this section some MBPC algorithms will be briefly discussed, highlighting their distinguishing features and their comparative advantages and disadvantages.

3.1 The Model Algorithmic Controller

The MAC algorithm utilises FIR model (impulse response). Its cost function is given by

$$J = \sum_{i=1}^{N_2} \left[e^2 + \lambda \Delta u(k+i-1)^2 \right] \tag{14}$$

with e defined as

$$e = w(k+i) - \hat{y}(k+i) - H\Delta u(k+i-1) \tag{15}$$

where the matrix H contains the plant impulse coefficients

$$H = \begin{pmatrix} h_1 & 0 & \dots & \dots & 0 \\ h_2 & h_1 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & 0 \\ h_{N_2} & \dots & & \dots & h_1 \end{pmatrix} \tag{16}$$

Thus, MAC approach fixes the values of the horizons $N_1 = 1$ and $N_u = N_2$.

3.2 The Dynamic Matrix Controller

The DMC algorithm uses an truncated step response model. The cost function is given as

$$J = \sum_{i=N_1}^{N_2} [\hat{y}(k+i) - w(k+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(k+i-1)]^2 \tag{17}$$

Advantages:

- No assumption about the order of the process is required,
- simple implementation,
- attractive for use in industry by personnel without extensive training.

Disadvantage:

- Open-loop unstable processes cannot be modeled or controlled.

The DMC algorithm has been developed into a very successful commercial package with many applications, mainly in the petrochemical industry.

3.3 The Extended Self-Adaptive Controller

This method uses a CARIMA model for prediction purposes. The optimal control law is found by minimizing the cost function that does not contain control penalisation term:

$$J = \sum_{i=1}^{N_2} \rho_i \, [\hat{y}(k+i) - Pw(k+i)]^2 \tag{18}$$

where $P(q^{-1})$ is a design polynomial and ρ_i an exponential weighting factor. Three factors are worth noting:

- The prefilter P can be used as a predesign parameter to affect the disturbance rejection properties.
- Open-loop unstable plants can be controlled by appropriately selecting P, N_2, ρ_i, however, the tuning is more involved.
- The absence of the control signal from the cost function implies that undesired large control signal variations cannot be suppressed.

3.4 The Extended-Horizon Adaptive Controller

The EHAC algorithm assumes an ARX model of the form

$$A(q^{-1})y(t) = B(q^{-1})u(t-d) \tag{19}$$

where d is time delay. The cost function is given as

$$J = [\hat{y}(k+N_2) - w(t-N_2)]^2 \tag{20}$$

In this approach, a sequence of inputs $[u(t), u(t+1), \ldots, u(t+N_2-d)]$, is computed. Of course, the cost must attain its minimum at zero and the control trajectory computed is not unique unless $N_2 = d$, resulting in a number of different ways to finding the control sequence. Possible approaches include assumption that the control is constant over the whole interval or that control effort is minimised.

EHAC Disadvantages

- Because only one tuning parameter is involved, a compromise between closed-loop performance and stability must be made.
- Finding the optimal control law is more involved when compared to the other MBPC methods.

3.5 The Generalized Predictive Controller

The GPC controller uses prediction based upon a CARIMA model. The cost function is given as

$$J = \sum_{i=N_1}^{N_2} [P(q^{-1})\hat{y}(k+i) - w(k+i)]^2 + \lambda \sum_{i=1}^{N_u} [\Delta u(k+i-1)]^2 \tag{21}$$

GPC Advantages

- The GPC is normally able to stabilize and control non-minimum phase, dead time, and open-loop unstable processes through judicious choice of the tuning parameters $N_1, N_2, \lambda, N_u, P$.
- The properties of GPC and LQ approaches are closely related.
- Stability proofs exist for a number of cases.
- A number of well-known controllers may be created by GPC (mean level, deadbeat control).

Disadvantages

- Some choice of the parameters and controlled systems may destabilise the closed-loop system.

4 Derivation and Implementation of GPC

The GPC method is in the principle applicable to both SISO and MIMO processes. We begin the derivation for SISO systems for simplicity and show in the actual implementation of the method, how to treat MIMO systems.

4.1 Derivation of the Predictor

The first step in the development of MBPC is derivation of the optimal predictor. We start with the CARIMA model (7) of the form

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + \frac{C(q^{-1})}{\Delta}\xi(t) \tag{22}$$

Note that we use explicitly $u(t-1)$ and thus the polynomial B has non-zero absolute coefficient. We use $u(t-1)$ because $u(t)$ will constitute one element of the optimised variables.

Now let us think about this equation j steps in the future. This is accomplished by multiplication of this equation by q^j and yields

$$y(t+j) = \frac{B}{A}u(t+j-1) + \frac{C}{\Delta A}\xi(t+j) \tag{23}$$

The last term of this equation contains past and future values of ξ. We may separate them by performing long division on the term $C/(\Delta A)$ and by separating the first j terms (quotient) with positive powers of q. This yields

$$\frac{C(q^{-1})}{\Delta A(q^{-1})} = E_j(q^{-1}) + q^{-j}\frac{F_j(q^{-1})}{\Delta A(q^{-1})} \tag{24}$$

where the polynomial E_j has degree $j-1$. Inserting back into (23) gives

$$y(t+j) = \frac{B}{A}u(t+j-1) + E_j\xi(t+j) + \frac{F_j}{\Delta A}\xi(t) \tag{25}$$

The last term contains actual value of the disturbance $\xi(t)$. This can be calculated from (22) and inserted back into the last equation and gives

$$\begin{aligned}
y(t+j) &= \frac{B}{A}u(t+j-1) - \frac{F_jB}{\Delta AC}\Delta u(t-1) + \frac{F_j}{C}y(t) + E_j\xi(t+j) \\
&= \left[\frac{B}{\Delta A} - q^{-j}\frac{F_jB}{\Delta AC}\right]\Delta u(t+j-1) + \frac{F_j}{C}y(t) + E_j\xi(t+j) \\
&= \frac{B}{C}\left[\frac{C}{\Delta A} - q^{-j}\frac{F_j}{\Delta A}\right]\Delta u(t+j-1) + \frac{F_j}{C}y(t) + E_j\xi(t+j)
\end{aligned} \tag{26}$$

and finally substituting (24) into the term containing $\Delta u(t+j-1)$ yields

$$y(t+j) = \frac{BE_j}{C}\Delta u(t+j-1) + \frac{F_j}{C}y(t) + E_j\xi(t+j) \tag{27}$$

Again, we separate unknown (future and present) control actions from the known (past) ones by the means of the polynomial division

$$\frac{B(q^{-1})E_j(q^{-1})}{C(q^{-1})} = G_j(q^{-1}) + q^{-j}\frac{\Gamma_j(q^{-1})}{C(q^{-1})} \tag{28}$$

This gives the final form for the future value of the system output

$$y(t+j) = G_j\Delta u(t+j-1) + \frac{\Gamma_j}{C}\Delta u(t-1) + \frac{F_j}{C}y(t) + E_j\xi(t+j) \tag{29}$$

It is obvious that the minimum variance prediction of $y(t+j)$ for given data up to time t is obtained by replacing the last term containing future disturbances by zero and yields

$$\hat{y}(t+j) = G_j\Delta u(t+j-1) + \frac{\Gamma_j}{C}\Delta u(t-1) + \frac{F_j}{C}y(t) \tag{30}$$

$$\hat{y}(t+j) = G_j\Delta u(t+j-1) + y_0(t+j) \tag{31}$$

Thus, to obtain the j step predictor, two polynomial divisions (or equivalently Diophantine equations) are to be solved

$$\begin{aligned} C &= E_j \Delta A + q^{-j} F_j & (32)\\ BE_j &= G_j C + q^{-j}\Gamma_j & (33)\end{aligned}$$

To implement calculation of the predictor efficiently, it is necessary to understand correctly the rôle of the equation (31) and the terms involved in it.

Let us at first assume that all future control increments are zero. Equation (31) gives

$$\hat{y}(t+j) = y_0(t+j) \tag{34}$$

Hence, the term y_0 can be determined by the *free response* of the system if the input remains to be constant at the last computed value $u(t-1)$.

Similarly, let us assume that the system is at the time t in the steady-state and we may without loss of generality assume that the steady-state is zero. This gives zero free response $y_0(t+j)$. If at time t the system is subject to unit step in input the system output is given from (31) as

$$\begin{aligned} \hat{y}(t+j) &= G_j(q^{-1})\Delta u(t+j-1)\\ &= g_{j0}\Delta u(t+j-1) + g_{j1}\Delta u(t+j-2) + \cdots + g_{j,j-1}\Delta u(t)\\ &= g_{j,j-1} & (35)\end{aligned}$$

Thus, the polynomial $G_j(q^{-1})$ contains the system step response coefficients. As an alternative way to show this consider (24) multiplied by B/C :

$$\begin{aligned} \frac{B}{\Delta A} &= \frac{BE_j}{C} + q^{-j}\frac{BF_j}{\Delta AC}\\ &= G_j + +q^{-j}\frac{\Gamma_j}{C} + q^{-j}\frac{BF_j}{\Delta AC} & (36)\end{aligned}$$

which shown that G_j is the quotient from the division $B/(\Delta A)$.

This also shows equivalence between DMC and GPC methods, as both use the same information about the process.

4.2 Calculation of the Optimal Control

The GPC cost function is given by (21). Let us now assume for simplicity that $N_1 = 1, N_u = N_2, P = 1$. It follows that all output prediction up to time $t + N_2$ are needed. Let us stack individual output predictions, future control increments, future reference trajectory, and free responses into corresponding vectors

$$\begin{aligned} \hat{\boldsymbol{y}}^T &= [\hat{y}(t+1), \hat{y}(t+2), \ldots, \hat{y}(t+N_2)] & (37)\\ \boldsymbol{y}_0^T &= [y_0(t+1), y_0(t+2), \ldots, y_0(t+N_2)] & (38)\\ \tilde{\boldsymbol{u}}^T &= [\Delta u(t), \Delta u(t+1), , \ldots, \Delta u(t+N_2-1)] & (39)\\ \boldsymbol{w}^T &= [w(t+1), w(t+2), \ldots, w(t+N_2)] & (40)\\ & & (41)\end{aligned}$$

To vectorise the predictor (31), let us form a matrix containing step response coefficients given as

$$\boldsymbol{G} = \begin{pmatrix} g_0 & 0 & \dots & \dots & 0 \\ g_1 & g_0 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & g_0 & 0 \\ g_{N_2-1} & \dots & & \dots & g_0 \end{pmatrix} \tag{42}$$

If we take into effect real value of N_1, then the first $N_1 - 1$ rows of the matrix $\boldsymbol{G}$ should be deleted. Similarly, only the first N_u columns are retained. Thus, the real matrix $\boldsymbol{G}$ has the dimension $[N_2 - N_1 + 1 \times N_u]$.

Hence, the predictor in the vector notation can be written as

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\tilde{\boldsymbol{u}} + \boldsymbol{y}_0 \tag{43}$$

and the cost function (21) as

$$\begin{aligned} J &= (\hat{\boldsymbol{y}} - \boldsymbol{w})^T(\hat{\boldsymbol{y}} - \boldsymbol{w}) + \lambda\tilde{\boldsymbol{u}}^T\tilde{\boldsymbol{u}} \\ &= (\boldsymbol{G}\tilde{\boldsymbol{u}} + \boldsymbol{y}_0 - \boldsymbol{w})^T(\boldsymbol{G}\tilde{\boldsymbol{u}} + \boldsymbol{y}_0 - \boldsymbol{w}) + \lambda\tilde{\boldsymbol{u}}^T\tilde{\boldsymbol{u}} \\ &= c_0 + 2\boldsymbol{g}^T\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{u}}^T\boldsymbol{H}\tilde{\boldsymbol{u}} \end{aligned} \tag{44}$$

where the gradient $\boldsymbol{g}$ and Hessian $\boldsymbol{H}$ are defined as

$$\boldsymbol{g}^T = \boldsymbol{G}^T(\boldsymbol{y}_0 - \boldsymbol{w}) \tag{45}$$

$$\boldsymbol{H} = \boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I} \tag{46}$$

Minimisation of the cost function (44) now becomes a direct problem of linear algebra. The solution in the unconstrained case can be found by setting partial derivative of J with respect to $\tilde{\boldsymbol{u}}$ to zero and yields

$$\tilde{\boldsymbol{u}} = -\boldsymbol{H}^{-1}\boldsymbol{g} \tag{47}$$

This equation gives the whole trajectory of the future control increments and as such it is an open-loop strategy. To close the loop, only the first element of $\tilde{\boldsymbol{u}}$, e.g. $\Delta u(t)$ is applied to the system and the whole algorithm is recomputed at time $t + 1$. This strategy is called *Receding Horizon Principle* and is one of the key issues in the MBPC concept.

To summarise the procedure, it should be noticed, that only two plant characteristics are needed: free response $\boldsymbol{y}_0$ that is changing at each sampling time and step response $G(z^{-1})$ which is in the case of time invariant system needed only once. Moreover, also the Hessian matrix $\boldsymbol{H}$ that should be inverted, contains only information from the step response and can also be calculated beforehand. The calculation of the actual control increment is thus dependent only on weighted sum of past inputs and outputs contained in $\boldsymbol{y}_0$ and forms therefore a linear control law.

4.3 An Example

To make the derivation more clear, let us consider a simple SISO plant with numerator and denominator polynomials given as

$$B(z^{-1}) = 0.4 + 0.1z^{-1}, \qquad A(z^{-1}) = 1 - 0.5z^{-1}$$

and assume that $C(z^{-1}) = 1$. The CARIMA description of the system is of the form

$$\Delta A(q^{-1})y(t) = B\Delta u(t-1) + \xi(t)$$

or

$$y(t) = 1.5y(t-1) - 0.5y(t-2) + 0.4\Delta u(t-1) + 0.1\Delta u(t-2) + \xi(t)$$

Let us now assume the cost function (21) with the parameters $N_1 = 1$, $N_2 = 3$, $N_u = 2$.

The predictions of future output can be obtained if $\xi(t+i) = 0$ and are as follows:

$$\begin{aligned}
\hat{y}(t+1) &= 1.5y(t) - 0.5y(t-1) + 0.4\Delta u(t) + 0.1\Delta u(t-1) \\
\hat{y}(t+2) &= 1.5y(t+1) - 0.5y(t) + 0.4\Delta u(t+1) + 0.1\Delta u(t) \\
\hat{y}(t+3) &= 1.5y(t+2) - 0.5y(t+1) + 0.4\Delta u(t+2) + 0.1\Delta u(t+1)
\end{aligned}$$

According to the assumptions, the term $\Delta u(t+2)$ is equal to zero. The higher output predictions contain the lower output predictions that can be back substituted an yield

$$\begin{aligned}
\hat{y}(t+1) &= 1.5y(t) - 0.5y(t-1) + 0.4\Delta u(t) + 0.1\Delta u(t-1) \\
\hat{y}(t+2) &= 1.75y(t) - 0.75y(t-1) + 0.4\Delta u(t+1) + 0.7\Delta u(t) + 0.15\Delta u(t-1) \\
\hat{y}(t+3) &= 1.875y(t) - 0.875y(t-1) \\
&\quad + 0.7\Delta u(t+1) + 0.85\Delta u(t) + 0.175\Delta u(t-1)
\end{aligned}$$

Stacking all predictions into a vector and separating the terms unknown at time t from the known ones gives

$$\begin{aligned}
\begin{pmatrix} \hat{y}(t+1) \\ \hat{y}(t+2) \\ \hat{y}(t+3) \end{pmatrix} &= \begin{pmatrix} 0.4 & 0 \\ 0.7 & 0.4 \\ 0.85 & 0.7 \end{pmatrix} \begin{pmatrix} \Delta u(t) \\ \Delta u(t+1) \end{pmatrix} \\
&\quad + \begin{pmatrix} 1.5y(t) - 0.5y(t-1) + 0.1\Delta u(t-1) \\ 1.75y(t) - 0.75y(t-1) + 0.15\Delta u(t-1) \\ 1.875y(t) - 0.875y(t-1) + 0.175\Delta u(t-1) \end{pmatrix}
\end{aligned}$$

An alternative to obtain the matrix $\boldsymbol{G}$ would be to perform the long division

$$\frac{B}{\Delta A} = \frac{0.4 + 0.1z^{-1}}{1 - 1.5z^{-1} + 0.5z^{-2}} = 0.4 + 0.7z^{-1} + 0.85z^{-2} + \cdots$$

Now let us assume that the weighting coefficient λ is equal to zero. Inversion of the Hessian matrix gives

$$\boldsymbol{H}^{-1} = \begin{pmatrix} 5.1383 & -6.9170 \\ -6.9170 & 10.8498 \end{pmatrix}$$

Finally, multiplication with $\boldsymbol{g}$ yields the closed-loop expression for the element $\Delta u(t)$

$$\begin{aligned}
\Delta u(t) &= -3.6461y(t) + 1.2351y(t-1) - 0.2470\Delta u(t-1) \\
&\quad + 2.0553w(t+1) + 0.8300w(t+2) - 0.4743w(t+3)
\end{aligned}$$

Simulation results show the behaviour of the closed-loop system in the Fig. 2.

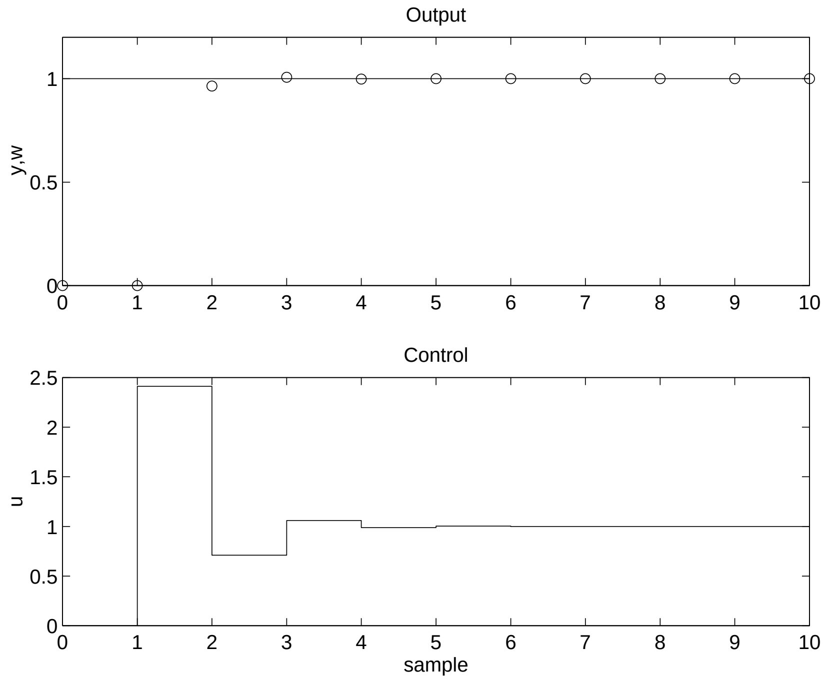

Figure 2: Closed-loop response of the controlled system

4.4 Multivariable GPC

In the same line of thought as in the SISO case, the multivariable GPC algorithm can de derived via Diophantine equations. From the practical point of view, the multivariable controller can come from the prediction equation (31) that holds exactly as before, only the vector and matrix elements are not scalar, but vectors and matrices. If m-input and n-output system is considered, then the matrix $\boldsymbol{G}$ is of the form

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{G}_0 & 0 & \dots & \dots & 0 \\ \boldsymbol{G}_1 & \boldsymbol{G}_0 & 0 & \dots & 0 \\ \vdots & & \ddots & \ddots & \vdots \\ \vdots & & & \boldsymbol{G}_0 & 0 \\ \boldsymbol{G}_{N_2-1} & \dots & & \dots & \boldsymbol{G}_0 \end{pmatrix} \tag{48}$$

with $\boldsymbol{G}_i$ being matrices of dimensions $[n \times m]$ and the overall $\boldsymbol{G}$ has dimensions $[n(N_2 - N_1 + 1) \times mN_u]$.

The vectors $\boldsymbol{y}_{0i}$ and matrices $\boldsymbol{G}_i$ can be obtained analogously as in the singlevariable case from free and step responses of the system.

The drawback of the proposed multivariable derivation are increased dimensions of the matrices involved in matrix inversion routines. The multivariable formulation can be broken into series of SISO GPC calculations if the system denominator matrix $A(z^{-1})$ is diagonal.

4.5 Implementation

As it was stated before, for the actual implementation of the linear GPC algorithm only two process characteristics are needed: step and free responses. The step response can be obtained directly from the process by performing a small step change in one of the manipulated inputs at a time if the process was before in a steady state. The magnitude of the step change is important if the process is non-linear. If the process steady-state gain is approximately known then the magnitude of the input step change should be chosen such that it produces step response approaching desired setpoint value. This strategy may be applied if a non-linear model is used for predictions.

If the polynomials A, B are estimated on-line by the means of a RLS algorithm, the step response is obtained from the polynomial division

$$\frac{B}{A\Delta} = g_0 + g_1 z^{-1} + \cdots + g_{N_1} z^{-N_1} + \cdots + g_{N_2} z^{-N_2} + \cdots \tag{49}$$

The matrix $\boldsymbol{G}$ can be formed from the coefficients of the step response and it is given by the last $N_2 - N_1 + 1$ rows and the first N_u columns of the Toeplitz matrix (42).

The free response is calculated as the process response from the actual initial conditions if the input is fixed to $u(t-1)$. At time t is should hold

$$y(t) = y_0(t) \tag{50}$$

However, the assumption about random-walk disturbance usually results in non-zero disturbance at time t and holds

$$y(t) = y_0(t) + d(t) \tag{51}$$

This disturbance is assumed to be constant in all future predictions. The disturbance and the free responses are thus calculated as

$$d = \sum_{i=0}^{\deg(A)} a_i y_f(t-i) - \sum_{i=0}^{\deg(B)} b_i u_f(t-i-1) \tag{52}$$

$$y_0(t+j) = y_f(t+j) = d - \sum_{i=1}^{\deg(A)} a_i y_f(t-i+j) + \sum_{i=0}^{\deg(B)} b_i \bar{u}(t-i+j-1) \tag{53}$$

where

$$\bar{u}(t-i+j) = \begin{cases} u_f(t-1) & j \geq i \\ u_f(t-i+j) & \text{otherwise} \end{cases} \tag{54}$$

Note: if polynomial P is assumed to be non-unity, then the above is valid if the system denominator A is changed for PA.

To ensure offset-free setpoint following, the polynomial P should be specified subject to condition $P(1) = 1$.

Note: The polynomial C is normally not estimated on-line, but used as a user-design parameter. From relation between state-space and input-oputput approaches, it can be shown that it acts as a observer polynomial and is used for disturbance rejection. More about its choice is given on the page 26.

4.6 Relation to Other Approaches

One of the features of GPC approach is its generality. With different values of its parameters it can be reduced to some well-known controllers:

Mean level control

$$N_1 = 1,\ N_2 \to \infty,\ N_u = 1,\ P = 1,\ \lambda = 0$$

Exact model following (GMV controller)

$$N_1 = 1,\ N_2 = D+1,\ N_u = D+1,\ P = 1,\ \lambda = 0$$

or

$$N_1 = 1,\ N_2 > D,\ N_u = N_2 - D,\ P = 1,\ \lambda = 0$$

Deat-beat control

$$N_1 \geq \deg(B)+1,\ N_2 \geq N_u + N_1 - 1,\ N_u \geq \deg(A)+1,\ P = 1,\ \lambda = 0$$

Pole placement Deat-beat $+P$. Poles are placed at zeros of P.

$$N_1 \geq \deg(B)+1,\ N_2 \geq N_u + N_1 - 1,\ N_u \geq \deg(A)+1,\ P \neq 1,\ \lambda = 0$$

4.7 Continuous-time approaches

Predictive Control is developed in discrete-time domain. The discrete formulation allows for easy prediction generation, because time response of discrete systems can be obtained from polynomial division of the system numerator and denominator.

The analogical formulation for continuous-time systems is by no means so simple. The first serious attempt is given in [11]. The authors show, that the principial polynomial equations remain the same, however, their interpretation is different. For the signal predictions, the Taylor expansion is used. However, the main advantage of MBPC - constraints handling, is lost with this approach.

The approach, that is more realistic while still allowing the constraints is presented in [42]. The main principle is to approximate the future control and output signals as a linear combination of selected continuous-time base functions – in this case B-splines. The real optimised variables become parameters of the splines. It can be shown that both input and output signal approximations are affine functions of these parameters and thus constrained continuous-time predictive control can be solved as a Quadratic Programming task.

The advantage of this approach is truly continuous-control where the choice of the sampling time is not so crucial as in the discrete time case. For the disadvantages, we mention larger number of user parameters and not very clear stability properties.

The real-time results of comparison of this method with GPC and CRHPC in [41] show a very good performance and robustness of this method.

5 Constrained Control

The GPC algorithm derived in the preceding section did not consider presence of constraints. This is not very realistic, as in practice, some kind of constraints is usually present in process control. Most often, inputs are constrained to be between some minimal and maximal values (flows cannot be negative, valves can be opened at 100% maximally) or input rate changes are limited. Usually, there also exist some recommended values of process outputs; these are often formulated as *soft* constraints as opposed to *hard* input constraints.

The ability to handle constraints is one of the key properties of MBPC and caused its spread and popularity also in industry. Nowadays, most of the industrial processes run at the constraints, if not, the process is unnecessarily overdesigned.

One might argue that input constraints can be respected if the calculated control by some control method is subsequently clipped to be within limits. There are at least two reasons not to do so: (i) there is a loss of anticipating action. If the algorithm predicts future behaviour of the system, it might be more correct not to go fully at the constraints at the moment. Otherwise, after some time the process may go totally unstable, out of limits of safety, to an emergency mode. This usually causes heavy economic losses connected with emergency stop and start-up procedures, (ii) if multivariable control is considered, certain influence between input vector elements must be respected. Clipping one input element may cause entirely different transient responses. This phenomenon is called *directionality* of a multivariable plant [5].

The cost function used in GPC is quadratic and of the form (44). If we assume only constraints that are linear with respect to the optimised vector $\tilde{\boldsymbol{u}}$ then the resulting optimisation problem may be casted as the *Quadratic Programming* problem which is known to be convex and for which efficient programming codes exist. The general constrained GPC formulation is thus given as

$$\min_{\tilde{\boldsymbol{u}}} 2\boldsymbol{g}^T\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{u}}^T\boldsymbol{H}\tilde{\boldsymbol{u}} \quad \text{subject to: } \boldsymbol{A}\tilde{\boldsymbol{u}} \geq \boldsymbol{b} \tag{55}$$

Several types of the constraints may be written in the general form:

Input rate limits $\Delta u_{min} \leq \Delta u \leq \Delta u_{max}$:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{I} \\ -\boldsymbol{I} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \mathbf{1}\Delta u_{min} \\ -\mathbf{1}\Delta u_{max} \end{pmatrix}$$

where $\mathbf{1}$ is a vector whose entries are ones.

Input amplitude limits $u_{min} \leq u \leq u_{max}$:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{L} \\ -\boldsymbol{L} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \mathbf{1}u_{min} - \mathbf{1}u(t-1) \\ -\mathbf{1}u_{max} + \mathbf{1}u(t-1) \end{pmatrix}$$

where $\boldsymbol{L}$ is a lower triagonal matrix whose entries are ones.

Output constraints $y_{min} \leq y \leq y_{max}$:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{G} \\ -\boldsymbol{G} \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} \boldsymbol{y}_{min} - \boldsymbol{y}_0 \\ -\boldsymbol{y}_{max} + \boldsymbol{y}_0 \end{pmatrix}$$

Several other types of output constraints can be handled similarly: overshoot, undershoot, monotonicity, etc.

Although input constraints can always be met, presence of output constraints can cause infeasibility of the Quadratic Programming. Therefore from practical point of view, hard output constraints should be changed to soft constraints where amount ε of constraint violation is penalised. In such a case the output constraints are of the form

$$-\varepsilon + y_{min} \leq y \leq y_{max} - \varepsilon, \quad \varepsilon > 0 \tag{56}$$

and the cost function (44) is of the form

$$J = 2\boldsymbol{g}^T\tilde{\boldsymbol{u}} + \tilde{\boldsymbol{u}}^T\boldsymbol{H}\tilde{\boldsymbol{u}} + \boldsymbol{\varepsilon}^T\bar{\boldsymbol{H}}\boldsymbol{\varepsilon} \tag{57}$$

and the variables $\boldsymbol{\varepsilon}$ are added to optimised variables.

6 Stability Results

Any predictive method minimising finite horizon cost function may become unstable in some cases. This can easily be imagined if the system controlled contains right half plane zeros, the output horizon is equal to one, and control penalisation is equal to zero. Inevitably, predictive controller that minimises only output error, is able to set it to zero at each sampling time. The price however is, that the control signals are increasing in magnitude and the system will be unstable.

This is one of the main issues against MBPC. Although the methods may work well in practice, some systems exists in theory for which the methods are highly sensitive [1]. Even more significant is, that there is no clear theory which predicts closed-loop behaviour for arbitrary horizons and control penalisations.

Therefore, two main streams toward stability have been developed. In the first case, some combinations of GPC parameters have been proven to be stabilising. The second line of research has been devoted to methods that overcome the basic GPC drawbacks.

6.1 Stability Results in GPC

Theorem 1 ([6]) *The closed-loop system is stable if the system is stabilisable and detectable and if:*

- $N_2 \to \infty$, $N_u = N_2$, *and* $\lambda > 0$ *or*
- $N_2 \to \infty$, $N_u \to \infty$, $N_u \le N_2 - n + 1$, *and* $\lambda = 0$ *where* n *is the system state dimension.*

Theorem 2 ([6]) *For open-loop stable processes the closed-loop system is stable and the control tends to a mean level law for* $N_u = 1$ *and* $\lambda = 0$ *as* $N_2 \to \infty$.

Theorem 3 ([6]) *The closed-loop system is equivalent to a stable state deat-beat controller if*

1. *the system is observable and controllable and*
2. $N_1 = n$, $N_2 \ge 2n - 1$, $N_u = n$, *and* $\lambda = 0$ *where* n *is the system state dimension.*

For more thorough discussion on stability properties of GPC and its relations to LQ control see [1].

6.2 Terminal Constraints

The first approach that forces MBPC methods to be stable is based on the state terminal constraints. The methods use the results given in [22, 27] where stability of time-varying discrete linear systems is discussed if the MBPC receding horizon quadratic cost is minimised. Roughly speaking, the system is stable if it is subject to the moving-terminal constraint on final states

$$\boldsymbol{x}(t + N_2) = \mathbf{0} \tag{58}$$

Several different algorithm have emerged that are based on this result:

6.2.1 CRHPC

SIORHC [35] and **CRHPC** [8] were developed independently, but are in fact equivalent. The idea behind these techniques is an equivalent of the state terminal constraint within input/output system description. Hence, these methods optimise the usual quadratic function over finite horizons subject to condition that the output exactly matches a reference value over a future constraint range (after $t + N_2$). Some degrees of freedom force the output to stay at setpoints while the remaining degrees of freedom are available to minimise the cost function. The output constraint description is

$$y(t + N_2 + i) = w(t + N_2), \quad i = 1, \dots, n \tag{59}$$

and n is the dimension of the system state vector.

Although the output constraints are added to the original formulation, the solution in the unconstrained case can still be found analytically.

Theorem 4 (SIORHC [35]) *Let the system polynomials ΔA and B are coprime. If $\lambda > 0$ and $\deg(B) \leq \deg(A) + 1$ then provided that*

$$N_2 \geq n = \max(\deg(A) + 1, \deg(B))$$

- *the SIORHC control law is unique;*
- *SIORHC stabilises the plant, and, irrespective of $\deg(A), \deg(B), \lambda$, for*

  $$N_2 = n$$

  yields a state deat-beat closed-loop system;
- *whenever stabilising, SIORHC yields asymptotic rejection of constant disturbances and offset free closed-loop system.*

Derivation As usual in GPC, consider predicted output to be of the form

$$\hat{\boldsymbol{y}} = \boldsymbol{G}\tilde{\boldsymbol{u}} + \boldsymbol{y}_0 \tag{60}$$

where all vectors are stacked from $t + 1$ to $t + N_2$. After this time, up to $t + N_2 + n$, output predictions are constrained to be equal to setpoint and future control increments to zero

$$\bar{\boldsymbol{w}} = \bar{\boldsymbol{G}}\tilde{\boldsymbol{u}} + \bar{\boldsymbol{y}}_0 \tag{61}$$

where the vector $\bar{\boldsymbol{y}}_0$ denotes free response between $t + N_2 + 1$ and $t + N_2 + n$ and $\bar{\boldsymbol{w}}$ is vector of $w(t + N_2)$ of corresponding dimension.

The cost function is as usual of the form

$$J = (\hat{\boldsymbol{y}} - \boldsymbol{w})^T(\hat{\boldsymbol{y}} - \boldsymbol{w}) + \lambda\tilde{\boldsymbol{u}}^T\tilde{\boldsymbol{u}} \tag{62}$$

subject to the constraint (61). This is solved analytically by the Lagrange multipliers. Let us denote $\bar{\boldsymbol{h}} = \bar{\boldsymbol{w}} - \bar{\boldsymbol{y}}_0$ and $\boldsymbol{h} = \boldsymbol{w} - \boldsymbol{y}_0$. Hence

$$J = (\boldsymbol{G}\tilde{\boldsymbol{u}} - \boldsymbol{h})^T(\boldsymbol{G}\tilde{\boldsymbol{u}} - \boldsymbol{h}) + \lambda\tilde{\boldsymbol{u}}^T\tilde{\boldsymbol{u}} + \boldsymbol{x}^T(\bar{\boldsymbol{G}}\tilde{\boldsymbol{u}} - \bar{\boldsymbol{h}}) \tag{63}$$

Partial derivatives of J with respect to $\tilde{\boldsymbol{u}}, \boldsymbol{x}$ ($\boldsymbol{x}$ are the Lagrange multipliers) are zero. We obtain a system linear equations as follows

$$\begin{pmatrix} 2(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}) & \bar{\boldsymbol{G}}^T \\ \bar{\boldsymbol{G}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \tilde{\boldsymbol{u}} \\ \boldsymbol{x} \end{pmatrix} = \begin{pmatrix} 2\boldsymbol{G}^T\boldsymbol{h} \\ \bar{\boldsymbol{h}} \end{pmatrix} \tag{64}$$

The block matrix inversion formula from [21] states

$$\begin{pmatrix} A^{-1} & D \\ C & B \end{pmatrix}^{-1} = \begin{pmatrix} A + AD\Delta CA & -AD\Delta \\ -\Delta CA & -\Delta \end{pmatrix}, \quad \Delta^{-1} = B - CAD \tag{65}$$

Therefore, the future control increment vector is given as

$$\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{G}}(\boldsymbol{I} - \bar{\boldsymbol{G}}^T\boldsymbol{Q}\bar{\boldsymbol{G}}\tilde{\boldsymbol{G}})\boldsymbol{G}^T\boldsymbol{h} + \tilde{\boldsymbol{G}}\bar{\boldsymbol{G}}^T\boldsymbol{Q}\bar{\boldsymbol{h}} \tag{66}$$

where

$$\tilde{\boldsymbol{G}} = \left(\boldsymbol{G}^T\boldsymbol{G} + \lambda\boldsymbol{I}\right)^{-1} \tag{67}$$

$$\boldsymbol{Q} = \left(\bar{\boldsymbol{G}}\tilde{\boldsymbol{G}}\bar{\boldsymbol{G}}^T\right)^{-1} \tag{68}$$

6.2.2 SGPC

Another method that can be shown to be equivalent to the preceding methods is **SGPC** [24]. Its advantages are more efficient computational implementation and better numerical robustness.

In this approach is GPC invoked after the application of the stabilising feedback control law

$$Y(q^{-1})\Delta u(t) = c(t) - X(q^{-1})y(t) \tag{69}$$

where the polynomials X, Y are calculated as deat-beat controller from the Diophantine equation

$$\Delta AY + BX = 1 \tag{70}$$

and where $c(t)$ denoted the reference signal for the closed-loop system that forms the vector of optimised variables. The deat-beat controller results in the control and output predictions of the form

$$y(t) = B(q^{-1})c(t) \tag{71}$$

$$u(t) = A(q^{-1})c(t) \tag{72}$$

These can be simulated forward in time to give the vectors of future output and control predictions and treated in the same way as in GPC.

Hence, this methods optimises future reference trajectory subject to (terminal) constraint that this trajectory should be equal to the desired setpoint after some horizon.

It can be shown that SGPC is equivalent to pole-placement method with the controller polynomials Y_r, X_r given from the Diophantine equation

$$\Delta AY_r + BX_r = P_r \tag{73}$$

and its stability depends on the roots of the P_r polynomial.

Theorem 5 (SGPC [24]) *For $N_2 \geq \deg(A) + 1 + N_c$, where N_c is the number of reference points optimised, is SGPC stable for any N_c.*

6.2.3 YKPC

The last approach within finite horizon formulation is the predictive control algorithm based on Youla Kučera parametrisation of all stabilising controllers (**YKPC** [17]).

As in the previous approach, in the first step a nominal controller with pole-placement technique is calculated and a nominal controller is given as a solution of 2 Diophantine equations

$$P_0 \Delta A + Q_0 B = M \tag{74}$$
$$\Delta S_0 + B R_0 = M \tag{75}$$

where M is the desired closed-loop polynomial and a two degree of freedom controller with integral action is defined as $Q/\Delta P$ (feedback) and $R/\Delta P$ (feedforward).

The minimum degree controller P_0, Q_0, R_0 only serves as a basis to find an expression for the set of all stabilizing controllers. Among these controllers the one is chosen, that minimises the GPC cost function.

The expression of such controllers (Youla-Kučera parametrisation) is as follows:

Theorem 6 (YK Controllers [17]) *A controller $(P(z), Q(z), R(z))$ gives rise to the closed-loop denominator matrix $M(z)$ if and only if it can be expressed as*

$$P = P_0 + ZB \tag{76}$$
$$Q = Q_0 - ZA\Delta \tag{77}$$
$$R = R_0 + \Delta X \tag{78}$$

X, Z are assumed to be polynomials for simplicity. Their coefficients form the vector of the optimised parameters.

Stability is proved as in the previous approaches via terminal constraint. It is interesting to note, that in this approach the state terminal constraint need not to be specified and is implicitly assured.

Theorem 7 (Choice of horizons [17]) *Let $n = \max(\deg(X), \deg(Z))$, $N_1 = 1$ and let the horizons be equal or greater than*

$$N_2 = \deg(B) + n \tag{79}$$
$$N_u = \deg(A\Delta) + n \tag{80}$$

Further assume that the sequences w, d (reference, disturbance) are bounded. Then unconstrained YKPC is uniformly asymptotically stable.

This controller is time-varying in spite of the fact that the system is assumed to be time-invariant.

6.3 Infinite Horizons

Another line of research has been focused into reformulation of the basic GPC method when N_2, N_u are infinity. Of course, if such a method can be implemented, stability problems disappear. However, number of the optimised parameters (future control moves) is also infinity and the original problem is untractable. Therefore, several suboptimal algorithms have emerged. The basic principle of all of them is to leave $N_2 = \infty$ but to play with N_u or with its equivalents.

Rawlings and Muske [39] developed a method in state-space formulation where the number of control moves N_u is finite. The feedback gain is calculated via recursive ARE.

Theorem 8 (Stable plants [39]) *For stable system matrix $\boldsymbol{A}$ and $N_u \geq 1$ is the receding horizon controller stabilising.*

Theorem 9 (Unstable plants [39]) *For stabilisable plant $(\boldsymbol{A}, \boldsymbol{B})$ with r unstable modes and $N_u \geq r$ is the receding horizon controller stabilising.*

Constrained control is also dealt with in their approach. The requirement added to the previous theorems is that the initial state at time t is feasible (within constraints).

The SGPC and YKPC methods can be modified to use both input and output horizons infinite [15, 16, 43]. The SGPC approach utilises finite reference sequence as the vector of optimised variables. The solution if found via Lyapunov equation.

The YKPC method utilises as the optimised variables coefficients of the Youla-Kučera polynomials. It is shown that in the unconstrained case the optimal predictive controller coincides with the nominal pole-placement controller whose poles are calculated via spectral factorisation equation - hence it is the standard LQ controller. If the constraints are active, piece-wise linear controller results.

6.4 Finite Terminal Penalty

The third approach to MBPC stability is to adopt a finite input and state horizon with a finite terminal weighting matrix [29]. With a state-space formulation

$$\boldsymbol{x}(t+1) = \boldsymbol{A}(t)\boldsymbol{x}(t) + \boldsymbol{B}(t)\boldsymbol{u}(t) \tag{81}$$

is the cost given given as

$$\begin{aligned} J &= \boldsymbol{x}^T(t+N)\boldsymbol{W}(t+N)\boldsymbol{x}(t+N) \\ &\quad + \sum_{i=0}^{N} \Big(\boldsymbol{x}^T(t+i)\boldsymbol{Q}(t+i)\boldsymbol{x}(t+i) + \boldsymbol{u}^T(t+i)\boldsymbol{R}(t+i)\boldsymbol{u}(t+i) \Big) \end{aligned} \tag{82}$$

If $\boldsymbol{W}(t+N) \to \infty$ then the terminal state constraint approach result. However, also some “smaller” terminal penalty matrix W can give a stabilising receding horizon control.

Theorem 10 ([29]) *Assume that the terminal weighting matrix $\boldsymbol{P}(t+k)$ satisfies the following matrix difference inequality for some matrix $\boldsymbol{H}(t)$*

$$\boldsymbol{P}(k) \geq \boldsymbol{F}^T(k)\boldsymbol{P}(k+1)\boldsymbol{F}(k) + \boldsymbol{Q}(k) + \boldsymbol{H}^T(k)\boldsymbol{R}(k+1)\boldsymbol{H}(k), \ \forall k \in [N, \infty) \tag{83}$$

where

$$\boldsymbol{F}(k) = \boldsymbol{A}(k) + \boldsymbol{B}(k)\boldsymbol{H}(k) \tag{84}$$

Further suppose that $\boldsymbol{Q}(k)$ is positive definite. Then the receding horizon control which stems from the optimisation problem minimising the performance index J, asymptotically stabilises the system. In addition, if $\boldsymbol{A}(k), \boldsymbol{Q}(k)$, and $\boldsymbol{P}(k)$ are bounded above $\forall k \geq 0$, then the receding horizon control exponentially stabilises the system.

7 Tuning

Let us recall the GPC parameters: horizons N_1, N_2, N_u, control weighting λ, and output weighting polynomial P. The effects of a change of the parameters are strongly coupled and the strategies dealing with adjustment of GPC parameters usually adjust only one parameter while all others are at some default values.

Several researchers have devoted their attention to different tuning strategies. Here we consider results given in [31, 49, 50].

7.1 Tuning based on First Order Model

The tuning strategy in [49] is based on the analysis of the first order system with time delay and proposed for DMC. However, it is also applicable to GPC if $P = 1$.

The strategy is given as follows:

1. Approximate the process dynamics with a first order plus dead time model:

$$F(s) = \frac{K}{Ts+1}e^{-Ds} \tag{85}$$

2. If the sampling time has not yet been specified, select it as the larger value that satisfies

$$T_s \leq 0.1T, \qquad T_s \leq 0.5D \tag{86}$$

3. Calculate the discrete dead time D_d (rounded to the next integer)

$$D_d = D/T_s + 1 \tag{87}$$

4. Set $N_1 = 1$, and

$$N_2 = 5T/T_s + D_d \tag{88}$$

5. Select the control horizon N_u (usually between 1–6) and calculate the control weighting λ as

$$f = \begin{cases} 0 & N_u = 1 \\ \frac{N_u}{500}\left(\frac{3.5T}{T_s} + 2 - \frac{N_u - 1}{2}\right) & N_u > 1 \end{cases} \tag{89}$$

$$\lambda = fK^2 \tag{90}$$

7.2 Multivariable Tuning based on First Order Model

The tuning strategy in [50] is a generalisation of the previous approach to multivariable systems with R outputs and S inputs. It is based on the analysis of the first order system with time delay.

The strategy is given as follows:

1. Approximate the process dynamics of all controller output-process variable pairs with first order plus dead time models:

$$\frac{y_r(s)}{u_s(s)} = \frac{K_{rs}}{T_{rs}s+1}e^{-D_{rs}s} \tag{91}$$

2. Select the sampling time as close as possible to:

$$T_{rs}^s = \max(0.1T_{rs}, 0.5D_{rs}), \quad T_s = \min(T_{rs}^s) \tag{92}$$

3. Set $N_1 = 1$, compute the prediction horizon N_2:

$$N_2 = \max\left(\frac{5T_{rs}}{T_s} + k_{rs}\right), \quad k_{rs} = \frac{D_{rs}}{T_s} + 1 \tag{93}$$

4. Select a control horizon N_u, equal to 63.2% of the settling time of the slowest sub-process in the multivariable system:

$$N_u = \max\left(\frac{T_{rs}}{T_s} + k_{rs}\right) \tag{94}$$

5. Select the controlled variable weights γ_r, to scale process variable measurements to similar magnitudes

6. Compute the control weightings λ_s as

$$\lambda_s = \frac{M}{500}\sum_{r=1}^{R}\left[\gamma_r K_{rs}^2 \left\{N_2 - k_{rs} - \frac{3}{2}\frac{T_{rs}}{T_s} + 2 - \frac{M-1}{2}\right\}\right] \tag{95}$$

Fine tuning of the method is performed by increasing the corresponding γ_r of the process variable for which tighter control is desired and increasing the corresponding λ_s of the manipulated variable for which less aggressive moves are desired.

7.3 Output Horizon Tuning

This tuning strategy assumes active tuning parameter to be the output horizon N_2 with all other fixed at the values

$$N_1 = 1,\ N_u = 1,\ P = 1,\ \lambda = 0 \tag{96}$$

It is well known that if $N_2 \to \infty$, mean-level controller results. This controller is rather conservative as its speed is the same as step responses.

The other limit for N_2 is the value of process dead time. If $N_2 = D+1$, where D represents process dead time, then we have Minimum Variance (MV) controller known to be unstable for non-minimum phase plants.

The practical range for N_2 can be specified as

$$D + 1 < N_2 \le t_r/T_s \tag{97}$$

where t_r is the time when process reaches after input step change about 90% of its final value and T_s is the sampling time.

If the process is uncertain, it is better to start with larger value of N_2. The minimum value of N_2 for non-minimum phase plants should be such that $\sum_i g_i$ has the same sign as the process gain.

7.4 λ Tuning

In this case is the active tuning parameter penalisation of control moves λ. All other parameters are fixed as

$$N_1 = \deg(B) + 1,\ N_u = \deg(A) + 1,\ N_2 \ge N_u + N_1 - 1 \approx t_r/T_s,\ P = 1 \tag{98}$$

With λ equal to zero, deat-beat controller results. This is in majority of cases too rapid. Hence, with increasing value of λ is the controller made more conservative. It might be shown that the closed-loop poles converge to the open loop poles if $\lambda \to \infty$. Thus λ tuning is not recommended for unstable plants.

It has been found that to desensitise the closed-loop system to changes in process dynamics, the actual λ should be proportional to $B(1)^2$:

$$\lambda = \lambda_0 B(1)^2 \tag{99}$$

with λ_0 being a constant.

To determine a starting value of λ, the following relation can be used:

$$\lambda = \frac{m\ \mathrm{tr}(G^T G)}{N_u} \tag{100}$$

and m is a factor of detuning the controller relative to deat-beat control. The control increments are approximately reduced by a factor $m+1$ compared to that of the deat-beat strategy.

From this starting value of λ, an initial guess for λ_0 can be determined from (99).

7.5 Tuning based on Model Following

As it has been shown before, the P polynomial can be used to generate reference trajectory w/P. GPC can be set up to follow this trajectory exactly and so to place the closed-loop poles at the process zeros. In order to have a more practical controller, the model following can be detuned. This may be accomplished by either increasing N_2 or λ.

The fixed parameters are as follows:

$$N_1 = 1,\ N_u = \deg(A) + 1,\ N_2 \geq N_u + D \approx t_r/T_s,\ \lambda = 0 \tag{101}$$

Most often, the models $M = 1/P$ are of the first and the second order. If the first order closed-loop model is assumed to be of the form

$$M(s) = \frac{1}{Ts+1} \tag{102}$$

then its discrete equivalent is

$$M(z^{-1}) = \frac{(1-p_1)z^{-1}}{1-p_1 z^{-1}} \tag{103}$$

where $p_1 = \exp(-T_s/T)$. The polynomial P can thus be chosen as (cf. equation (13))

$$P(z^{-1}) = \frac{1-p_1 z^{-1}}{1-p_1} \tag{104}$$

and $P(1)$ is equal to 1 to ensure offset-free behaviour. This model is applicable mainly for simpler plants as the first order trajectory may sometimes generate excessive control actions.

The second order model can be of the form

$$M(s) = \frac{1}{T^2 s^2 + 2T\xi s + 1} \tag{105}$$

Its discrete time equivalent is

$$M(z^{-1}) = \frac{n_1 z^{-1} + n_2 z^{-2}}{1 + p_1 z^{-1} + p_2 z^{-2}} \tag{106}$$

where

$$p_1 = -2\exp\left(\frac{-\xi T_s}{T}\right)\cos\left[\left(\frac{T_s}{T}\right)\sqrt{1-\xi^2}\right] \tag{107}$$

$$p_2 = \exp\left(\frac{-2\xi T_s}{T}\right) \tag{108}$$

Ignoring the numerator dynamics, the polynomial P may be specified as

$$P(z^{-1}) = \frac{1 + p_1 z^{-1} + p_2 z^{-2}}{1 + p_1 + p_2} \tag{109}$$

The dominant time constant of the closed-loop system is approximately $2T$ and the fractional overshoot is solely a function of the damping factor ξ:

$$o_v = \exp\left[\frac{-\pi\xi}{\sqrt{1-\xi^2}}\right] \tag{110}$$

and thus the user can then specify desired rise time and overshoot and translate these settings into an appropriate P polynomial.

7.6 The C polynomial

The CARIMA model includes knowledge about the disturbance properties in the polynomial C. This can be estimated on-line using a suitable recursive identification algorithm. However, this is rather difficult in practice, because the convergence of the C polynomial coefficients is rather slow.

Therefore, a more realistic approach is to set C by user directly. The value that has been suggested as a default in the literature is of the form

$$C = (1 - 0.8z^{-1})^2 \tag{111}$$

Another possibility that follows from optimal LQ theory is to calculate it as a stable polynomial from spectral factorisation of the denominator polynomial A as

$$C^* C = A^* A \tag{112}$$

8 Examples

In this section are shown some examples of the GPC control algorithm. The first examples show some effects of the tuning parameters that have been described in the previous sections.

The example dealing with control of a tubular chemical reactor describes adaptive implementation of GPC. A linear model is estimated on-line with a RLS algorithm and successively controlled.

The bioreactor control example demonstrates possibility of using a non-linear model for predictions. Here, an artificial neural network model is used. Comparison with adaptive control based on a linear model shows some drawbacks of adaptive methods applied to non-linear processes.

Finally, the pH control example shows a real-time control problem. It is demonstrated that GPC is able to control such a strongly non-linear process.

8.1 A Linear Example

Let us consider a linear continuous-time system with transfer function

$$F(s) = \frac{1}{(s+1)^2}$$

that is discretised with the sampling time $T_s = 1$.

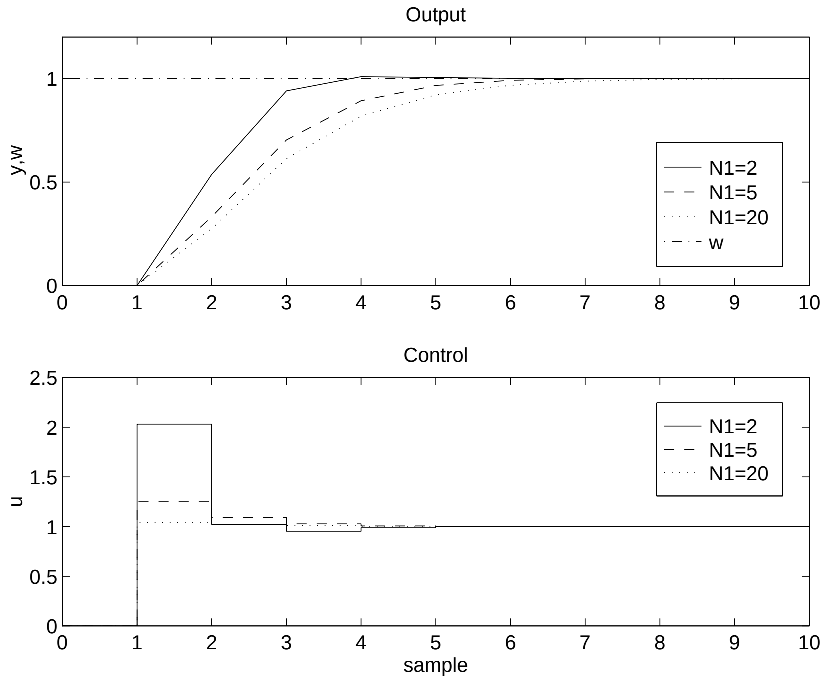

Figure 3: Increasing value of N_2 towards mean-level control

Two simulation runs were performed. In the first one, mean level settings were given. The GPC parameters were $N_1 = 1$, $N_u = 1$, $\lambda = 0$, and N_2 varied between 2–20. The results are shown in Fig. 3 and illustrate that with increasing N_2 are the control action smoother, more conservative and approach input step change.

In the second simulation was λ given as the varying parameter. The settings of other parameters were $N_1 = 3$, $N_u = 3$, $N_2 = 5$ which for $\lambda = 0$ gives deat-beat controller. Increasing λ makes more weight on control increments and slows down the controller. The results are shown in Fig. 4.

8.2 Adaptive Control of a Tubular Reactor

This example is in full length described in [13]. An ideal plug-flow tubular chemical reactor with an exothermic consecutive reaction $A \to B \to C$ in the liquid phase and with counter-current cooling is considered. It is assumed that A is the educt, B is the desired product and C the unwanted by-product of the reaction. Such reactors are central components of many plants in the chemical industry and exhibit highly nonlinear dynamics.

Mathematical model of this reactor is given as

$$\frac{\partial c_A}{\partial t} = -v\frac{\partial c_A}{\partial z} - k_1 c_A \tag{113}$$

$$\frac{\partial c_B}{\partial t} = -v\frac{\partial c_B}{\partial z} k_1 c_A - k_1 c_B \tag{114}$$

$$\frac{\partial T}{\partial t} = -v\frac{\partial T}{\partial z} + \frac{q_r}{\rho c_p} - \frac{4U_1}{d_1 \rho c_p}(T - T_w) \tag{115}$$

$$\frac{\partial T_w}{\partial t} = \frac{4}{(d_2^2 - d_1^2)\rho_w c_{pw}}[d_1 U_1 (T - T_w) + d_2 U_2 (T_c - T_w)] \tag{116}$$

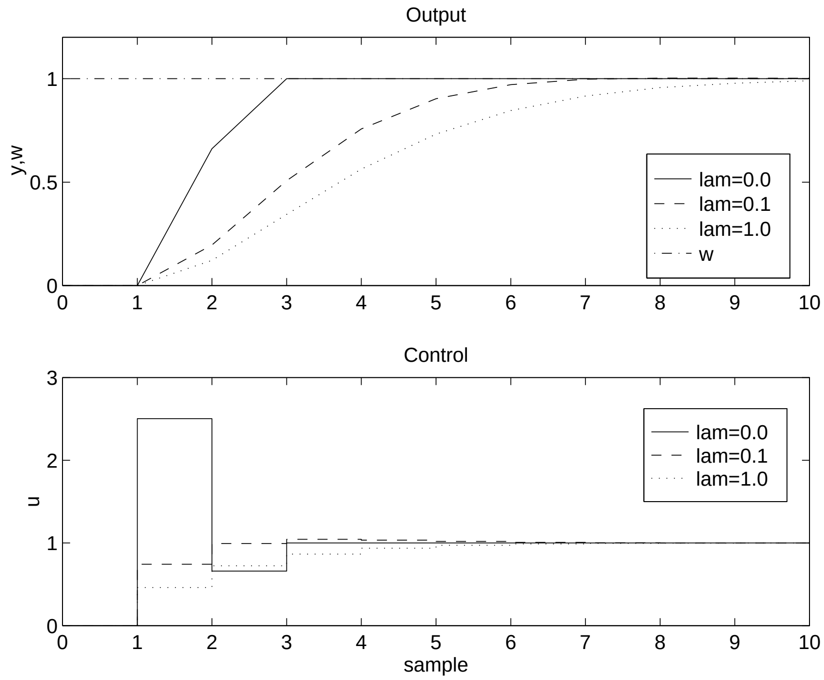

Figure 4: Increasing value of λ - detuned deat-beat control

$$\frac{\partial T_c}{\partial t} \;=\; v\frac{\partial T_c}{\partial z} + \frac{4n_1 d_2 U_2}{(d_3^2 - n_1 d_2^2)\rho_c c_{pc}}(T_w - T_c) \tag{117}$$

with initial conditions

$$\begin{array}{lll} c_A(z,0) = c_A^s(z), & T(z,0) = T^s(z), & T_c(z,0) = T_c^s(z) \\ c_B(z,0) = c_B^s(z), & T_w(z,0) = T_w^s(z) & \end{array} \tag{118}$$

and with boundary conditions

$$\begin{array}{ll} c_A(0,t) = c_{A0}(t), & T(0,t) = T_0(t) \\ c_B(0,t) = c_{B0}(t), & T_c(L,t) = T_{cL}(t). \end{array} \tag{119}$$

Here t is time, z space variable along the reactor, c are concentrations, T are temperatures, v are fluid velocities, d are diameters, ρ are densities, c_p are specific heat capacities, and U are heat transfer coefficients. The subscripts are $(.)_w$ for metal wall of tubes, $(.)_c$ for coolant, and $(.)^s$ for steady-state values. The reaction rates k and the heat of reactions are expressed as

$$k_j \;=\; k_{0j}\exp(-E_j/RT)\;, \quad j = 1,2 \tag{120}$$

$$q_r \;=\; h_1 k_1 c_A + h_2 k_2 c_B \tag{121}$$

where k_0 are exponential factors, E are activation energies and h are reaction enthalpies.

Assuming the reactant temperature measurement along the reactor at points z_j, the mean temperature profile can be expressed as

$$T_m(t) = \frac{1}{n}\sum_{j=1}^{n} T(z_j,t) \tag{122}$$

where n is the number of measured temperatures along the reactor.

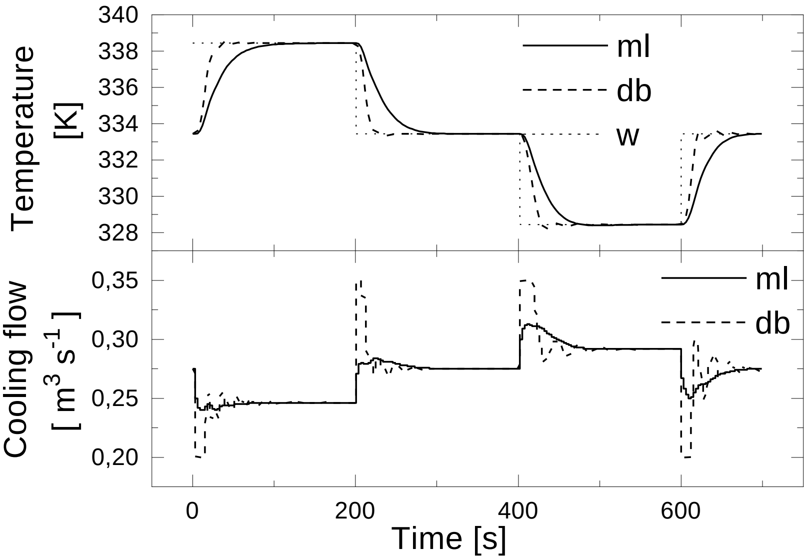

Figure 5: Comparison of deat-beat (db) and mean-level (ml) control strategy

The input variable the value of q_c has been assumed to be constrained in the interval

$$0.2 \leq q_c \leq 0.35 \tag{123}$$

For the control purposes both the manipulated input and the controlled output were defined as scaled deviations from their steady-state values

$$u(t) = \frac{q_c(t) - q_c^s}{q_c^s}, \quad y(t) = \frac{T_m(t) - T_m^s}{T_m^s}. \tag{124}$$

This scaling helps to obtain variables with approximately same magnitude and reduces the possibility of ill-conditioned control problem and round-off errors.

The sampling time was chosen $T_s = 3s$ and the reactor was on-line identified as SISO discrete system with $\deg(A) = 2,\ \deg(B) = 3$ of the form

$$y(t) = -a_1 y(t-1) - a_2 y(t-2) + b_1 u(t-1) + b_2 u(t-2) + b_3 u(t-3) + d_c + \xi$$

The estimation method used is the recursive least-squares algorithm LDDIF with exponential and directional forgetting [2, 26]. The value of exponential forgetting was set to 0.8 and the minimum of the covariance matrix was constrained to $0.01\boldsymbol{I}$. The purpose of these settings was to improve tracking properties of the estimation algorithm.

The result of the first simulation is shown in Fig. 5. It shows comparison of two GPC settings: mean-level (ml) and deat-beat (db) control.

Upper graph represents behaviour of the controlled variable T_m together with its reference value and the lower graph manipulated variable q_c.

The values of GPC tuning parameters $[N_1, N_2, N_u, \lambda]$ were $[1, 15, 1, 10^{-1}]$ for mean-level and $[3, 7, 3, 10^{-5}]$ for deat-beat, respectively. These values correspond to slow open loop response (ml) and the fastest deat-beat response. The polynomials P, C were set to 1 as the effect of disturbances is very small. One can notice that the deat-beat control strategy uses actively constraints on manipulated variable defined by Eq. (123).

The purpose of the second simulation was to investigate the behaviour of GPC with respect to unmeasured disturbances. The output variable was corrupted by measurement noise with

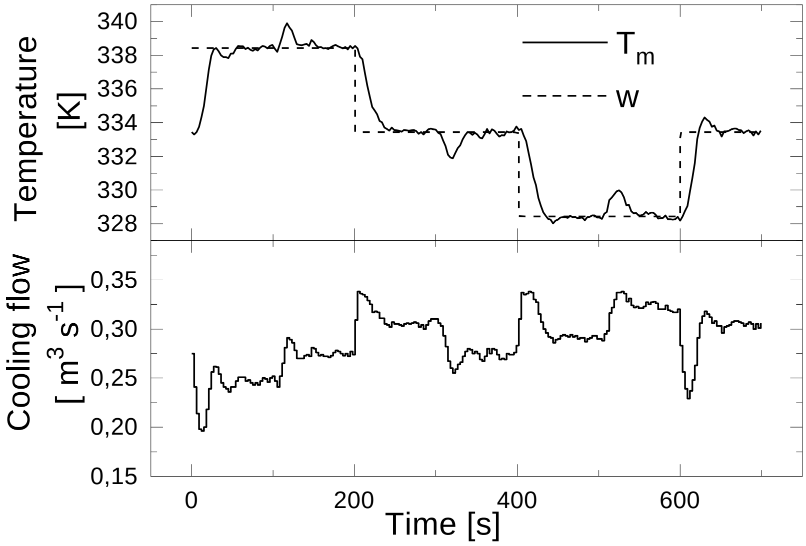

Figure 6: Effect of step disturbances in inlet concentration.

variance 0.1K. The inlet concentration c_{A0} of the component A varied in steps and was given as

t	0	100	300	500
$c_{A0} - c_{A0}^s$	0	0.1	0	0.1

Due to the presence of disturbances, the design polynomials P, C were used. The polynomial C attenuates effects of measurement noise and the polynomial P shapes responses of the closed-loop system subject to load disturbances in c_{A0} and also to reference step changes. The degrees of the polynomials were chosen 1 and their values as

$$P = 0.6 - \frac{2}{3}z^{-1}, \quad C = 1 - 0.8z^{-1}.$$

The GPC controller was implemented with the mean-level strategy and had the values of the tuning parameters given as $[1, 15, 1, 10^{-1}]$.

The result of the simulation is shown if Fig. 6. One can notice that the behaviour of GPC controller was very good and no abrupt control actions can be observed. Also the controlled variable tracks the reference temperature fast and the effects of load changes in c_{A0} are suppressed very well.

8.3 Neural Network based GPC

This example is in full length described in [46]. It compares adaptive GPC based on linear model (AGPC) and implemented in the same way as in the previous example and GPC based on non-linear neural network model (GPCNN).

The process studied was a bioprocess that describes the growth of Saccharomyces cerevisiae on glucose. The oxygen concentration c_o and the dilution rate D_g have been selected as the controlled and the manipulated variables, respectively.

A feedforward ANN plant model with third order input dynamics and one hidden layer was used. This means six neurons in the input layer with signals

$$y(t-1), y(t-2), y(t-3), u(t-1), u(t-2), u(t-3)$$

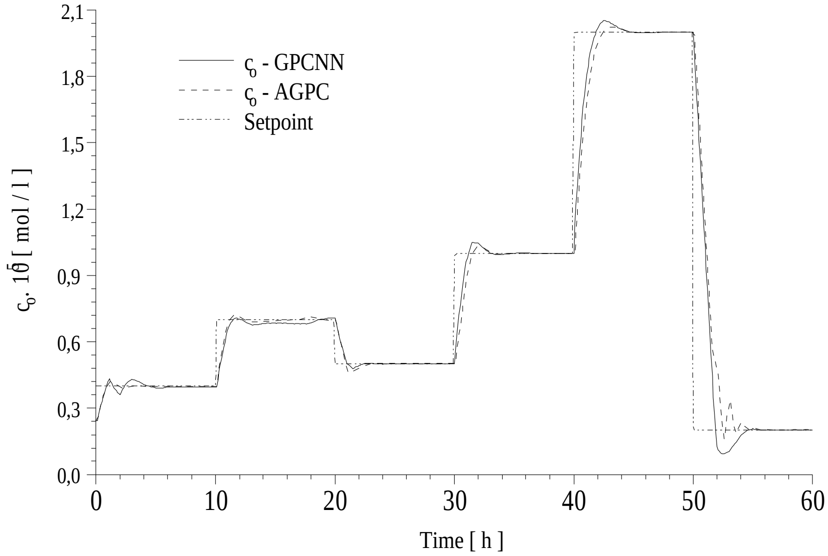

Figure 7: Trajectories of the oxygen concentration (the process output)

For the calculation of the step response, the ANN inputs are

$$y(t-1), y(t-1), y(t-1), u_n, u(t-1), u(t-1)$$

where the step change magnitude u_n was specified as

$$u_n = u(t-1) + \frac{w - y(t-1)}{k}$$

and the process static gain k was determined from the step response estimated in the previous sampling period. The static gain k was initially set equal to 1. To take into account the fact that the initial conditions are not equal to zero and the step input is not of unit value, the ANN approximation of the step response is subsequently normalised.

For the free response the ANN inputs are

$$u(t-1), u(t-2), u(t-3), y(t-1), y(t-2), y(t-3)$$

and it is assumed that the input is constant in the future.

The sampling period was set equal to 0.5 h. A training and validation data sets (800 input-output pair samples) were obtained using a pseudo random binary sequence input. The conjugate gradients algorithm was used as a learning method and a genetic algorithm was used for the initialization of the ANN weights.

For the AGPC, a third order discrete model has been considered for process modelling. The model parameters have been estimated using the parameters estimation algorithm LD-DIF [26].

The GPC parameters were $N_1 = 1, N_2 = 14, N_u = 4, \lambda = 0.1$. The obtained profiles of the process output controlled by the AGPC (dashed line) and the GPCNN (solid line) are shown in Fig. 7. Figure 8 shows the profiles of the control actions generated by AGPC (dashed line) and GPCNN (solid line), respectively. As it can be seen from these figures, both algorithms achieve good results. When a large change of the setpoint occurs (see Fig. 8, t=50 h), the GPC based on linear model leads to a generation of a bad transient behaviour. Unlike the AGPC, the GPC based on ANN generates a smooth control action which leads to a good control behaviour. This behaviour was expected as the AGPC is based on linear model.

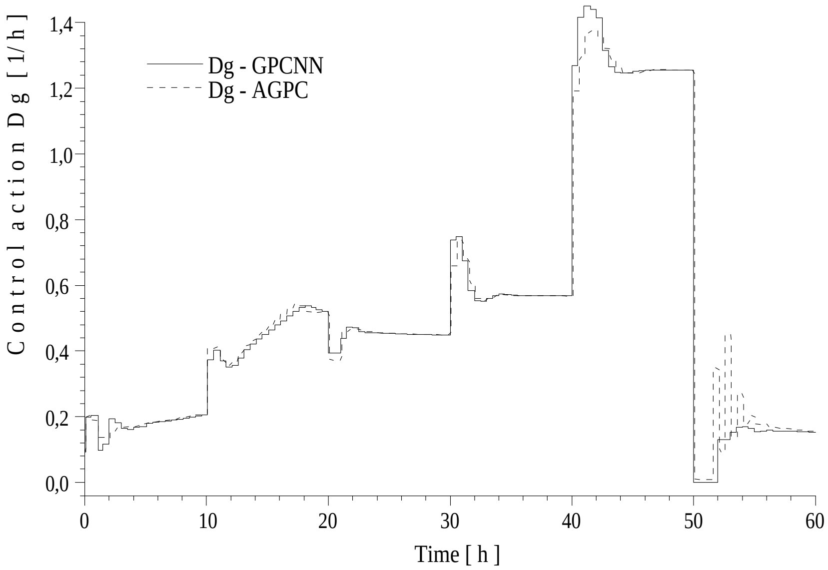

Figure 8: Trajectories of the dilution rate (the manipulated variable)

Owing to the nonlinear characteristic of the bioprocess, a large change of the setpoint or some disturbance can bring the process into other operating points with different dynamical properties.

8.4 pH Control

This example is in full length given in [14]. The experimental pH control has been studied at University of Dortmund.

The neutralisation plant to be controlled consists of a laboratory-scale continuous stirred tank with two inlets and one outlet (see Fig. 9), in which acetic acid is neutralized with sodium hydroxide.

The hold-up of the tank is 5.57 l, the concentrations of the acid and the sodium hydroxide solution streams are approximately 0.01 mol/l. The acid flow rate F_A is fixed at 0.33 l min^{-1}, whereas the NaOH flow F_B is manipulated by the controller. In order to obtain the necessary precision of the flow rates diaphragm pumps were chosen. All control actions are performed by a PC-based control system. The flow F_B is controlled by the modulation of an impulse frequency f, which leads to a quantisation of the control amplitude because the frequency can assume only certain discrete values.

In the tank, the following reaction takes place:

$$\mathrm{NaOH} + \mathrm{CH_3COOH} \rightleftharpoons \mathrm{CH_3COONa} + \mathrm{H_2O} \tag{125}$$

Due to the incomplete dissociation of acetic acid in water and the equilibrium reaction with sodium hydroxide the system behaves like a buffer solution between pH 4 and 6.5. Consequently, the process gain varies extremely over the range of pH-values that can be controlled.

The controlled variable pH and the control variable F_B have been scaled for control and identification purposes as

$$y = \frac{\mathrm{pH} - 7}{7} \qquad u = \frac{F_B - F_B^s}{F_B^s} \tag{126}$$

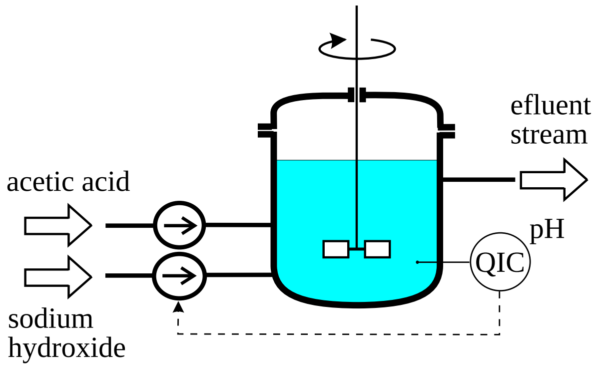

Figure 9: Neutralisation plant

where F_B^s denotes the approximate steady state value of F_B corresponding to pH = 7.

The parameters for the GPC controller were chosen as $N = 50, N_u = 4, \lambda = 1, \alpha = 0.3$. Several model orders have been tried, the best results have been obtained with the third order model. The sampling time was set to 5s.

The tuning of the predictive algorithm was performed at pH = 9 with the requirement, that the deviations from the steady-state have to be within ±0.1pH. It was observed, that the small values of N_u led to rather active control actions and the final value $N_u = 4$ was chosen as the result of trade-off between the performance and the complexity of the calculations. The parameter λ influenced the penalisation of the future control increments. Very large values caused limit cycles of pH as the control was unable to compensate satisfactorily the disturbances, therefore the smallest possible value was chosen.

For the final tuning of the algorithm, the P polynomial was used. Only first order polynomial of the form $P(z^{-1}) = (1 - \alpha z^{-1})/(1 - \alpha)$ was assumed. The smaller values resulted in increased steady-state deviations and the larger values in very slow and oscillatory response to setpoint changes. The value $\alpha = 0.3$ was chosen as a compromise.

The experimental results of the adaptive GPC controller were compared with a carefully tuned PI controller [33]. All experiments were carried out with the same pattern of setpoint changes. At first the reactor was stabilised at pH = 7 and then controlled to pH = 9, 7, 8.3 (Fig. 10). Finally, the disturbance rejection performance has been studied. As a disturbance, a 20% decrease of the acid flow was performed at $t = 0$ and held constant afterwards (see Fig. 11).

The experiments have been confirmed, that the adaptive GPC method is able to control the strongly nonlinear plant and that it behaves much better compared to linear PI control. However the tuning of the controller parameters must have been done with some care and only the parameters which slowed down the closed-loop system substantially, gave good results. This is because a neutralisation reactor control is known to be not very succesful with linear controllers.

9 Conclusions

This report has dealt with Model Based Predictive Control. Its aim was to explain the principles of predictive control on one of the most used method - GPC. The derivation of the method was based on two process characteristics - step and free responses and avoided to solve recursively Diophantine equations. With this approach, non-linear models can easily be included. This, however, leads to control based on linearised models. If full non-linear

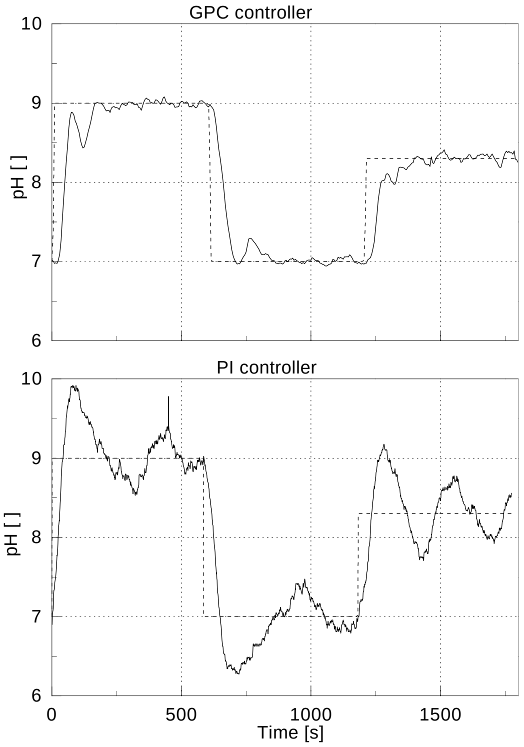

Figure 10: Setpoint tracking

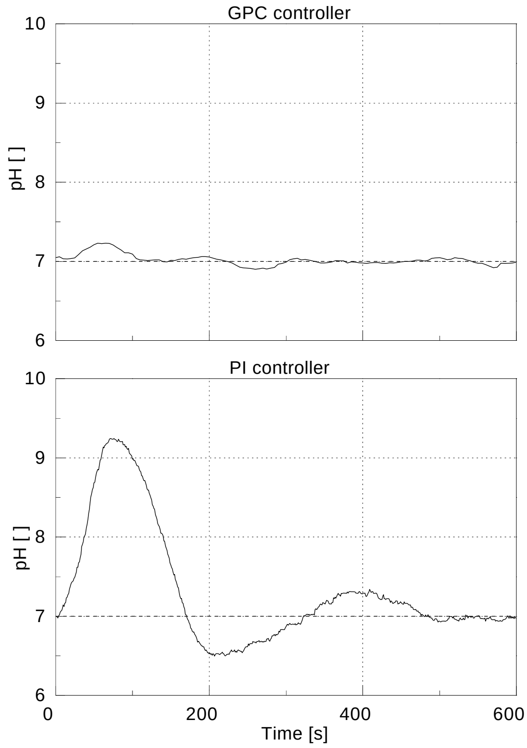

Figure 11: Disturbance rejection

models are to be used, analytical solution for future control increments is no longer possible to obtain.

Various other approaches have also been discussed - the methods that have been developed on the same principles as well as the methods that try to overcome the main drawback - stability problem. Within these, three mayor streams of research have been identified and shortly described.

The second part of the report has dealt with more practical issues. Some of the tuning strategies for GPC have been mentioned. The selected simulation and real-time examples were chosen to highlight some of the GPC features as well as some useful hints for implementation.

Acknowledgements

Financial support of this work from VEGA MSSR, (grants no. 95/5195/198 and 1/4200/97) is very gratefully acknowledged.

References

[1] R. R. Bitmead, M. Gevers, and V. Wertz. *Adaptive Optimal Control*. Prentice Hall, Englewood Cliffs, New Jersey, 1990. 4, 18

[2] S. Bittanti, P. Bolzern, and M. Campi. Exponential convergence of a modified directional forgetting identification algorithm. *Systems & Control Letters*, 14:131–137, 1990. 29

[3] G. Bortolotto and S. B. Jørgensen. Finite horizon multivariable optimal control of constrained chemical processes. In *Proc. 6th IFAC Workshop on Control Applications of Non-linear Programming and Optimization, London*, 1986. 6

[4] E. F. Camacho and C. Bordons. *Model Predictive Control in the Process Industry*. Springer-Verlag, London, 1995. 4

[5] P. J. Campo and M. Morari. Robust control of processes subject to saturation nonlinearities. *Computers chem. Engng.*, 14(4/5):343 – 358, 1990. 4, 17

[6] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control - part II. Extensions and interpretations. *Automatica*, 23(2):149 – 160, 1987. 4, 18

[7] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control - part I. The basic algorithm. *Automatica*, 23(2):137 – 148, 1987. 4

[8] D. W. Clarke and R. Scattolini. Constrained receding-horizon predictive control. *IEE Proc. D*, 138(4):347 – 354, 1991. 4, 19

[9] D. W. Clarke, editor. *Advances in Model-Based Predictive Control*. Oxford University Press, 1994. 4

[10] C. R. Cutler and B. L. Ramaker. Dynamic Matrix Control - A computer control algorithm. In *Proc. of JACC, San Francisco*, 1980. 4

[11] H. Demircioğlu and P. J. Gawthrop. Continuous-time GPC (CGPC). *Automatica*, 27(1):55 – 74, 1991. 16

[12] G. De Nicolao and S. Strada. On the stability of receding-horizon LQ control with zero-state terminal constraint. *IEEE Trans. Automatic Control*, 42(2):257 – 260, 1997. 4

[13] M. Fikar, P. Dostál, and J. Mikleš. Adaptive predictive control of tubular chemical reactor. *Petroleum and Coal*, 38(3):51 – 57, 1996. 27

[14] M. Fikar and A. Draeger. Adaptive predictive control of a neutralization reactor. In *Preprints of 10th Conf. Process Control'95, June 4 – 7, Tatranské Matliare, Slovakia*, volume 1, pages 153 – 157, 1995. 32

[15] M. Fikar, S. Engell, and P. Dostál. Design of infinite horizon predictive LQ controller. In *CD-ROM Proceedings of ECC'97, Bruxelles, Paper No. 698*, 1997. 4, 22

[16] M. Fikar, S. Engell, and P. Dostál. Design of predictive LQ controller. *Kybernetika*, 35(4):459–472, 1999. 4, 22

[17] M. Fikar and S. Engell. Receding horizon predictive control based upon Youla-Kučera parametrization. *European Journal of Control*, 3(4):304 – 316, 1997. 4, 21

[18] C. E. García and A. M. Morshedi. Quadratic programming solution of dynamic matrix control (QDMC). *Chem. Engng. Commun.*, 46:73 – 87, 1986. 4

[19] C. E. García, D. M. Prett, and M. Morari. Model predictive control: Theory and practice - a survey. *Automatica*, 25(3):335 – 348, 1989. 4

[20] J. R. Gossner, B. Kouvaritakis, and J. A. Rossiter. Stable generalized predictive control with constraints and bounded disturbances. *Automatica*, 33(4):551 – 568, 1997. 4

[21] T. Kailaith. *Linear Systems*. Prentice Hall, Englewood Cliffs, New Jersey, 1980. 20

[22] S. S. Keerthi and E. G. Gilbert. Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: stability and moving horizon approximations. *Journal of Optimization Theory and Application*, 57:265 – 293, 1988. 4, 18

[23] M. Kinnaert. Adaptive generalized predictive controller for MIMO systems. *Int. J. Control*, 50(1):161 – 172, 1989. 4

[24] B. Kouvaritakis, J. A. Rossiter, and A. O. T. Chang. Stable generalised predictive control: an algorithm with guaranteed stability. *IEE Proc. D*, 139(4):349 – 362, 1992. 4, 20

[25] B. Kouvaritakis and J. A. Rossiter. Multivariable stable generalised predictive control. *IEE Proc. D*, 140(5):364 – 372, 1993. 4

[26] R. Kulhavý and M. Kárný. Tracking of slowly varying parameters by directional forgetting. In *Proc. 9th IFAC World Congr., Budapest*, pages 79–83, 1984. 29, 31

[27] W. H. Kwon and A. E. Pearson. On feedback stabilization of time-varying discrete linear systems. *IEEE Tr. Aut. Control*, 23(3):479 – 481, 1978. 4, 18

[28] J. H. Lee, M. Morari, and C. E. García. State-space interpretation of model predictive control. *Automatica*, 30(4):707 – 717, 1994. 4

[29] J.-W. Lee, W. H. Kwon, and J. Choi. On stability of constrained receding horizon control with finite terminal weighting matrix. In *CD-ROM Proceedings of ECC'97, Bruxelles, Paper No. 93*, 1997. 4, 22

[30] D. Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *IEEE Tr. Aut. Control*, 35:814 – 824, 1990. 4

[31] A. R. McIntosh, S. L. Shah, and D. G. Fisher. Analysis and tuning of adaptive generalized predictive control. *Can. J. Chem. Eng.*, 69:97 – 110, 1991. 4, 22

[32] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. B. Rawlings. Receding horizon control and discontinuous state feedback stabilization. *Int. Journal of Control*, 62(5):1217 – 1229, 1995. 4

[33] U. Meier. Mathematische Modellierung und Entwurf eines nichtlinearen Reglers für einen kontinuierlich betriebenen Neutralisationsreaktor. Master's thesis, Universität Dortmund, Fachbereich Chemietechnik, 1993. 33

[34] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Tr. Aut. Control*, 38:1623 – 1633, 1993. 4

[35] E. Mosca and J. Zhang. Stable redesign of predictive control. *Automatica*, 28:1229 – 1233, 1992. 4, 19

[36] K. R. Muske and J. B. Rawlings. Linear model predictive control of unstable processes. *Journal of Process Control*, 3(2):85 – 96, 1993. 4

[37] K. R. Muske and J. B. Rawlings. Model predictive control with linear models. *AIChE Journal*, 39(2):262 – 287, 1993. 4

[38] D. M. Prett and C. E. García. *Fundamental Process Control*. Butterworths, 1988. 4

[39] J. B. Rawlings and K. R. Muske. The stability of constrained receding horizon control. *IEEE Trans. Automatic Control*, 38(10):1512 – 1516, 1993. 4, 21, 22

[40] J. Richalet, A. Rault, J. L. Testud, and J. Papon. Model predictive heuristic control application to industrial processes. *Automatica*, 14(5):413 – 428, 1978. 4

[41] B. Rohaľ-Ilkiv and B. Benčurik. Practical experiments with heat exchanger predictive control. In *Proc. 3rd Int. Conf. New Trends in Automation of of Energetic Processes*. Academia Centrum, Zlín, Czech Republic, 1998. 16

[42] B. Rohaľ-Ilkiv. One approach to continuous-time predictive control. In Š. Kozák and M. Huba, editors, *Preprints 2nd IFAC Workshop New Trends in Desing of Control Systems*, pages 96 – 103. Smolenice, Slovakia, 1997. 16

[43] J. A. Rossiter, J. R. Gossner, and B. Kouvaritakis. Infinite horizon stable predictive control. *IEEE Trans. Automatic Control*, 41(10):1522 – 1527, 1996. 4, 22

[44] J. A. Rossiter, B. Kouvaritakis, and J. R. Gossner. Feasibility and stability results for constrained stable predictive control. *Automatica*, 31(3):863 – 877, 1995. 4

[45] J. A. Rossiter and B. Kouvaritakis. Constrained stable generalised predictive control. *IEE Proc. D*, 140(4):243 – 254, 1993. 4

[46] A. Rusnák, K. Fikar, M. Najim, and A. Mészaros. Generalized predictive control based on neural networks. *Neural Processing Letters*, 4(2):107 – 112, 1996. 30

[47] P. O. M. Scokaert and D. W. Clarke. Stabilizing properties of constrained predictive control. *IEE Proc.-Control Theory Appl.*, 141(5):295 – 304, 1994. 4

[48] P. O. M. Scokaert, J. B. Rawlings, and E. S. Meadows. Discrete-time stability with perturbations: Application to model predictive control. *Automatica*, 33(3):463 – 470, 1997. 4

[49] R. Shridhar and D. J. Cooper. A tuning strategy for unconstrained SISO model predictive control. *Ind. Eng. Chem. Res.*, 36:729 – 746, 1997. 4, 22, 23

[50] R. Shridhar and D. J. Cooper. A tuning strategy for unconstrained multivariable model predictive control. *Ind. Eng. Chem. Res.*, 37:4003 – 4016, 1998. 4, 22, 23

[51] R. Soeterboek. *Predictive Control: A Unified Approach*. PhD thesis, Dep. of Electrical Engineering, Delft University of Technology, Netherlands, 1992. 4, 6

[52] E. Zafiriou and H. W. Chiou. On the dynamic resiliency of constrained processes. *Computers chem. Engng.*, 20(4):347 – 355, 1996. 4

[53] E. Zafiriou and A. L. Marchal. Stability of SISO Quadratic Dynamic Matrix Control with hard output constraints. *AIChE Journal*, 37(10):1550 – 1560, 1991. 4

[54] E. Zafiriou. Robust model predictive control of processes with hard constraints. *Computers chem. Engng.*, 14:358 – 371, 1990. 4

[55] A. Zheng and M. Morari. Global stabilization of linear discrete-time systems with bounded controls. A model predictive control approach. In *Proc. of ACC*, volume 3, pages 2847 – 2851. Baltimore, Maryland, 1994. 4

[56] A. Zheng and M. Morari. Stability of model predictive control with mixed constraints. *IEEE Trans. Automatic Control*, 40(10):1818 – 1823, 1995. 4