Practical Implementation of Advanced Process Control for Linear Processes

Jørgen K . H. Knudsen *, **Jakob Kjøbsted Huusom** **
John Bagterp Jørgensen ***

* *2-control Aps, Frimodtsvej 11, DK-2900 Hellerup, Denmark (e-mail: JoeK@2-control.dk)*
** *CAPEC, Department of Chemical and Biochemical Engineering, Technical University of Denmark, DK-2800 Lyngby (e-mail: jkh@kt.dtu.dk)*
*** *DTU Compute, Technical University of Denmark, DK-2800 Lyngby (e-mail: jbjo@dtu.dk)*

Abstract: Most advanced process control systems are based on Model Predictive Control (MPC). In this paper we discuss three critical issues for the practical implementation of linear MPC for process control applications. The first issue is related to offset free control and disturbance models; the second issue is related to the use of soft output constraints in MPC; and the third issue is related to the computationally efficient solution of the quadratic program in the dynamic regulator of the MPC. We have implemented MPC in .Net using C# and the MPCMath library. The implemented MPC is based on the target-regulator structure. It enables offset free control; it can be computed efficiently on-line using several optimization algorithms; and accommodates soft constraint for the outputs and for shaping the set-point tracking penalty function.

We report selected observations using this implementation and discuss their practical implications for process control. If the control and evaluation intervals are chosen too short, the predicted behaviour of the controllers may have unstable characteristics. Depending of the degrees of freedom, offset-free control of a number of the controlled variables can be achieved by introduction of noise models and integration of the innovation errors. If the disturbances increases, offset-free control cannot be achieved without violation of process constraints. A target calculation function is used to calculate the optimal achievable target for the process. The use of soft constraints for process output constraints in the MPC controllers, ensures feasible solutions. The computational load as function of controllers type, model dimension and constraint type are shown.

Keywords: Advanced Process Control, Linear Quadratic Regulators, Model Predictive Control, Disturbance Modelling, Offset Free control, LQR, MPC, Riccati, C#, .NET, MPCMath

1. INTRODUCTION

This paper describes some of the practical issues encountered when implementing a LQR or MPC controller for a linear process. During the initial phase of implementation of an Advanced Process Control system, the attention is often focused around the important task of developing a suitable model for the process. Having the model, a number of new practical issues arises. Which controller algorithm should be applied? How long control and evaluation horizons are required? How are stationary offsets due to set-point changes, unmeasured disturbances and model errors treated? What are CPU requirements for the MPC controller?

The LQR and MPC algorithms used in this work are widely described in literature. Rao et al. (1998) describes application of interior-point method to MPC and uses a discrete-time Riccati recursion to solve the linear equations efficiently. Jørgensen (2004) has a systematic treatment of numerical methods for MPC, and recently Frison (2012) made a systematic analysis of numerical methods for MPC comparing condensed and Riccati based MPC algorithms.

Muske and Rawlings (1993) present the framework for MPC based on state-space models, and integral control schemes designed to remove steady-state offsets. Muske and Badgwell (2002) and Pannochia and Rawlings (2003) presents generalized disturbance models for unmeasured disturbances and analyses the conditions required for detectability and the necessary requirements for achieving offset-free control. Rajamani et al. (2009) show that the disturbance model does not affect the closed-loop performance if appropriate covariances are used in specifying the state estimator. Huusom et al. (2012) shows how to apply these methods on plants described by ARX models. Olesen et al. (2012) tuning procedures for offset-free MPC controllers.

The Four Tank Process introduced by Johanson (2000), is used for simulated examples.

The LQR and MPC controllers are implemented in C# using the MPCMath library suitable for inclusion in industrial DCS systems. (Knudsen, 2010a,b)

The paper is organized as follows. Section 2 describes the control problem and gives an introduction to the algorithms used to solve the problem. Section 3 introduces the Four Tank Process used as a simulation example for the controllers. Section 4 discusses the terminal effect from finite control and prediction horizons. Section 5 handles noise models and offset-free control. Section 6 introduces the test scenario with simulated measurement and process noise. Section 7 reports the achieved CPU usage with the applied control algorithms. Conclusions are presented in section 8.

2. THE CONTROL PROBLEM

The LQR and MPC controllers have the objective:

$$\min_{\{y,\Delta u\}} \Phi = \sum_{k=0}^{eh} \frac{1}{2}(y_k - r_k)'\theta_k(y_k - r_k) + \sum_{k=0}^{ch-1} \frac{1}{2}\Delta u_k' \rho_k \Delta u_k \quad (1)$$

Subject to process dynamics equality constraints:

$$x_{k+1} = Ax_k + Bu_k \qquad k = 0, ..., eh-1 \quad (2)$$

$$y_k = Cx_k \qquad k = 0, ..., eh \quad (3)$$

where k is the time or controller step, ch is the control horizon and eh is the evaluation horizon ($eh \geq ch$). y_k are the measured plant outputs, r_k the references and Δu_k are the movements of the process inputs, defined as $\Delta u_k = u_k - u_{k-1}$.

In-equality constraints for the MPC controllers will introduced in section 2.3.

2.1 Condensed controllers

Many MPC implementations eliminate the state variables x_k from the control problem using a finite impulse response, FIR, model instead of the state-space model (Prasath et al., 2010) :

$$y_k = f_k + \sum_{j=1}^{i} H_j u_{k-j} \quad (4)$$

where $f_k = CA^k X_0$ is the free response to the initial state-space condition X_0 . The Markov parameters H_k are defined by $H_k = CA^{k-1}B$

The FIR model requires the process to be stable. Processes with unstable poles or pure integrators must be stabilized by a simple proportional controller in order to apply the FIR model approach (Maciejowski, 2002).

Defining the vectors Y, R, F and U as

$$Y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{eh} \end{bmatrix} R = \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{eh} \end{bmatrix} F = \begin{bmatrix} f_0 \\ f_2 \\ \vdots \\ f_{eh} \end{bmatrix} U = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{ch-1} \end{bmatrix} \quad (5)$$

and the matrix Γ

$$\Gamma = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ H_1 & 0 & \dots & 0 & 0 \\ H_2 & H_1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ H_{eh-1} & H_{eh-2} & & H_1 & 0 \\ H_{eh} & H_{eh-1} & \dots & H_2 & \sum_{j=1}^{eh-ch} H_j \end{bmatrix} \quad (6)$$

The prediction from (4) can be expressed as

$$Y = F + \Gamma U \quad (7)$$

Define the matrix Λ and vector I_0 by

$$\Lambda = \begin{bmatrix} I & 0 & 0 & \dots & 0 & 0 \\ -I & I & 0 & \dots & 0 & 0 \\ 0 & -I & I & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & I & 0 \\ 0 & 0 & 0 & \dots & -I & I \end{bmatrix} I_0 = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

define Q_θ and Q_ρ

$$Q_\theta = \begin{bmatrix} \theta_0 & & & \\ & \theta_1 & & \\ & & \ddots & \\ & & & \theta_{eh} \end{bmatrix} Q_\rho = \begin{bmatrix} \rho_0 & & & \\ & \rho_1 & & \\ & & \ddots & \\ & & & \rho_{ch-1} \end{bmatrix} \quad (9)$$

then the control problems ends in a QP problem:

$$\min_{\{U\}} \Phi = \frac{1}{2}U'QU + q'U \quad (10)$$

where Q and q are defined by

$$Q = \Gamma' Q_\theta \Gamma + \Lambda' Q_\rho \Lambda \quad (11)$$

$$q = \Gamma' Q_\theta (F - R) - \Lambda' Q_\rho I_0 u_{-1} \quad (12)$$

There are no equality constraints included. The matrix Q is dense. The computational effort in solving the QP (1) is proportional to $(ch\ n_u)^3$, as the main operation is the factorization of the Hessian matrix, Q.

If the process moves to a new operation region, the matrix of Markow parameters Γ (6) has to be recalculated.

2.2 Riccati based controllers

In the Riccati based LQR and MPC algorithms, the process dynamics (2) are kept as equality constraints.

There a many advantages in the optimization by augmenting the state space space model (2) to:

$$\begin{bmatrix} x_{k+1} \\ u_k \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x_k \\ u_{k-1} \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u_k \quad (13)$$

$$\tilde{x}_{k+1} = \tilde{A}\tilde{x}_k + \tilde{B}\Delta u_k \quad (14)$$

The QP problem becomes

$$\min_{\{W\}} \Phi = \frac{1}{2}W'QW + q'W \quad (15)$$

subject to:

$$AW = b \quad (16)$$

with:

$$W' = [\tilde{x}_0'\ \Delta U_0'\ \tilde{x}_1'\ \Delta U_1'\ \dots\ \tilde{x}_{N-1}'\ \Delta U_{N-1}'\ \tilde{x}_N'] \quad (17)$$

$$Q = \begin{bmatrix} C'\theta_0 C & 0 & \dots & 0 \\ 0 & C'\theta_1 C & & \vdots \\ \vdots & & \ddots & \\ 0 & 0 & \dots & C'\theta_N C \end{bmatrix} \quad q = \begin{bmatrix} -r_0\theta_0 C \; 0 \\ -r_1\theta_1 C \; 0 \\ \vdots \\ -r_N\theta_N C \; 0 \end{bmatrix} \tag{18}$$

$$A = \begin{bmatrix} I & 0 & 0 & & & & \\ -\tilde{A} & -\tilde{B} & I & & & & \\ 0 & 0 & -\tilde{A} & -\tilde{B} & I & & \\ \vdots & & & & \ddots & I & \\ 0 & 0 & 0 & & \dots & -\tilde{A} & -\tilde{B} \; I \end{bmatrix} \quad b = \begin{bmatrix} X_0 \\ b \\ b \\ \vdots \\ b \end{bmatrix} \tag{19}$$

The dimension of the QP problem has increased, but the equations can be solved using a Riccatti sequence.

The computational effort is $\approx ch\ (n_{\tilde{x}}+n_u)^3$. The computational effort is linearly proportional to the control horizon, which is important when the control horizon is increased.

Including the process dynamic as equality constraints (19) makes it straightforward to exchange the model during controller executions if the plant moves to another operational region. The real problem is to find the all the required models for all the plants operational regions.

2.3 Process constraints for MPC controllers

The process constraints are:

$$\begin{aligned} u_{\min} \le u_k \le u_{max} & \qquad k = 0, \dots, N-1 \\ \Delta u_{\min} \le \Delta u_k \le \Delta u_{\max} & \qquad k = 0, \dots, N-1 \\ y_{\min} + \eta_{Lk} \le y_k \le y_{\max} - \eta_{Lk} & \qquad k = 0, \dots, N \\ dy_{\min} + \eta_{Dk} \le y_k - r_k \le dy_{\max} - \eta_{Dk} & \qquad k = 0, \dots, N \\ 0 \le \eta_{Lk} & \qquad k = 0, \dots, N \\ 0 \le \eta_{Dk} & \qquad k = 0, \dots, N \end{aligned}$$

The soft constraint on $y_{\min} \le y_k \le y_{\max}$ ensures that a feasible solution to the QP problems can be found. The soft constraint on deviation from the reference $dy_{\min} \le y_k \le dy_{\max}$ can be used to minimize unnecessary control actions due to measurement noise.

The two soft constraints requires an expansion of objective (1) to:

$$\min_{\{y,\eta_L,\eta_D,\Delta u\}} \Phi = \sum_{k=0}^{eh} \frac{1}{2}(y_k - r_k)'\theta_k(y_k - r_k)$$
$$+\sum_{k=0}^{ch-1} \frac{1}{2}\Delta u_k'\rho_k\Delta u_k + \sum_{k=0}^{eh} \frac{1}{2}\eta_{Lk}'\mu_{Lk}\eta_{Lk} + \sum_{k=0}^{eh} \frac{1}{2}\eta_{Dk}'\mu_{Dk}\eta_{Dk} \tag{20}$$

3. THE FOUR TANK PROCESS EXAMPLE

The four tank process is used to illustrate controller actions. The four tank system, illustrated in Fig. 1, was introduced by Johanson (2000) as a benchmark for control design.

The process outputs are the water levels in the four tanks, $H1$, $H2$, $H3$ and $H4$. The process inputs are the two inflows, $F1$ and $F2$.

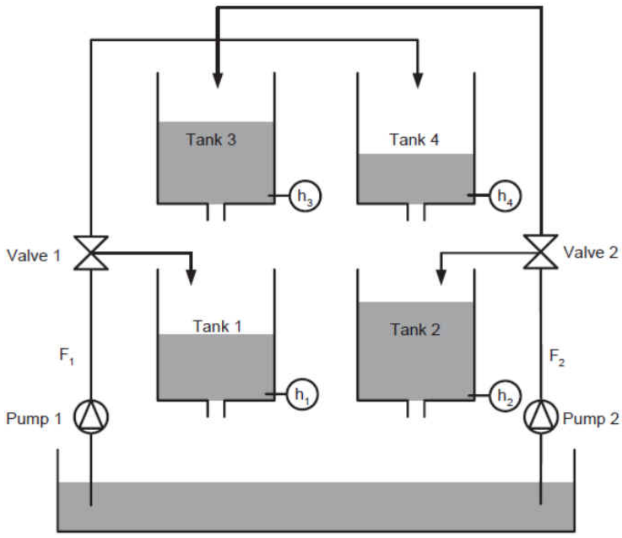

Fig. 1. Four tank process used to illustrate controller actions

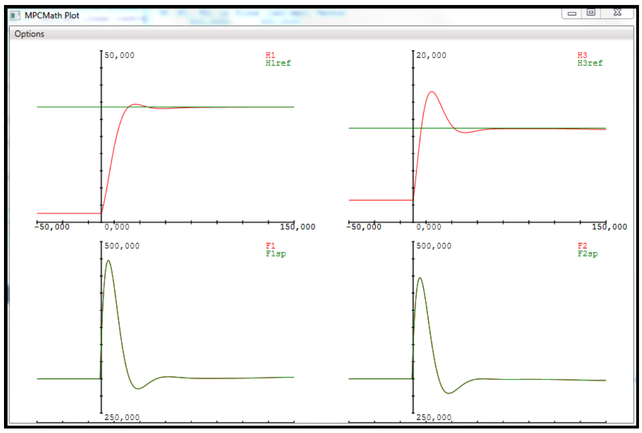

Fig. 2. LQR controller, $horizon = 150$, $\theta = (100, 100, 1, 1)$, $\rho = (10, 10)$

Outputs	y_{min}	y_s	y_{max}	
$H1$	0.0	33.58	50.0	Left lower tank level
$H2$	0.0	27.49	50.0	Right lower tank level
$H3$	0.0	10.97	50.0	Left upper tank level
$H4$	0.0	9.21	50.0	Right upper tank level
Inputs	u_{min}	u_s	u_{max}	
$F1$	0.0	300.0	450.0	Left pump
$H2$	0.0	300.0	450.0	Right pump

Fig. 2 shows the performance of a LQR controller where the input movement penalties ρ are tuned to keep the flows $F1$ and $F2$ below their high limits. Fig. 3 shows the performance of a MPC controller with less penalty on the inputs movements. The MPC keeps $F1$ and $F2$ below their high limits.

4. TERMINAL EFFECT FROM FINITE CONTROL AND PREDICTION HORIZONS

The prediction part of the typical operator pictures are very useful for the operator to evaluate the performance of the controller. In some cases the controller shows an unstable behaviour at the end of the evaluation horizon,

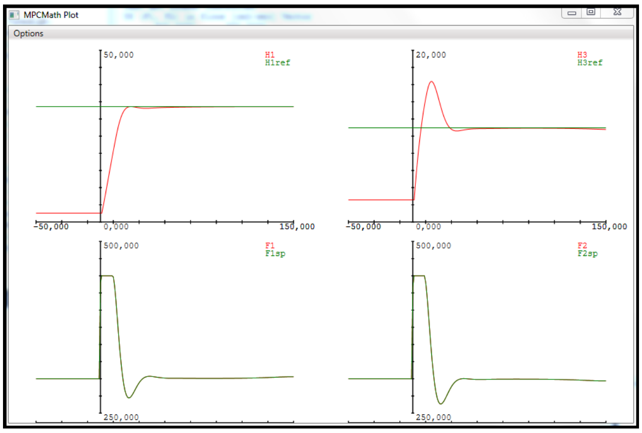

Fig. 3. MPC controller. *horizon* = 150, $\theta = (100, 100, 1, 1)$, $\rho = (1, 1)$

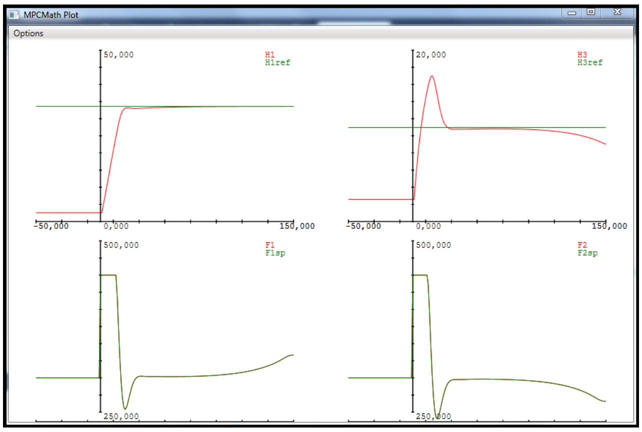

Fig. 4. Terminal effect from finite control and prediction horizons. *horizon* = 150, $\theta = (100, 100, 1, 1)$, $\rho = (0.1, 0.1)$

which might be optimal, but will not be accepted as such by the operators. Fig. 4 shows an example of such a behaviour obtained by decreasing penalty on the input movement further. This problem can be removed by applying a increased control and evaluation horizons. Increasing the evaluation horizon to 300 removes the deficiency. Riccati based controllers can be implemented with increased evaluation horizon without increasing the CPU load, by calculation of a suitable penalty at the end of the control horizon.

5. OFFSET-FREE OPERATION

The LQR and MPC controllers above are unable to remove offsets caused by set-point changes, unmeasured disturbances and models errors. The offsets are demonstrated in Fig. 5 and Fig. 6

Something like the integral action in the PID controllers is needed. The solution is to introduce noise-models, integrate the estimated errors and calculate best possible target, depending on the estimated disturbances. An overview of these methods is given by Jørgensen (2004).

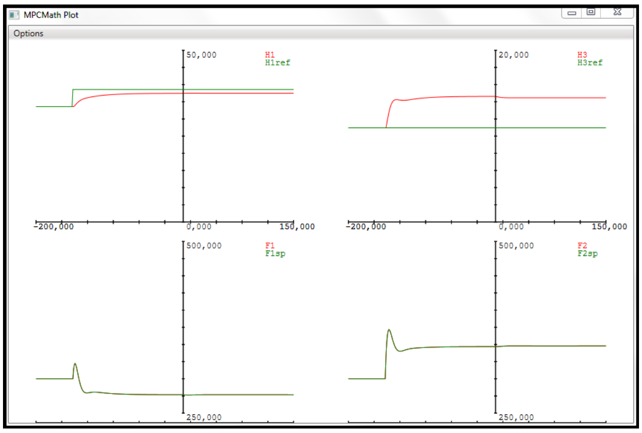

Fig. 5. Increasing the reference for $H1$ by 5.0 cm results in a stationary offset in $H1$

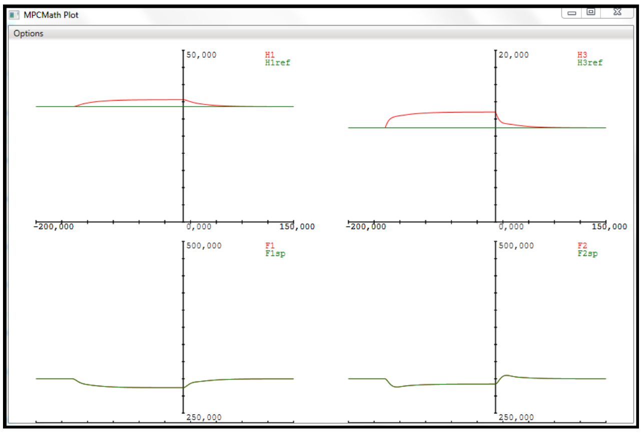

Fig. 6. An unmeasured disturbance is simulated by adding extra water into tank $H3$. A stationary offset is achieved, but the controller thinks it can remedy the offset in the future.

5.1 Noise Models

The plant model (2) is expanded with a vector of unmeasured disturbances d_k.

$$x_{k+1} = Ax_k + Bu_k + B_d d_k \tag{21}$$
$$y_k = Cx_k + C_d d_k \tag{22}$$

The disturbances in d_k are assumed to be constant, only changing value now and then. The prediction equations are:

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k + B_d\hat{d}_{k|k} \tag{23}$$
$$\hat{d}_{k+1|k} = \hat{d}_{k|k} \tag{24}$$

The innovation

$$\hat{\epsilon} = y_k - \hat{y}_k = y_k - C\hat{x}_{k|k-1} - C_d\hat{d}_{k|k-1} \tag{25}$$

end the Kalman filtering equations

$$\hat{x}_{k|k-1} = \hat{x}_{k|k} + L_x\hat{\epsilon}_k \tag{26}$$
$$\hat{d}_{k|k-1} = \hat{d}_{k|k-1} + L_d\hat{\epsilon}_k \tag{27}$$

If the noise characteristics are known, L_x and L_d can be calculated as the stationary Kalman filter gains. A pragmatic solution is to specify L_d as diagonal matrix, with integrations factors in the range $0.0 \geq I_{fac} \geq 1.0$. The different noise models used in practice are described by (21) by substituting B_d and C_d with:

Noise model	B_d	C_d
Input noise	B	0
Output noise	0	I
ARX process	K	I

where the ARX process is described by the model:

$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k + K\hat{\epsilon}_{k|k} \tag{28}$$

$$\hat{y}_{k|k} = CX_{k|k} + \hat{\epsilon}_{k|k} \tag{29}$$

The output noise model cannot be applied for plants with pure integrators because the resulting model is not detectable.

5.2 *Target calculation*

Having estimated the unmeasured disturbance $\hat{d}$, the optimal achievable target for the controller can be calculated solving the small quadratic problem:

$$\min_{\{x_{tg},u_{tg}\}} (y_{tg} - y_{sp})' Q_s (y_{tg} - y_{sp}) + (u_{tg} - u_{sp})' R_s (u_{tg} - u_{sp}) \tag{30}$$

subject to

$$(I - A)x_{tg} - Bu_{tg} = B_d\hat{d}_{k|k} \tag{31}$$

combined with the limits on the process inputs and outputs (20). Offset-free operation can be insured by including the set-points in the equality constraints (31), but this approach has the disadvantage that the these constraint must be removed again if the QP problem becomes infeasible due to large disturbances. Achieving offset-free operation by selection of high values in Q_s in (30) eliminates this problem.

Offset-free operation off course has to obey the rules given by the degrees of freedom. With two process inputs its only possible to achieve offset-free operation for two of the controlled variables for the Four tank Process, i.e. $H1$ and $H2$ in this case .

5.3 *The APC structure*

Fig. 7 illustrates the structure of the APC. The estimator block estimates the states including the disturbances from the measurements and the history summarized by the previous state estimate. The target calculation uses the estimated disturbances and the set points to compute steady target states and target inputs for the regulator. The regulator computes an optimal input sequence that will drive the process towards the steady target states and target inputs.

The offset-free handling of a $H1$ set-point change and $H3$ disturbance is shown in Fig. 8 and 9. If the disturbances increases, the offset-free control cannot be achieved as shown in Fig. 10.

6. CONTROL WITH MEASUREMENT AND PROCESS NOISE

The scenario for testing the performance of the controllers are shown in Fig. 11 without noise. The initial condition are nearly empty tanks. A disturbance in $H3$ is introduced from step 100 to 200. Fig. 12 shown the final scenario with added measurement and process noise.

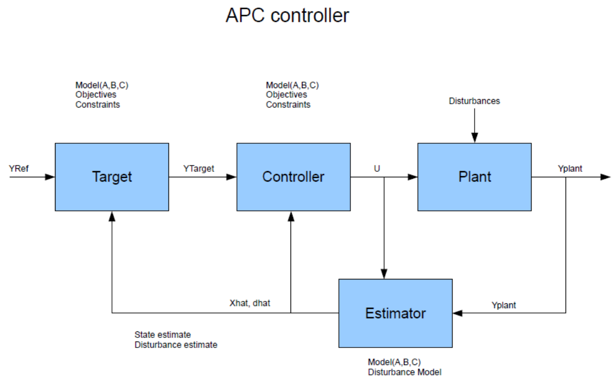

Fig. 7. APC structure

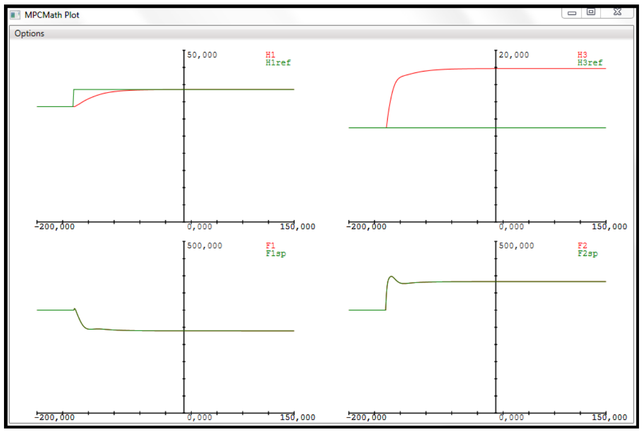

Fig. 8. Offset-free control of a $H1$ set-point change

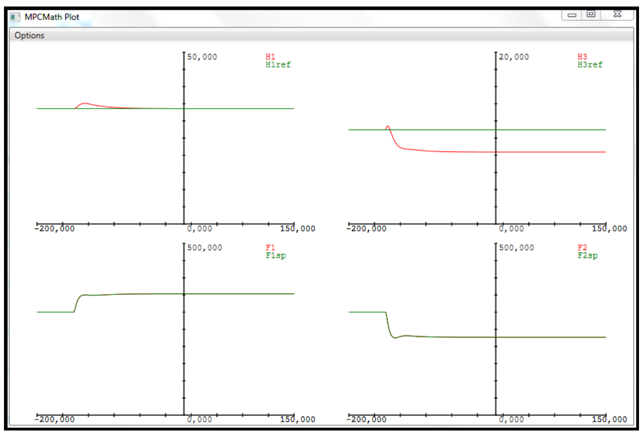

Fig. 9. Offset-free $H3$ disturbance handling

The MPC controller in Fig. 12 runs with control and evaluation horizons = 150, $\theta = (100, 100, 1, 1)$ and $\rho = (1, 1)$. With this tuning the controller compensates the process noise too aggressively. The soft constraint on the deviation between measurement and reference can be used to ignore the measurement noise as shown on Fig. 13. This MPC controller is tuned with $\theta = (1, 1, 0.01, 0.01)$, $\rho = (1, 1)$ and $\eta_D = (100, 100, 1, 1)$, $dy_{min} = (-5, -5, -5, -5)$ and $dy_{max} = (5, 5, 5, 5)$. This controller preserves the ability to get a fast compensation to the initial condition and does not try to compensate for the measurement noise.

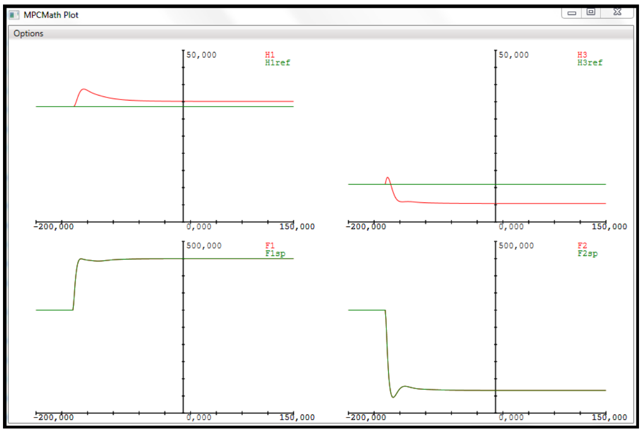

Fig. 10. Large disturbance in $H3$ forces a relaxation of offset-free operation

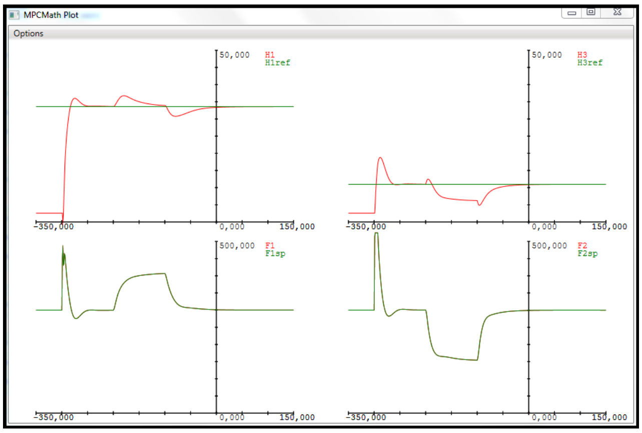

Fig. 11. Test case before adding measurement and process noise

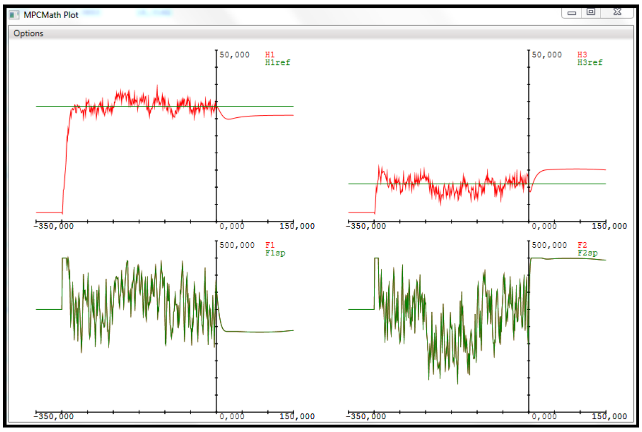

Fig. 12. MPC controller performance with measurement and process noise

7. CPU REQUIREMENTS

In MPC a QP for the regulator must be solved on line at each sample time. More computational power has increased our ability to solve larger problems. However, an even larger speed-up can be gained by choosing the correct numerical algorithm for a specific problem. Using the test scenario in Fig. 12, the results in Table 1 and 2 illustrate the speed-up by using the Riccati based factorization in a structured optimization problem compared to dense linear algebra in a QP arising from state elimination.

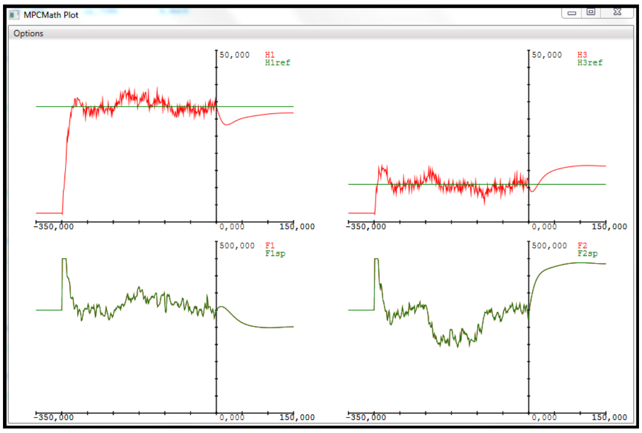

Fig. 13. MPC controller performance with soft constraint on deviations from references

ch	eh	LQR	$RiccatiMPC$	$CondensedMPC$
40	40	1	146	228
40	80	2	121	324
40	150	2	137	587
40	200	3	141	757
40	300	4	157	1107
80	80	2	267	636
80	150	3	252	1030
80	200	3	260	1387
80	300	4	267	2152
150	150	4	467	2033
150	200	4	482	2637
150	300	5	498	4035
200	200	4	605	3767
200	300	5	623	5427
300	300	6	924	9695

Table 1. CPU time in ms per control step

The required CPU time per control step is shown in Table. 1 for LQR, Riccati based MPC and Condensed MPC without target calculation and Soft constraints. The LQR controllers are much less CPU demanding than the MPC controllers. In all the test cases the condensed MPC algorithm is slower than the Riccati based MPC. The table also shows how the Riccati MPC's CPU demand increases linearly with the control horizon, whereas the condensed MPC's CPU demand has the cubic dependence of the control horizon. Finally Table 1 shows how an increase in the evaluation horizon can be implemented without increased CPU demand for the Ricccati based LQR and MPC. For all test cases the interior-point based QP algorithm required approximately 10 iterations per control step.

Adding the Target calculation block increased the the CPU load with 2-3 ms seconds per control step. Including soft limit on deviation from the set-point and the target calculation block gives the CPU usage shown in Table. 2.

The condensed MPC is extremely CPU demanding when the control and evaluation horizon increases, but this algorithm could still be interesting for applications with many states and few process inputs and outputs as shown by Frison (2012).

ch	eh	LQR	$RiccatiMPC$	$CondensedMPC$
40	40	-	218	414
40	80	-	206	733
40	150	-	230	1255
40	200	-	210	1688
40	300	-	210	2493
80	80	-	392	1330
80	150	-	432	2381
80	200	-	405	3183
80	300	-	398	4935
150	150	-	752	4644
150	200	-	718	6156
150	300	-	728	9363
200	200	-	960	8636
200	300	-	956	12802
300	300	-	1494	22098

Table 2. CPU time in ms per control step with target calculations and soft constraints

The CPU times are only indicative. They were obtained using and Intel Core I5 2.5 MHz CPU. The algorithms are written in C# using the MPCMath library (Knudsen, 2010a,b).

8. CONCLUSIONS

The Riccati based LQR algorithm is extremely CPU efficient, but lacks treatment of process limitations.

The condensed MPC and Riccati based MPC algorithms provides the same control actions. The Ricatti based MPC algorithm is much more CPU efficient than the Condensed MPC algorithm, especially for longer prediction horizons. The condensed MPC algorithm might be the optimal choice for processes with many states and few inputs and outputs.

Terminal effect from finite control and prediction horizons can be remedied by increasing the control and evaluation horizon. Preserving the control horizon and increasing the evaluation horizon can achieve the same stabilization. For Riccati based LQR and MPC algorithms this can be done without increasing the CPU demand.

Soft limits on process output limits, guarantees feasible solutions to the QP problem. Soft limits on deviation between plant output and set-points can be used to dampen control actions with sacrificing the ability to handle real disturbances.

Offset-free control can be achieved by augmenting the process model with constant step disturbances. These disturbances are estimated using Kalman type filers and the optimal achieve-able targets are calculated. Including offset-free control only increases CPU load marginally.

REFERENCES

Frison, G. (2012). *Numerical Methods for Model Predictive Control.* Master's thesis, IMM, Technical University of Denmark.

Huusom, J.K., Poulsen, N.K., Jørgensen, S.B., Jørgensen, and Jørgensen, J.B. (2012). Tuning siso offset-free model predictive control based on arx models. *Journal of Process Control*, 22, 1997–2007.

Johanson, K.H. (2000). The quadruple-tank process: A multivariable laboratory process with an adjustable zero. *IEEE Transactions on control systems technology*, 8(3), 456–465.

Jørgensen, J. (2004). *Moving Horizon Estimation and Control.* Ph.D. thesis, Department of Chemical Enineering, Technical University of Denmark.

Knudsen, J.K.H. (2010a). Implementing model predictive control in the csharp/.net environment. In *Model Based Control Conference. DTU.* "`http://www.2-control.dk`".

Knudsen, J.K.H. (2010b). Introduction to mpcmath. "`http://www.2-control.dk`".

Maciejowski, J. (2002). *Predictive Control with constraints.* Prentice Hall.

Muske, R.J. and Badgwell, T.A. (2002). Disturbance modeling for offset-free linear model predictive control. *Journal of Process Control*, 12, 617–632.

Muske, R.J. and Rawlings, J.B. (1993). Model predictive control with linear models. *AICHe Journal*, 39, 262–287.

Olesen, D.H., Huusom, J.K.., and Jørgensen, J.B. (2012). A tuning procedure for arx-based mpc of multivariate processes. Submitted to 2013 American Control Conference.

Pannochia, G. and Rawlings, J. (2003). Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49, no 2, 426–437.

Prasath, G., Recke, B., Chidambaram, M., and Jørgensen, J. (2010). Application of soft constrained mpc to a cement mill circuit. In *9th International Symposium on Dynamics and Control of Process Systems, pages: 288-293.*

Rajamani, M.R., Rawlings, J.B., and Qin, S.J. (2009). Achieving state estimation equivalence for misaligned distrubances in offset-free model predictive control. *AIChE Journal*, 55, no 2, 396–407.

Rao, C.V.., Wright, S.J., and Rawlings, J.B. (1998). Application of interior-point methods to model predictive control. *JOTA*, 3, 723–757.