Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Real-time Implementation of Fault-tolerant Control Using Model Predictive Control

T. Miksch A. Gambier E. Badreddin

Automation Laboratory, University of Heidelberg, B6, 26, 68131 Mannheim, Germany e-mail: miksch@ti.uni-mannheim.de, gambier@ieee.org, badreddin@ti.uni-mannheim.de

Abstract: Fault-tolerant control using model predictive control with online accommodation to recover from faults is investigated. A framework for this purpose is presented and problems that one encounters by changing the control law online like error-free tracking, feasibility and computational effort are addressed. In a real-time implementation, the model predictive controller is tested under actuator faults like saturation, freezing and total loss as well as under a structural fault.

1. INTRODUCTION

The industrial need of process systems to be reliable, safe and economical under faulty conditions has increased the demand to provide strategies and techniques from the researchers' community. One can divide the emerged research area into two branches: the part of Fault Detection and Identification/Diagnosis (FDI/FDD) (Gertler [1998], Patton et al. [2000]) and the part of fault-tolerant control (FTC)(Blanke et al. [2001],Patton [1997] and Zhang and Jiang [2003]). Because of the close relationship between these two topics, most available current textbooks about this theme do include treatments of both areas (Isermann [2006],Blanke et al. [2003]). Whereas FDI is established since the 70s, FTC is relatively new. It is linked to robust control, system reconfiguration/accommodation and is carried out using different control laws. One of these control laws is constrained linear model predictive control (MPC).
MPC is a today widely used and accepted control strategy in and not only for industrial environments. This is also due to its inherent property to handle constraints on inputs, outputs and state variables in an optimal manner. There are several current good textbooks giving an insight into MPC (Maciejowski [2001], Camacho and Bordons [2004], Rossiter [2003], de Dona et al. [2004])
The advantage of using model predictive control for FTC is its property to handle nonlinear faults like saturation or blocking of actuators that are reflectable by constraints very easily and also its robust nature giving passive fault tolerance. Because MPC is an accepted control algorithm for industry, modifications that offer fault tolerance are more likely to be accepted as well as implemented. The disadvantage of the computational effort to calculate the MPC control law online is in industrial environments not a severe drawback because the controlled processes have usually slow dynamics.
Getting the information about an actuator fault immediately and accurately may be seen a bit too optimistically. However, current industrial actuators have embedded fault detection and analysis algorithms and even standard on-off valves have sensors to detect failed operation. This for sure does not mean that FDI will be unnecessary in future as these mechanisms may be erroneous too. Once the new set of constraints and system parameters is provided by the FDI the calculation of the accommodated control law is not different from the calculation of the nominal control law. The difficulty arises to find a suitable and feasible set of control parameters as well as to detect unrecoverable faults. MPC used in FTC is an approach that has also been investigated by Maciejowski and Jones [2003], Huzmezan and Maciejowski [1998], Kale and Chipperfield [2005] and recently by Ocampo-Martinez [2007] among others.
However, nearly none real-time implementation has been reported so far. To our knowledge, the only exceptions are Abdel-Geliel et al. [2006] and Gopinathan et al. [2000]. Hence the main contribution of the current work is the real-time implementation and the practical results obtained on a Three-Tank-System.
The rest of the paper is organized as follows: In Section II, we will describe a Framework of an FTC-Scheme for accommodating actuator and structural faults using MPC. We describe the difficulties that arise when FTMPC is used like feasibility, error free tracking and computational issues and possibilities to cope with them are proposed. In Section III we show the benefits of the FTMPC on a real-time example under miscellaneous fault cases. Finally, Section IV is devoted to draw the conclusions.

2. THE FRAMEWORK

We assume that a FDI-Unit is available and its information is obtained accurate. The detection of the faults is without delay.

2.1 Fault Recovery

Fault tolerant control systems (FTCS) can be divided into two approaches: active and passive fault tolerance control systems (A/PFTCS). In PFTC the designer tries to achieve a robust controller against a predefined or assumed class or number of faults. AFTCS in contrast try to react to the fault by activeley recovering the system by accommodation or reconfiguration. In the literature, two possibilities for active fault recovery are distinguished:

- *Accommodation:* In fault accommodation the fault effect is compensated by changing the control law.
- *Reconfiguration:* Recovery based on reconfiguration changes control loop elements and the control law. Activating redundant pumps or using additional (backup-)sensors while switching off the faulty components one attempts to recover from the fault.

The approach taken in this work belongs to the AFTCS and uses *fault accommodation* as recover strategy.

2.2 MPC Formulation

All in nowadays sub-summized as MPC strategies share the same design philosophy: It is based on the knowledge of the process model to predict the output vector $\mathbf{y}$ H_p steps in the future. A performance index subject to constraints is then optimized during a finite period of time delimited by the Horizon H_p for the control error and H_u for the control action. Applying the *receding horizon principle*, only the first element $\mathbf{u}(k)$ of the optimized input sequence is directed to the plant and the whole procedure starts over again in the next time step.
The nearly in all MPC formulation used performance index is a quadratic one:

$$\mathbf{J}(\Delta\mathbf{u},\varepsilon) = ||\mathbf{e}(k+H_p|k)||^2_{\mathbf{S}} + \sum_{i=0}^{H_p-1} ||\mathbf{e}(k+i|k)||^2_{\mathbf{Q}} \quad (1)$$

$$+ \sum_{i=0}^{H_u-1} ||\Delta\mathbf{u}(k+i|k)||^2_{\mathbf{R_{\Delta u}}} + \rho\varepsilon^2 \quad (2)$$

$$+ \sum_{i=0}^{H_u-1} ||\mathbf{u}_s(k+i|k)||^2_{\mathbf{R}} \quad (3)$$

with

$$\mathbf{u}_s(k+i|k) = \mathbf{u}(k+i|k) - \mathbf{u}_t(k+i) \quad (4)$$

$$\mathbf{e}(k+i|k) = \mathbf{y}(k+i|k) - \mathbf{r}(k+i) \quad (5)$$

$$\Delta\mathbf{u}(k+i|k) = \mathbf{u}(k+i|k) - \mathbf{u}(k+i-1|k) \quad (6)$$

$\mathbf{e}(k+i|k) \in \mathbb{R}^l$ being the predicted output error, $\mathbf{y}(k+i|k) \in \mathbb{R}^l$, $\mathbf{r}(k+i) \in \mathbb{R}^l$, $\Delta\mathbf{u}(k+i|k) \in \mathbb{R}^m$,$\mathbf{u}(k+i|k) \in \mathbb{R}^m$, $\mathbf{u}_s(k+i|k) \in \mathbb{R}^m$, $\mathbf{u}_t(k+i) \in \mathbb{R}^m$ being the predicted output, the reference vector, the predicted control increments, the control vector, the control error and the control target respectively at time k and $\mathbf{S} = \mathbf{S^T}$ and $\mathbf{Q} = \mathbf{Q^T}$ are positive semidefinite matrices and $\mathbf{R} = \mathbf{R^T}$ and $\mathbf{R}_{\Delta u} = \mathbf{R}^T_{\Delta u}$ positive definite. The slack variable is $\varepsilon \in \mathbb{R}$ tuned by the factor $\rho \in \mathbb{R}$ which is used to relax the constraints.
We acquire the new input by minimizing

$$\min_{\Delta\mathbf{u},\varepsilon} \mathbf{J}(\Delta\mathbf{u},\varepsilon)$$

subject to

$$Constraints: \begin{cases} \mathbf{x}(k+1) & = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) \\ \mathbf{u}(k) & \in \mathbf{C}^u \\ \Delta\mathbf{u}(k) & \in \mathbf{C}^{\Delta u} \\ \mathbf{x}(k) & \in \mathbf{C}^x \end{cases} \quad (7)$$

with

$$\mathbf{C}^u = \bigcup_j C^{u_j} \quad (j = 1 \dots m) \quad (8)$$

$$C^{u_j} = \{u_j \in \mathbb{R}|\ u_j^{min} \le u_j \le u_j^{max}\} \quad (9)$$

$$\mathbf{C}^{\Delta u} = \bigcup_j C^{\Delta u_j} \quad (j = 1 \dots m) \quad (10)$$

$$C^{\Delta u_j} = \{\Delta u_j \in \mathbb{R}|\ \Delta u_j^{min} \le \Delta u_j \le \Delta u_j^{max}\} \quad (11)$$

$$\mathbf{C}^x = \bigcup_j C^{x_j} \quad (j = 1 \dots n) \quad (12)$$

$$C^{x_j} = \{x_j \in \mathbb{R}|\ x_j^{min} - \varepsilon E_j^{min} \le x_j \le x_j^{max} + \varepsilon E_j^{max}\} \quad (13)$$

and system matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, input matrix $\mathbf{B} \in \mathbb{R}^{n\times m}$, $k \in \mathbb{N}$ and relaxation coefficients $E_j^{min/max} \in \mathbb{R}$. If a constraint is hard, the corresponding relaxation coefficient is set to zero.

2.3 Fault Tolerant MPC

In Fig. 1 a scheme of the FTMPC can be seen. Let $\mathbf{M}$ be the model of the plant

$$\mathbf{M} = (\mathbf{A}, \mathbf{B}) \quad (14)$$

and $\mathbf{P}$ be the tuple of the parameters of the FTMPC

$$\mathbf{P} = (\mathbf{S}, \mathbf{Q}, \mathbf{R}_{\Delta u}, \mathbf{R}, H_p, H_u) \quad (15)$$

and let $\mathbf{C}$ be the tuple of the constraint set

$$\mathbf{C} = (\mathbf{C}^u, \mathbf{C}^{\Delta u}, \mathbf{C}^x) \quad (16)$$

The subscript n denotes nominal, f faulty, a accommodated and c corrected model, parameter and constrained sets. No subscript denotes the final set given to the MPC for control execution.
After the occurrence of a fault, the FDI provides the new set $(\mathbf{M}_f, \mathbf{C}_f)$ that is directed to the accommodation block. Here, the new objective function is constructed online, taking into account the initial and nominal sets. This means creation of the new prediction matrices depending on the given parameters. The accommodated $(\mathbf{M}_a, \mathbf{P}_a, \mathbf{C}_a)$ is fed into the *Analysis and Decision* block. The *feasibility* of the accommodated objective function is tested and when feasible directed to the MPC that executes the computation of the next input signal. If feasibility is not achieved, a corrected set $(\mathbf{M}_c, \mathbf{P}_c, \mathbf{C}_c)$ is given back to build a new objective function or, if no valid control law can be found, the system is shutdown or awaits user interaction.

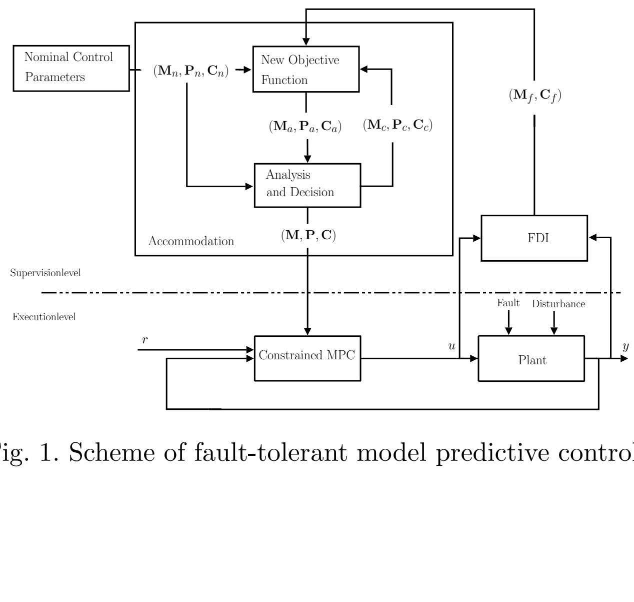

Fig. 1. Scheme of fault-tolerant model predictive control

2.4 Actuator faults

Following actuator faults can be introduced to the MPC with change in the constraint set:

(1) *Actuator region decreased:* The set of available inputs for $u_j(k)$ is reduced to the new set
$$C_f^{u_j} = \{u_j \in \mathbb{R} \mid u_j^{min} \le u_j \le u_j^{max}\}$$
(2) *Actuator freezing:* The actuator is stuck at a position in its range having as set
$$C_f^{u_j} = \{u_j \in \mathbb{R} \mid u_j = const\}$$
(3) *Control increment decreased:* The speed at which the input u_j moves is decreased. (E.g. fowling in a pipe reducing the speed of a valve)
$$C_f^{\Delta u_j} = \{\Delta u_j \in \mathbb{R} \mid \Delta u_j^{min} \le \Delta u_j \le \Delta u_j^{max}\}$$
(4) *Loss of an actuator:* The total loss of control authority for input u_j is injected as
$$C_f^{u_j} = \{u_j \in \mathbb{R} \mid u_j = 0\}$$
This fault could also be reflected by zeroing the corresponding column in the $\mathbf{B}$ matrix.

A reduction in actuator effectivity is normally reflected by a change in the input matrix resulting in $\mathbf{B}_f$. After the new $\mathbf{B}_f$ is available one needs also to adapt the concerning $\mathbf{C}^{u_j}$. If, for example, the actuator looses half its effectivity, an accommodated controller will demand twice the nominal inputsignal, which would be prohibited by the constraint otherwise.

2.5 Structural faults

Structural faults are faults that change the dynamics of the system like a blocked up pipe connection or a damaged planewing. These changes are reflected by the FDI provided new $\mathbf{A}_f$. At first, there has to be checked whether the $(\mathbf{A}_f, \mathbf{B}_f)$ is controllable or at least stabilizable before carring on.
With a change in the matrices $(\mathbf{A}_f, \mathbf{B}_f)$ the problem arises of finding a new set of $(\mathbf{M}_f, \mathbf{P}_a, \mathbf{C}_a)$ in such a way, that the behaviour of the faulty system equals the behaviour of the nominal one. In Huzmezan and Maciejowski [1998] this problem setup has been reduced to match the eigenvalues of the closed loop of $\mathbf{A}_f$ to $\mathbf{A}_n$ through an nonlinear minimization with the predicted $\mathbf{Q}$ as decision variables. This approach is rather brute-force and is not easy usable in real-time as the computational demand is high. At a first glance one could use the $\mathbf{P}_n$ without any change but the performance will detoriate with a large change in $(\mathbf{A}_f, \mathbf{B}_f)$. This point is open to be solved.

2.6 Infeasibility

One big disadvantage of using constrained MPC is the danger of running into infeasibility. Especially in the case of a model/plant mismatch, hard constraints and large disturbances this problem arises. Unfortunately this is exactly the scenario when FTMPC is used. The solution might be to drop all performance constraints and let only the hard constraints that reflect safety boundaries be active. Additionally an extension of H_p to the maximum computational possibility is advisable, since it gives the controller more time to satisfy the constraints. The only use of constraint relaxation is not advisable, as there could be a permanent violation of a safety boundary. In Vada et al. [2001a],Vada et al. [2001b] a procedure based on the lexicographic multi-objective optimization is proposed to recover the system from infeasibility. A similar approach concerning also objectives is shown in Kerrigan and Maciejowski [2002].
There is a situation, for which infeasibility is useful: With the knowledge that we only have hard constraints representing safety boundaries and there is no feasible solution for the optimization problem, we know for sure that somewhere within the H_p lasting duration there will be a violation of safety, so an emergency shutdown can be initiated before crossing that boundary.

2.7 Error-free tracking

For MPC there are different possibilities to obtain error-free tracking. A way is to omit (3) and the remaining *velocity form* of the MPC will be able to track the reference without error when the process model is correct and no permanent step disturbance is present.
Under these assumptions another way is to recalculate the steady state values of u_t in (3) by a quadratic program (See Muske [1995]). Another possibility is to use integral error augmentation of the system model. This forces the use of a prestabilization of the model to make it usable in MPC. For plants with unstable modes, this approach may be the best fit as a prestabilization is needed in either case (Kale and Chipperfield [2005]). Also a constant persistent step disturbance is filtered out.

2.8 Computational issues

A drawback of FTMPC for online accommodation or reconfiguration is often its computational demand limiting the areas of applicability. During a faulty condition this problem increases: Additionally to the load of finding a new valid (feasible) control law, a supervisory process will inform the process-user of the faulty condition who will in turn give manual inputs or run analyse programs. This may very easily insert a delay time between the information of the FDI and the adjustment of the new FTMPC control law.
The online optimization algorithms that are available for quadratic programs are *interior point* and *active set*. While *interior point* is in practice faster than *active set*, the substeps during algorithm iteration can give nonfeasible solutions. *Active set* algorithms have the advantage that each substep gives an (sub)optimal solution, which enhances from step to step and is also feasible. This point makes *active set* more preferable for FTMPC since the time between two sample steps is limited and a feasible solution, even if suboptimal, is better than an infeasible one. See Wright [1997] for the treatment of these two optimization algorithms for MPC.

3. REALTIME IMPLEMENTATION

3.1 Model

The Three-Tank-System is shown in Fig. 2. It is a well-known, often used benchmark not only for FTC. The

Table 1. Parameters for the Three-Tank-System

K_1	K_2	K_3
$2.1034\ 10^{-5}$	$3.0609\ 10^{-5}$	$2.1034\ 10^{-5}$
$k_{1..3out}$	A	g
0	$0.141(m^2)$	$9.81(m/s^2)$

Table 2. Operating point

q_{10}	q_{20}	h_{10}	h_{20}	h_{30}
$2.5\ 10^{-5}(m^3/s)$	$2.5\ 10^{-5}(m^3/s)$	$0.280m$	$0.136m$	$0.208m$

dynamic model for this plant is derived using the incoming and outgoing flow rate under consideration of TORRICELLI's law and it is described by the following non-linear equations:

$$\frac{h_1(t)}{dt} = \frac{1}{A}\left[q_{in1} - K_1\sigma(h_1(t)-h_3(t))\sqrt{2g|h_1(t)-h_3(t)|}\right]$$

$$\frac{h_2(t)}{dt} = \frac{1}{A}\left[q_{in2} + K_3\sigma(h_3(t)-h_2(t))\sqrt{2g|h_3(t)-h_2(t)|} - K_2\sqrt{2gh_2(t)}\right]$$

$$\frac{h_3(t)}{dt} = \frac{1}{A}\left[K_1\sigma(h_1(t)-h_3(t))\sqrt{2g|h_1(t)-h_3(t)|} - K_3\sigma(h_3(t)-h_2(t))\sqrt{2g|h_3(t)-h_2(t)|}\right]$$

with h_i the heights of the tanks, K_i outflow-coefficients, g the acceleration constant, A the cross-section of the tanks and $q_{in/out}$ the in and outflows. In the configuration, we used for the experiments all interconnected valves were fully open while none of the disturbance valves were open. The parameters are given in Table 1. The value of the $K_{1..3}$ were evaluated from the stationary equations after reaching the operational condition with parameters shown in Table 2.

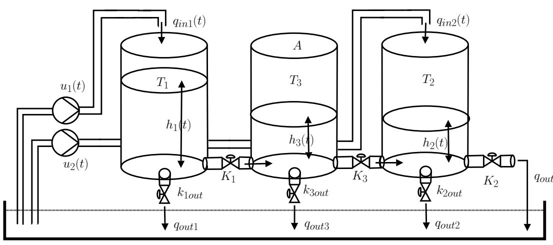

Fig. 2. Schematic diagram of the Three-Tank-System

3.2 Nominal Controller Design

After the linearisation using the parameters of Table 2 and choice of the sample time of $T_s = 1s$ the following time-discrete linear system was obtained:

$$\mathbf{A}_n = \begin{bmatrix} 0.9879 & 0.0001 & 0.0121 \\ 0.0001 & 0.9751 & 0.0120 \\ 0.0121 & 0.0120 & 0.9759 \end{bmatrix}, \mathbf{B}_n = \begin{bmatrix} 70.3047 & 0.0018 \\ 0.0018 & 69.8505 \\ 0.4290 & 0.4272 \end{bmatrix}$$

$$y(k) = h_3(k)$$

The control aim was to track the height in Tank 3 with minimal overshoot. For the nominal controller the choice of the parameters for the MPC is

$$\mathbf{S} = 0, \mathbf{Q} = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 100 & 0 \\ 0 & 300 \end{bmatrix}$$

$$\mathbf{R}_{\Delta\mathbf{u}} = \begin{bmatrix} 1000 & 0 \\ 0 & 1000 \end{bmatrix}, H_p = 15, H_u = 3 \quad (17)$$

$$C_n^{u_1} = \{u_1|0\% \leq u_1 \leq 100\%\} \quad (18)$$

$$C_n^{u_2} = \{u_2|0\% \leq u_2 \leq 100\%\} \quad (19)$$

$$\mathbf{C}_n^{\Delta u} = \emptyset, \mathbf{C}_n^{x} = \emptyset \quad (20)$$

The nominal behaviour of the plant to a step input $r(k)$ without any fault occuring is presented in Fig. 3.

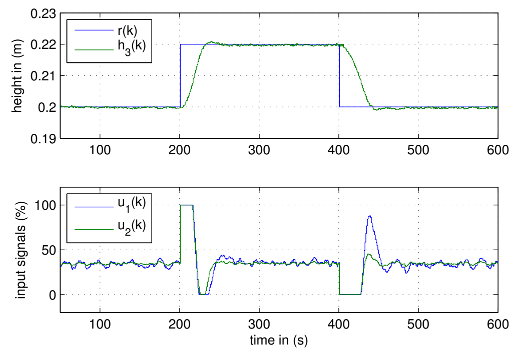

Fig. 3. Plot of nominal output of Tank 3 with reference signal and the input signals u_1, u_2

3.3 Fault Scenarios

To show the benefit of FTMPC in real-time we conducted several experiments. The duration of the experiments was in each case $t_{final} = 600(s)$ and the fault was injected at $t_f = 200(s)$. Only the sets described in the fault cases were changed, the rest remained unchanged. The accommodated MPC has been calculated offline and switched on at t_f. Besides an optical evaluation we also used a performance index to get a more objective comparison. The PI was chosen as:

$$PI = \sum_{l=t_f}^{t_{final}} |r(l) - h_3(l)| \quad (21)$$

See Table 3 for the PI of the different fault scenarios with and without accommodation. The PI of the nominal operation was $PI_n = 0.9310$

The following fault-scenarios were conducted and results were achieved:

Fault case 1: Saturation fault on input $u_1(k)$

The input $u_1(k)$ has a decreased maximum of

$$C_f^{u_1} = \{u_1|0\% \leq u_1 \leq 24.14\%\} \quad (22)$$

equaling a maximum inflow of q_{in1} to $2.0\ 10^{-5}(m^3/s)$. In Fig. 4, the result of the experiment is shown. The non-accommodated controller is not able to reach the reference $r(k)$. The FTMPC is able to reach the reference with nearly zero tracking error.

Fault case 2: Actuator-freezing on input $u_1(k)$

The input $u_1(k)$ is stuck at

$$C_f^{u_1} = \{u_1 \mid u_1 = 17.7\%\ \} \quad (23)$$

or $q_{in1} = 2.0\ 10^{-5}(m^3/s)$ permanent inflow in Tank 1. Like in the fault-case before the FTMPC reaches the reference with nearly zero tracking error (see Fig. 5). The non-accommodated MPC has a permanent steady-state error.

Table 3. Performance index PI of the fault scenarios

	non-accommodated	accommodated
Fault case 1	1.7675	1.1322
Fault case 2	1.9343	1.1896
Fault case 3	1.6226	1.3391
Fault case 4	1.8115	1.6220
Fault case 5	1.6833	1.7143

Fault case 3: Decreased input difference $\Delta u_1(k)$
In this case, the fault decreases the $\Delta u_1(k)$ to

$$C_f^{\Delta u_1} = \{\Delta u_1 | -1.5\% \leq \Delta u_1 \leq 1.5\%\} \quad (24)$$

In Fig. 6 the unaccommodated controller shows an oscillatory behaviour. The FTMPC exhibits also this behaviour but to a minor degree resulting in an optically better performance.

Fault case 4: Loss of input $u_2(k)$
The total loss of input $u_2(k)$ occurs at t_f. The resulting constraint is

$$C_f^{u_2} = \{u_2 \mid u_2(k) = 0\} \quad (25)$$

In Fig. 7 one can observe that the FTMPC has a slightly better performance achieving zero tracking error. However, the unaccommodated controller is also able to give satisfying result.

Fault case 5: Structural fault
In this case the inflow from Tank 1 to Tank 3 is decreased (simulated by hand tuning the valve for K_1 at t_f, the new operating condition and resulting matrices have been determined before). The new matrices $\mathbf{A}_f, \mathbf{B}_f$ are:

$$\mathbf{A}_f = \begin{bmatrix} 0.9961 & 0 & 0.0039 \\ 0 & 0.9684 & 0.0125 \\ 0.0039 & 0.0125 & 0.9835 \end{bmatrix}, \mathbf{B}_f = \begin{bmatrix} 70.5961 & 0.0006 \\ 0.0006 & 69.6110 \\ 0.1388 & 0.4440 \end{bmatrix}$$

The result (Fig. 8) shows the FTMPC with better tracking performance but has a larger negative overshoot after the occurrence of the fault giving it a worse PI than the unaccommodated case. This example shows that using the same set of constraints and parameters for the accommodation can despite the availability of the correct model give worse results. Here, a tuning of the $(\mathbf{P})$ could improve the results.

4. CONCLUSION

In this work we presented a framework for online accommodation of faults with MPC. The problems one encounters are challenging. Not only the computational demand but more the possibility of infeasibility is a major drawback for the use of the FTMPC online. However, our real-time results clearly showed the advantage of the FTMPC in the face of non-linear faults.

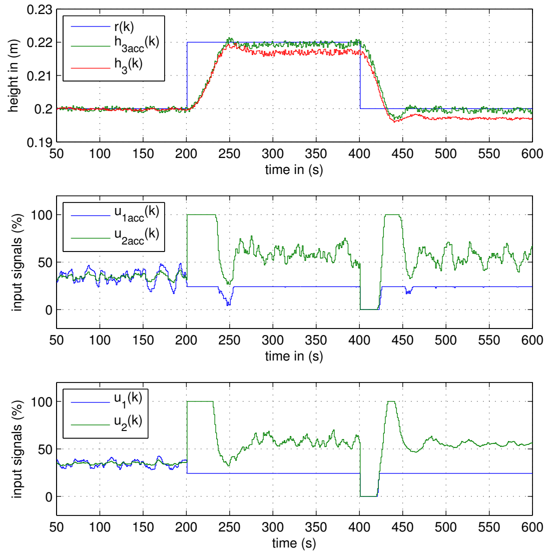

Fig. 4. Saturation fault on input u_1. Subscript *acc* is the accommodated, none subscript the non-accommodated case

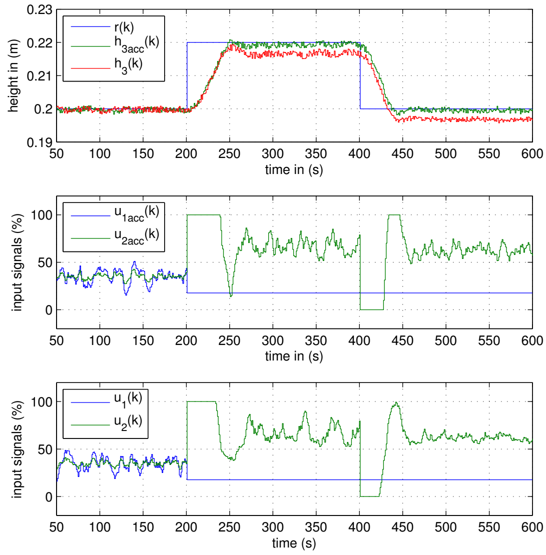

Fig. 5. Actuator stuck on input u_1. Subscript *acc* is the accommodated, none subscript the non-accommodated case

REFERENCES

Janos J. Gertler *Fault Detection and Diagnosis in Engineering Systems* Marcel Dekker Ltd, 1998

Ron J.Patton, Paul M. Frank, Robert N. Clark,*Issues for Fault Diagnosis for Dynamic Systems*,Springer-Verlag, 2000

M. Blanke, M. Staroswiecki, and E. Wu, *Concepts and methods in fault-tolerant control.* American Control Conference, Arlington, Virginia, USA, 2001, pp. 26062620.

R. Patton, *Fault tolerant control: the 1997 situation.* 3rd IFAC Symposium on Fault Detection, Supervision and Safety for Technica Processes - SAFEPROCESS, Hull, UK, 1997, pp. 10331054.

Y. Zhang and J. Jiang, *Bibliographical review on reconfigurable fault tolerant control systems.* 5th IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes - SAFEPROCESS Washington DC, USA, 2003, pp. 265276.

R. Isermann *Fault-Diagnosis Systems - An Introduction from Fault Detection to Fault Tolerance* Springer-Verlag, 2006

M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki *Diagnosis and Fault-Tolerant Control* Springer-Verlag, 2003

J.M. Maciejowski. *Predictive Control with Constraints.*, Prentice Hall, UK, 2001.

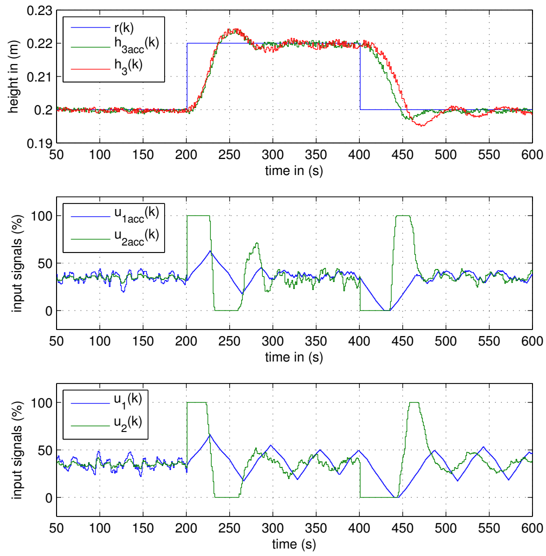

Fig. 6. Control increment decreased on input u_1. The subscript *acc* is the accommodated, none subscript the non-accommodated case

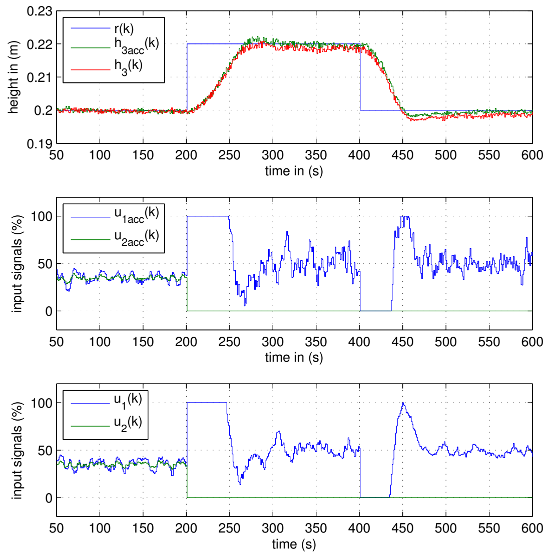

Fig. 7. Loss of input u_2. Subscript *acc* for the accommodated, none subscript for the non-accommodated case

E. F. Camacho and C. Bordons, *Model Predictive Control*, Springer-Verlag, London, 2004
J. A. Rossiter, *Model-Based Predictive Control: A Practical Approach* CRC Press, Inc., 2003
Jose De Dona, Graham C. Goodwin, Maria Marta Seron. *Constrained Control and Estimation. An Optimization Approach.* Springer-Verlag, 2004.
Maciejowski, J. and Jones, C. (2003), *MPC Fault-Tolerant Flight Control Case Study: Flight 1862* in Proceeding of the SAFEPROCESS 2003: 5th Symposium on Detection and Safety for Technical Processes, Washington D.C., USA: IFAC, p. 265-276
M. Huzmezan and J. Maciejowski, *Automatic tuning for model based predictive control during reconfiguration* Automatic Control in Aerospace 1998, Proceedins from the 14th IFAC Symposium, Soeul, Korea, 24-18 August,

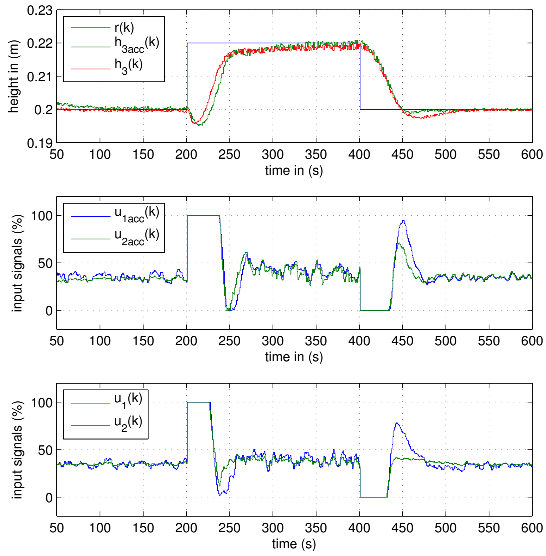

Fig. 8. Structural fault - pipe to Tank 3 partially blocked. Subscript *acc* denotes the accommodated, none subscript the non-accommodated case

1998. Edited by Jang Gye Lee. Published by Pergamon Press for the International Federation of Automatic Control, 1999., p.237
M.M. Kale,A.J. Chipperfield,*Stabilized MPC formulations for robust reconfigurable flight control*, Control Engineering Practice,Volume 13,p771-788,2005
C. Ocampo-Martinez,*Model Predictive Control of Complex Systems including Fault Tolerance Capabilities: Application to Sewer Networks.* Technical University of Catalonia, Automatic Control Department. 2007
Abdel-Geliel, M. and Badreddin, E. and Gambier, A., *Application of model predictive control for fault tolerant system using dynamic safety margin*,American Control Conference, 2006
M. Gopinathan,Raman K. Mehra,Joseph C. Runkle *Hot Isostatic Pressing Furnaces- Their Modeling and Predictive Fault-Tolerant Control*,IEEE Control Systems Magazine,pp. 67-82,Dec. 2000
J. Vada, O. Slupphaug and T.A. Johansen. *Optimal prioritized infeasibility handling in model predictive control: parametric preemptive multiobjective linear programming approach.*, Journal of Optimization Theory & Applications 109:2, 385413, 2001.
J. Vada, O. Slupphaug, T.A. Johansen and B.A. Foss. *Linear MPC with optimal prioritized infeasibility handling: application, computational issues and stability.* Automatica 37:11 1835 1843, 2001.
E.C. Kerrigan, J. M. Maciejowski,*Designing model predictive controllers with prioritised constraints and objectives*, Proceedings of the IEEE Symposium on Computer Aided Control System Design,33-38,2002
Kenneth Robert Muske,*Linear Model Predictive Control of Chemical Processes*, Ph.D. The University of Texas at Austin,1995
S. J. Wright. *Applying new optimization algorithms to model predictive control.*, In Chemical Process Control-V, volume 93 of AIChE Symposium Series, Number 316, pages 147–155. CACHE Publications, 1997.