

# Fault-Tolerant Explicit MPC of PEM Fuel Cells

Vicenç Puig, Albert Rosich, Carlos Ocampo-Martínez and Ramon Sarrate

**Abstract**—In this paper, fault-tolerant explicit MPC control of fuel cell systems is presented. MPC is one of the control methodologies that allows to introduce fault-tolerance more easily. Here, this capability is extended using recent explicit MPC control theory. Explicit MPC control allows to derive off-line the control without using optimization. Moreover, it allows to introduce as additional parameters faults since it is based on parametric programming. This makes possible to change in real-time controller parameters without recomputing the MPC controller or having a bank of pre-computed MPC controllers. Finally, the proposed approach is assessed on a known test bench PEM fuel cell system.

## I. INTRODUCTION

FUEL cell systems have been developed considerably along the last years. Although they were invented more than a century ago, they have received much attention in the last decade as good candidates for clean electricity generation both in stationary and automotive applications. There are still many open issues related to fields such as materials, manufacturing or maintenance, being automatic control one of the most important. There exist many types of fuel cells [9], being this work devoted to PEM (Polymer Electrolyte Membrane) fuel cells, which run at low temperature and show fast dynamical response, which make them suitable for mobile applications. It is clear that good performance of these devices is closely related to the kind of control that is used, so a study of different control alternatives is justified [11]. A fuel cell system is not composed of the fuel cell alone but it integrates many components into a power system, which supplies electricity to an electric load or to the grid. Several devices such as DC/DC or DC/AC converters, batteries or ultracapacitors are included in the system and, in case the fuel cell is not fed directly with hydrogen, a reformer must also be used. Therefore, there are many control loops schemes depending on the devices that must be controlled. The lower control level takes care of the main control loops inside the fuel cell, which are basically fuel/air feeding, humidity, pressure and temperature. The upper control level is in charge of the whole system, integrating the electrical conditioning, storage and reformer (if necessary). Many control strategies have been proposed in literature, ranging from feedforward

control [11], LQR [12][15], Neural Networks [1], [7] or Model Predictive Control [5][17].

This paper is focused on the control of fuel cell systems using explicit Model Predictive Control (MPC) [2][16]. Explicit MPC control allows to derive off-line the control without need of using optimization, since it is based on parametric programming. The MPC controller has as a control objective “oxygen starvation” prevention and uses as manipulated variables air feeding through the compressor voltage. There are other variables such as output voltage and power, cell temperature, reactive pressures or humidity that can be easily included in the control strategy to improve performance. Notice that air feeding has crucial importance on fuel cell behaviour, as shown in [11]. However, due to a fuel cell system is very complex, it is prone to suffer from faults in its operation time. So, some fault tolerant capabilities should be added to the control system in order to maintain the fuel cell system under control even in the presence of faults. This paper explores the possibility of making using of the known inherent fault-tolerant capabilities of MPC control. Moreover, these capabilities are extended using explicit MPC control. The use of parametric programming allows to introduce as additional parameters faults what allow in real-time to change controller parameters without re-computing the MPC controller or having a bank of pre-computed MPC controllers. Finally, the fault tolerant MPC controller is tested on a full nonlinear model of a PEM fuel cell, showing the usefulness the proposed approach. The remainder of paper is organized as follows: in *Section II*, constrained MPC principles are recalled and new results on explicit MPC are briefly summarized. In *Section III*, the inclusion of fault-tolerance in explicit MPC is discussed. Then, the MPC fault tolerant controller is implemented and tested using a non-linear model of the fuel cell system and showing the results in *Section IV*. Finally, the major conclusions are drawn in *Section V*.

## II. IMPLICIT AND EXPLICIT MPC CONTROL

### A. Implicit MPC law computation

Model Predictive Control (MPC) has become the accepted standard for complex constrained multivariable control problems in the process industry. At each sampling time, starting at the current state  $x(0)$ , that either is considered measured or estimated, an open-loop optimal control problem is solved over a finite horizon  $N$ :

Manuscript received March 9, 2007. The authors wish to thank the support received by the Research Commission of the Generalitat of Catalunya (Grup SAC ref. 2005SGR00537) and by CICYT (ref. DPI-2005-05415) of Spanish Ministry of Education.

V. Puig, A. Rosich, C. Ocampo-Martínez and R. Sarrate are with the Automatic Control Department-Campus de Terrassa, Universidad Politècnica de Catalunya (UPC), Rambla Sant Nebridi 10, 08222 Terrassa (Spain) (e-mail: vicenc.puig@upc.edu).

$$\begin{aligned}
& \min_{u(0), \dots, u(N-1)} \sum_{k=0}^{N-1} \left[ x^T(k) Q x(k) + u^T(k) R u(k) \right] \\
& \quad + x^T(N) P x(N) \\
& \text{subject to :} \\
& x(k+1) = A x(k) + B u(k) \quad k = 1, \dots, N-1 \\
& u(k) \in [\underline{u}, \bar{u}] \quad k = 0, \dots, N-1 \\
& x(k) \in [\underline{x}, \bar{x}] \quad k = 1, \dots, N
\end{aligned} \tag{1}$$

where  $[\underline{u}, \bar{u}]$  and  $[\underline{x}, \bar{x}]$  are the operating ranges of controls and states. Assuming that the optimization problem (1) is feasible, there exists an optimal solution given by the sequence of controls:  $u^*(0), \dots, u^*(N-1)$ . However, only the first control of this sequence  $u^*(0)$  is applied. Then, at the next time step, the computation is repeated starting from the new measured (or estimated) state and over a shifted horizon, leading to a moving horizon policy. The solution relies on a control law that respects all input and output constraints, and optimizes a quadratic performance index. Over the last decade, a solid theoretical foundation for MPC has emerged so that in real-life large-scale MIMO applications controllers with non-conservative stability guarantees can be designed routinely and with ease [14][13].

Implicit MPC is based on the solution of a quadratic program (QP) that allows to determine the optimal control action:

$$\begin{aligned}
V(x(0)) &= \frac{1}{2} x^T(0) Y x(0) \\
&+ \min_U \left[ \frac{1}{2} U^T H U + x^T(0) F U \right] \\
&\text{subject to : } GU \leq W + S x(0)
\end{aligned} \tag{2}$$

where:

$$U = [u^T(0), \dots, u^T(N-1)]^T \tag{3}$$

is the optimizer vector and  $H, F, Y, G, W, S$  depend on weights  $Q, R, P$ , upper and lower bounds of  $u$  and  $y$ , and model restrictions  $A, B$  and  $C$ . Since QP optimization problem is convex there is no problem with local optimums (only an optimum exists). Additionally, efficient algorithms are available (active set and interior point methods) that allow to solve this problem very fast. Thus, the standard way of computing the MPC law, which is implemented in all commercial MPC packages, is to solve the QP problem (2) numerically on line at each time  $k$  [13]. The big drawback of implicit MPC is the on-line computational effort which may limit its applicability to relatively slow and/or small problems.

### B. Explicit MPC law computation

In [2], it has been shown how to move the computations necessary for the implementation of MPC off-line while preserving all its other characteristics. This should largely

increase the range of applicability of MPC to other problems. Such an explicit form of the controller provides also additional insight for better understanding the control policy of MPC. In this paper, the explicit form will provide additional insight on how an MPC controller can manage faults. There are other advantages obtained by using explicit MPC controller. Since the resulting explicit control law allows implementation without real-time optimization software, it can be made on inexpensive hardware, using fixed point arithmetic instead of the floating point operations required by numerical optimization software. A software implementation would require only a few lines of code, which would simplify the verification of the implementation. Such solutions will be particularly well suited for safety-critical applications (automotive, biomedical etc.), where the industry would not accept real-time numerical solvers due to software verification and software complexity issues. Another advantage is that the worst-case computation time for the control law can be clearly stated a priori, guaranteeing a solution to be computed within possibly tight hard real-time bounds. However, there are also some disadvantages of using MPC controllers compared to using the more conventional method with on-line solution of an optimization problem. The most obvious disadvantage is the rapid growth of size in the explicit solution as the problem size increases. This limits the use of these solutions to small problems. This limitation is primarily due to the on-line memory requirements becoming too high. In general one can say that using an explicit solution leads to lower requirements for CPU power, but higher memory requirements.

Since linear MPC is based on the solution of a quadratic program (QP), whose coefficients of the linear term in the cost function and the right hand side of the constraints depend linearly on the current state, the quadratic program can be viewed as a multiparametric quadratic program (*mp-QP*). Then, the QP program (2) can be converted into a multiparametric Quadratic Program (*mp-QP*) given by

$$V(x) = \min_U \left[ \frac{1}{2} U^T H U + x^T F^t U \right] \tag{4}$$

$$\text{subject to : } GU \leq W + Sx$$

that must be solved for all  $x$ . In [2], properties of *mp-QP* are analyzed showing that the optimal solution is a piecewise affine function of the vector of parameters. As a consequence, the solution is an explicit MPC law that is piecewise affine (PWA) with respect to states:

$$u(x) = \begin{cases} F_1 x + g_1 & \text{if } H_1 x \leq K_1 \\ \vdots & \\ F_m x + g_m & \text{if } H_m x \leq K_m \end{cases} \tag{5}$$

which not only ensures feasibility and stability, but is also optimal with respect to LQR performance. This allows to solve QP optimization problem associated to the MPC problem off-line. As in the case of implicit MPC, if not all states are measured, they should be estimated. An algorithm based on a geometric approach for solving *mp-QP* problems, and therefore obtain explicit MPC controllers, was proposed

in [2]. More recently, in [16], a faster algorithm based on an active-set approach is proposed.

### III. EXPLICIT FAULT-TOLERANT MPC CONTROL

#### A. Introduction

Fault-tolerant control is an incipient research area in the automatic control field [6]. One way of achieving fault-tolerance, known as active, is to employ a fault detection and isolation (FDI) module on-line. The FDI module will generate a discrete event signal to a supervisor system when a fault is detected and isolated. The supervisor, in turn will activate some accommodation action in response, which can be pre-determined for each fault or obtained from real-time analysis and optimization. Due to these discrete event nature of fault occurrence and the reconfiguration/accommodation, a FTC system is hybrid system by nature. Therefore, the analysis and design of FTC systems is not trivial. For design purposes, traditionally the hybrid nature of FTC has been neglected in order to facilitate a simple design, reliable implementation, and systematic testing. Then, the whole FTC scheme can be expressed using the three-level architecture for FTC systems proposed by Blanke [4] (see Fig. 1).

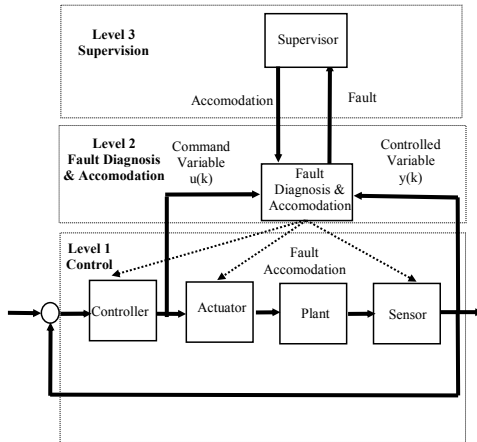


Fig. 1 Fault-tolerant architecture

#### B. Inclusion of fault tolerance in MPC

Fault-tolerance against faults can be embedded in MPC is relatively easy [10]. This can be done in several ways:

- Changing the constraints in order to represent certain kinds of fault, being specially “easy” to adapt the algorithms for faults in actuators, assuming that the fault can be located and may be their effects estimated using an FDI module.
- Modifying the internal plant model used by the MPC in order to reflect the fault influence over the plant using the information provided by an FDI module.
- Relaxing the initial control objectives in order to reflect the system limitations under fault

conditions.

#### C. Inclusion of fault tolerance in explicit MPC

Easy reconfiguration is traditionally considered one of the advantages of MPC, but reconfiguring an explicit solution might seem at a first glance that will need considerable off-line computation time. However, the use of parametric programming allows to express MPC control problems as parametric programs. This fact leads to introduce faults as extra parameters into the parametric program:

$$u(x, f) = \begin{cases} F_l \begin{bmatrix} x \\ f \end{bmatrix} + g_l & \text{if } H_l \begin{bmatrix} x \\ f \end{bmatrix} \leq K_l \\ \vdots \\ F_{m+n_f} \begin{bmatrix} x \\ f \end{bmatrix} + g_{m+n_f} & \text{if } H_{m+n_f} \begin{bmatrix} x \\ f \end{bmatrix} \leq K_{m+n_f} \end{cases} \quad (6)$$

For example, in the case of faults affecting actuator bounds, since the maximum control input from an actuator is often constrained in the optimization formulation, this constraint can be considered a parameter. Then, if, for instance, an actuator has failed, one can handle this by constraining the corresponding control input to be zero or to the range where the actuator is still operating.

#### D. Motivational example of fault-tolerant explicit MPC

An motivational example is used to show how an explicit MPC controller handles a fault situation. Let us consider a first-order continuous system described by the transfer function

$$G(s) = \frac{0.8}{2s+1} \quad (7)$$

whose equivalent discrete-time state-space description, using a sampling time  $\Delta t = 0.1$  s, is given by

$$\begin{aligned} x_{k+1} &= 0.9512x_k + 0.0975u_k \\ y_k &= 0.8x_k \end{aligned} \quad (8)$$

with the following constraints in the actuator:  $|u_k| < \delta$  with  $\delta = 1$ , i.e., the actuator operating range is  $u_k \in [\underline{\delta}, \bar{\delta}] = [-1, 1]$ . An MPC controller is used in the closed-loop system satisfying the associated control constraints and considering the cost function

$$J(x_k, u_k) = Px_{H_p}^2 + \sum_{i=0}^{H_p-1} (Qx_i^2 + Ru_i^2) \quad (9)$$

where  $H_p = 2$  and the terminal weight matrix  $P$  is determined using the Ricatti equation with  $Q = 1$  and  $R = 0.1$ . According to *Theorem 6.2.1* in [8], since in this particular case the prediction horizon is 2, the explicit form of the optimal control law  $u_k^* = \mathcal{K}_2(x)$ , which depends on the current system state  $x_0 = x$ , is given by

$$\mathcal{K}_2(x) = \begin{cases} -\text{sat}_\delta(Gx+h) & \text{if } x \in \mathbb{Z}^- \\ -\text{sat}_\delta(Kx) & \text{if } x \in \mathbb{Z} \\ -\text{sat}_\delta(Gx-h) & \text{if } x \in \mathbb{Z}^+ \end{cases} \quad (10)$$

where the saturation function  $\text{sat}_\delta(\bullet)$  is defined, for the saturation level  $\delta$ , as

$$\text{sat}_\delta(u_k) = \begin{cases} \delta & \text{if } u_k > \delta \\ u_k & \text{if } |u_k| \leq \delta \\ -\delta & \text{if } u_k < -\delta \end{cases} \quad (11)$$

$K$  and  $P$  are obtained by the algebraic Ricatti equation

$$\begin{aligned} P &= A^T P A + Q - K^T (R + B^T P B) K \\ K &= (R + B^T P B)^{-1} B^T A \end{aligned} \quad (12)$$

which gives  $P = 3.2419$  and  $K = 2.2989$  for the current example. Also from *Theorem 6.2.1* in [8], the gain  $G \in \mathbb{R}^{1 \times n}$  and the constant  $h \in \mathbb{R}$

$$G = \frac{K + K B K A}{1 + (K B)^2} \quad \text{and} \quad h = \frac{K B}{1 + (K B)^2} \delta \quad (13)$$

which gives  $G = 2.6557$  and  $h = 0.2135$ . The state space partitions for control law (10) are defined by

$$\begin{aligned} \mathbb{Z}^- &= \{x : K(A - BK)x < -\delta\} \\ \mathbb{Z} &= \{x : |K(A - BK)x| \leq \delta\} \\ \mathbb{Z}^+ &= \{x : K(A - BK)x > \delta\} \end{aligned} \quad (14)$$

which determines in this particular case the sets

$$\begin{aligned} \mathbb{Z}^- &= \{x : 1.6713x < -\delta\} \\ \mathbb{Z} &= \{x : |1.6713x| \leq \delta\} \\ \mathbb{Z}^+ &= \{x : 1.6713x > \delta\} \end{aligned} \quad (15)$$

It can be noticed that control law (10) depends indirectly of the actuator limits given by  $\delta$ , through expressions in (13) and (14). Therefore, it is clear how the effect of a fault over the actuator operating range can modify the expression of control law. This suggests that (10) can be parameterized in function of the actuator faults (limits). This parametrization is possible using results given in *Section II.B*, where state-feedback explicit control law for the MPC controller, piecewise affine with respect to the states, can be derived using multiparametric quadratic programming (*mpQP*). Using this approach in the current example, the expression of  $\mathcal{K}_2(\bullet)$  is

given in function of the parameters  $\theta = [x \quad \underline{\delta} \quad \bar{\delta}]^T$ , which constitutes an extended system composed by the system state and the control input bounds of its operating range. Thus, expression

$$\mathcal{K}_2(\theta) = \begin{cases} [-2.2990 \ 0 \ 0]\theta & \text{if } \begin{bmatrix} 1.6713 & 1 & 0 \\ -1.6713 & 0 & -1 \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ [-2.6560 \ 0 \ -0.2135]\theta & \text{if } \begin{bmatrix} 0 & 1 & -1 \\ 1.2720 & 0 & 0.7611 \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ [-2.6560 \ -0.2135 \ 0]\theta & \text{if } \begin{bmatrix} 0 & 1 & -1 \\ -1.2720 & -0.7611 & 0 \end{bmatrix} \theta \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{cases} \quad (16)$$

corresponds to the explicit PWA control law, which has been obtained and represented graphically (see Figure 2) using the *mp-programming* tools included in the *Hybrid*

*Toolbox* [3]. Comparing expressions (10) and (16), it can be seen that both control laws are equivalent.

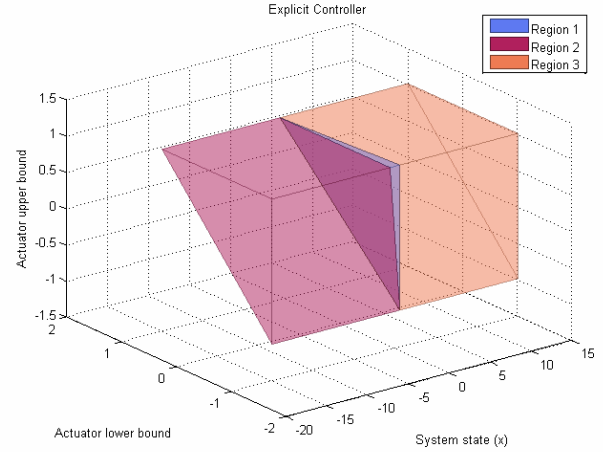


Fig. 2 Explicit controller as a function of state and actuator faults (limits)

#### IV. APPLICATION EXAMPLE: A FUEL CELL SYSTEM

##### A. Fuel-cell based system description

To test the proposed approach a known test-bench PEM fuel cell based on the model proposed by [12] will be used. This model is widely accepted nowadays in the control community as a good representation of the behaviour of an actual fuel cell system for control purposes. It is a lumped parameter model that describes quite well the system dynamics. This model considers that the operating temperature inside the cells and reactive humidity are controlled. So, these variables can be considered to be constant. Hydrogen supply is controlled using the inlet valve in such a way that hydrogen pressure in the anode tracks oxygen pressure in the cathode. This is done by a simple proportional controller in order to avoid high differential inlet pressure which could spoil the device. The main control action is therefore oxygen (or air) pressure, which is manipulated by acting on the compressor voltage, as shown in Figure 3. The control objective chosen is the oxygen excess ratio. This variable is used to avoid starvation phenomenon that can deteriorate or even spoil the fuel cell. The main characteristics of the fuel cell used in this work are [12]: Number of cells is 381, material of the membrane is Nafion 117, active area is 280 cm<sup>2</sup>, nominal stack voltage is 45V, nominal stack current is 191A and maximum power is 75kW.

##### B. MPC control for the fuel-cell based system

Model Predictive Control oxygen excess ratio control will be implemented using the *Hybrid Toolbox* [3]. The fuel cell system linear model used to implement the MPC is derived, through a linearization at operating point:  $P_{net}=40\text{kW}$ ,  $\lambda_{O_2}=2$  and  $V_{st}=235\text{V}$  in measured variables;  $I_{st}=191\text{A}$  in measured input disturbances; and  $V_{cm}=164\text{V}$  in manipulated variable, as it is also suggested in [12]. MPC weights are tuned to achieve the desired control goal, i.e., to maintain the oxygen excess ratio. The weight associated this objective has of a

value of 10 for a good performance control, after some “trial and error” experimentation. The air compressor voltage is modeled as a constrained input due to physical limits (maximum compressor voltage cannot exceed 230V, and voltage value is never negative). The oxygen excess ratio is modeled using output constraint (the operating range is between 1.5 and 3) in order to avoid starvation. Figure 4 shows the evolution of the oxygen excess ratio when the designed MPC controller is used when a series of step changes in stack current are applied. This variable is considered as measured disturbance for MPC controller. The compressor voltage is the control action computed by MPC. Notice that the control goal is achieved, providing a maintained value (2.0) of oxygen excess ratio.

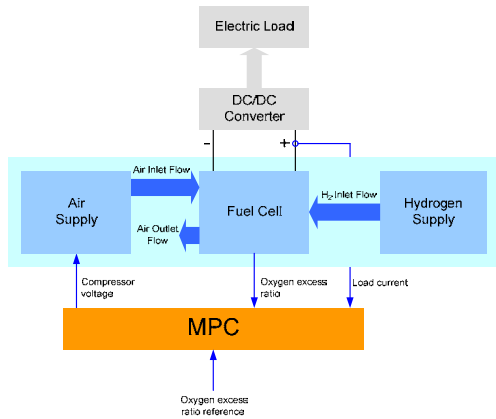


Fig. 3 Schematic diagram of the fuel cell system

### C. Fault tolerant explicit MPC

As explained in previous sections, the MPC formulation allows to include easily fault tolerant control capabilities in the control law. In this paper, faults affecting the compressor range of operation are treated. According to *Section III*, the FDI module should provide the MPC controller the new limits of compressor voltage in every sample time, once the fault has been detected, isolated and estimated. A global structure is shown in Figure 5 where the limits for the actuator range are computed by FDI module. The FDI module (drawn in dashed line) is not implemented in this work, assuming it is available and works perfectly. In order to take into account changes in the compressor range (actuator limits), linear model for MPC design is modified by including the actuator limits as a new states that will be estimated by the FDI module:

$$\begin{aligned} \begin{bmatrix} \dot{x}_{1..n} \\ \dot{x}_{n+1} \\ \dot{x}_{n+2} \end{bmatrix} &= \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1..n} \\ x_{n+1} \\ x_{n+2} \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} u \\ \begin{bmatrix} y_{1..m} \\ y_{m+1} \\ y_{m+2} \end{bmatrix} &= \begin{bmatrix} C & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{1..n} \\ x_{n+1} \\ x_{n+2} \end{bmatrix} + \begin{bmatrix} D \\ -1 \\ -1 \end{bmatrix} u \end{aligned} \quad (17)$$

where  $A$ ,  $B$ ,  $C$ ,  $D$  are the original fuel system matrices. In (17),  $x_{n+1}$  corresponds to the current upper limit while  $x_{n+2}$  corresponds to lower limit role. Moreover, additional output constraints have been added to the MPC controller:  $y_{m+1} \geq 0$  and  $y_{m+2} \leq 0$ , which ensures that the computed control

variable  $u$  will be into the range estimated by FDI module. Notice that from the controller point of view,  $y_{m+1} = x_{n+1}$  and  $y_{m+2} = x_{n+2}$ . Thus, the only way to respect these constraints is by modifying the control variable  $u$ . Extended model (17) is used in order to parameterize the controller with respect to faults in actuator limits. The result is a PWA affine controller with 79 regions following the structure (6). In Figure 6, a projection of this PWA controller on two variables is presented: the Oxygen Excess Ratio (state variable) and the Upper Limit of the Compressor Operating Range (fault variable). This allows to visualize how the controller gain changes depending on the size of the fault in the actuator. Figures 7, 8 and 9 show the simulation results of FTC scheme considering several fault actuator scenarios. The current applied to the stack is the same than in the non-faulty scenario presented in Figure 4. Dashed line represents the actuator limit that should have been estimated by the FDI module. The control action is shown in Figure 7 when the actuator (air compressor) fault causes a limit range reduction to 0-75% of the original one. In this case, the control degradation is minimal as the fault does not affect the control action. In Figure 8, shows the case corresponding to the range is reduced to 0-50%. Now, the control degradation is visible when the values of stack current are high. Finally, in Figure 9 the limit range is reduced to 0-25%. In this case, the control goal is not achieved once the actuator fault has appeared.

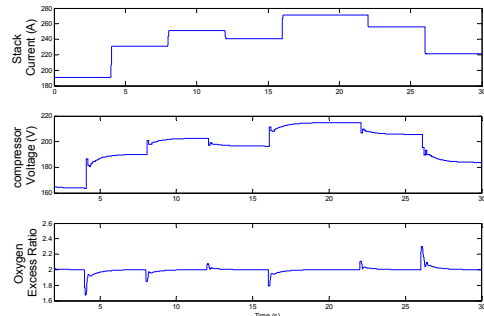


Fig. 4. Simulation results of the fuel cell system for the oxygen excess ratio control using implicit MPC (non-faulty scenario)

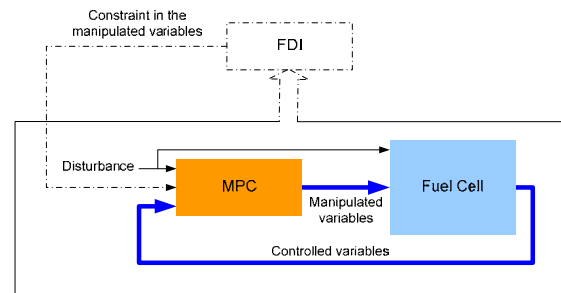


Fig. 5 Fault Tolerant MPC schema for air compressor faults.



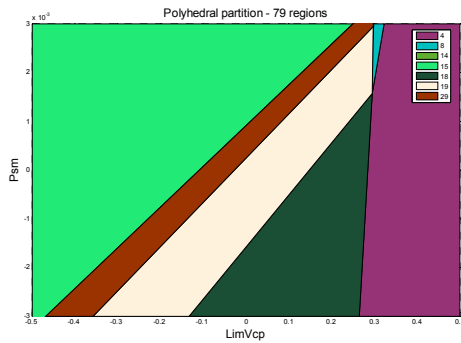


Fig. 6. Projection of PWA explicit controller on output state variable (oxygen excess ratio) and fault variable (upper limit of the compressor operating range).

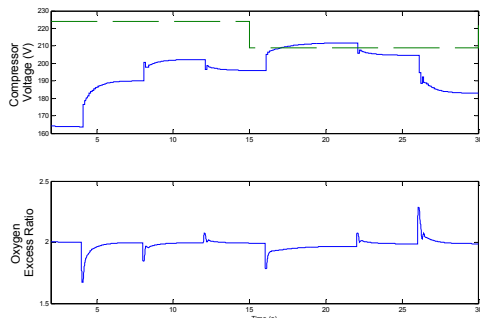


Fig. 7. Fault-tolerant implicit MPC results in case of an actuator fault that limits operating range to 0-75%.

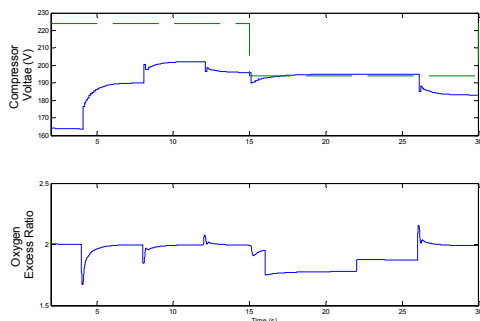


Fig. 8. Fault-tolerant implicit MPC results in case of an actuator fault that limits operating range to 0-50%.

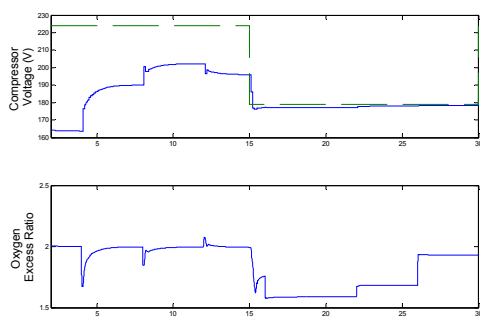


Fig. 9. Fault-tolerant implicit MPC results in case of an actuator fault that limits operating range to 0-25%.

## V. CONCLUSIONS

In this paper, explicit MPC control of fuel cell systems has been presented. MPC is one of the control methodologies that can introduce more easily fault-tolerance. Here this capability has been extended using new results on explicit MPC control. Explicit MPC control allows to derive off-line the control law without having to solve an optimization problem on-line. Moreover, since explicit MPC is based on parametric programming allows to introduce as additional parameters faults what allow in real-time to change controller gains without re-computing the MPC controller or having a bank of pre-computed MPC controllers. Finally, the proposed approach has been assessed on a known test bench fuel cell obtaining very promising results.

## REFERENCES

- [1] Almeida, P.E.M. and M. Godoy Simoes. "Neural optimal control of PEM fuel cells with parametric CMAC networks", IEEE Transactions on Industry Applications. Volume 41 (1), pp. 237 – 245. 2005
- [2] Bemporad, A., Morari, M., Dua, V., and E.N. Pistikopoulos, E.N. The explicit linear quadratic regulator for constrained systems. Automatica, 38(1), pp. 3–20. 2002
- [3] Bemporad, A. Hybrid Toolbox – User's Guide, <http://www.dii.unisi.it/hybrid/toolbox>. 2006.
- [4] Blanke, M., Kinnaert, M., Lunze, J. and M. Staroswiecki. Diagnosis and Fault-Tolerant Control. Springer-Verlag Berlin Heidelberg, 2003.
- [5] Bordons, C., Arce, A., del Real, A.J. Constrained Predictive Control Strategies for PEM fuel cells. Proceedings of the American Control Conference (ACC'06). Minneapolis, Minnesota, USA. 2006.
- [6] Zhang, Y. and J. Jiang. "Bibliographical review on reconfigurable fault-tolerant control systems". Proceedings of IFAC SAFEPROCESS, pages 265–276, Washington, USA, 2003.
- [7] El-Sharkh, M.Y., Rahman, A., Alam, M.S. "Neural networks-based control of active and reactive power of a stand-alone PEM fuel cell plant", Journal of Power Sources, 135, pp. 88-94. 2004.
- [8] Goodwin, G. Seron, M., and J. de Dona. Constrained control and estimation: An optimization approach. Springer-Verlag, 2005.
- [9] Larminie, J., A. Dicks, "Fuel Cell Systems Explained", John Wiley & Sons, England. 2003.
- [10] Maciejowski, J.M. Predictive Control with Constraints. Essex, Prentice Hall, 2002.
- [11] Prukushpan, J. T., Stefanopoulou, A. G., Peng, H. "Control of fuel cell Power Systems: Principles, Modeling and Analysis and Feedback Design", series Advances in Industrial Control, Springer, 2004
- [12] Pukrushpan, J., Peng, H., Stefanopoulou, A. "Control-oriented modeling and analysis for automotive fuel cell systems," ASME Journal of Dynamic Systems, Measurement and Control., vol. 126, pp. 14–25, 2004.
- [13] Qin, S., Badgwell, T., "A survey of industrial model predictive control technology," Contr. Eng. Practice, vol. 11, no. 7, pp. 733–764, July 2003.
- [14] Rawlings, J., "Tutorial overview of model predictive control," IEEE Control Systems Magazine, pp. 38–52. 2000.
- [15] Rodatz, P., G. Paganelli, A. Sciarretta, L. Guzzella. "Optimal power management of an experimental fuel cell/supercapacitor-powered hybrid vehicle," Control Engineering Practice. Vol 13, p.41-53. 2005.
- [16] Tøndel, P., Johansen, T.A., Bemporad, A. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," Automatica, vol. 39, no. 3, pp. 489–497. 2003
- [17] Vahidi, A., Stefanopoulou, A., Peng, H. "Current Management in a Hybrid Fuel Cell Power System: A Model-Predictive Control Approach". IEEE Transactions on Control Systems Technology, Vol. 14, (6), November. 2006.