FAULT-TOLERANT CONSTRAINED MPC OF PEM FUEL CELLS

Albert Rosich, Vicenç Puig, Joseba Quevedo

Automatic Control Department (ESAII) - Campus de Terrassa
Universidad Politécnica de Cataluña (UPC)
Rambla Sant Nebridi, 10. 08222 Terrassa (Spain)
vicenc.puig@upc.edu

Abstract: In this paper, fault tolerant constrained MPC control of fuel cells is presented. MPC is one of the control methodologies that can introduce more easily fault-tolerance. Here this capability is extended using new results on explicit MPC control. Explicit MPC control allows to derive off-line the control without need of using optimization. Moreover, since it is based on parametric programming allows to introduce as additional parameters faults what allow in real-time to change controller gains without the need of 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.

Keywords: Fault-tolerant control, MPC, explicit law, fuel cells.

1. INTRODUCTION

Fuel cells have developed considerable in 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 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 (Larmine, 2003), being this work devoted to PEM (Polymer Electrolyte Membrane) 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 (Prukushpan, 2004a). This study can lead to improved control strategies in this field. 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 (Prukushpan, 2004a), LQR (Prukushpan, 2004a) (Rodatz, 2005), Neural Networks (Almeida, 2005), (El-Sharkh, 2004)) or Model Predictive Control (Bordons, 2006)(Vahidi, 2006).

This paper is focused on the low level control of the fuel cell fulfil one of three main objectives: maximum efficiency, voltage control and starvation prevention. In all cases, the controller manipulates air and fuel feeding, playing with compressor voltage and hydrogen supply valve. There are other variables such as cell temperature, reactives pressures or humidity that can be included in the control strategy to improve performance. Notice that air feeding has crucial importance on fuel cell behaviour, as shown in (Pukrushpan, 2004a). Therefore, once the control objective has been chosen, it is very important to design a good control algorithm to keep the fed oxygen to its desired value. In this paper, Constrained Model Predictive Control (MPC) will be used for that purpose. However, due to a fuel 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 system under control even in the presence of faults. This paper explore the possibility of making using of the known inherent fault-tolerant capabilities of MPC control. Moreover, these capabilities are extended using new results on explicit MPC control. Explicit MPC control allows to derive off-line the control without need of using optimization. Moreover, since it is based on parametric programming allows to introduce as additional parameters faults what allow in real-time to change controller gains without the need of recomputing 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 that good results can be obtained. The remainder of paper is organized as follows: in *Section 2*, constrained MPC principles are recalled and in *Section 3*, new results on explicit MPC are briefly summarised. In *Section 4*, the inclusion of fault-tolerance in classical and explicit MPC is discussed. The behaviour of the MPC fault tolerant controller is tested on a nonlinear model of the plant and the result are shown in *Section 5*. Finally, the major conclusions are drawn in *Section 6*.

2. MPC CONTROL WITH CONSTRAINTS

2.1 Introduction

Model Predictive Control (MPC) has become the accepted standard for complex constrained multivariable control problems in the process industries. At each sampling time, starting at the current state, an open-loop optimal control problem is solved over a finite horizon N:

$$\begin{aligned} &\min_{u(0),\cdots,u(N-1)} \sum_{k=0}^{N-1} \left[x^T(k)Qx(k) + u^T(k)Ru(k) \right] \\ &\qquad + x^T(n)Px(N) \\ &\text{subject to :} \\ &x(k+1) = Ax(k) + B(u) \quad k = 0,\cdots,N-1 \\ &u(k) \in \left[\underline{u}, \overline{u} \right] \quad k = 0,\cdots,N-1 \\ &y(k) \in \left[\underline{y}, \overline{y} \right] \quad k = 0,\cdots,N \end{aligned} \tag{1}$$

At the next time step the computation is repeated starting from the new state and over a shifted horizon, leading to a moving horizon policy. The solution relies on a linear dynamic model, respects all input and output constraints, and optimizes a quadratic performance index. Thus, as much as a quadratic performance index together with various constraints can be used to express true performance objectives, the performance of MPC is excellent. 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 (Rawlings, 2000) (Qin, 2003).

2.2 MPC law computation

Constrained linear MPC is based on the solution of a quadratic program (QP) which needs to be solved to determine the optimal control action:

$$\begin{aligned} V(x(0)) &= \frac{1}{2} x^T(0)Yx(0) \\ &\quad + \min_{U} \left[\frac{1}{2} U^T H U + x^T(0)FU \right] \\ \text{subject to :} &\quad GU \le W + Sx(0) \end{aligned} \tag{2}$$

where: $U = \left[u^T(0),\cdots,u^T(N-1) \right]^T$ 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 a unique optimum is guaranteed. Additionally, efficient algorithms exists (active set and interior point methods) that allow to solve this problem very fast.

2.3 Tools for implementing MPC

The standard way of computing the MPC law, which is implemented in all commercial MPC packages, is to solve the QP problem (3) numerically on line at each time k. Commercial software tools that implement MPC can be separated into two categories: (1) tools with a proprietary real-time industrial control system (e.g., DMCplus by Aspen Technology, Inc. and RMPCT by Honeywell, Inc.) (Qin, 2003); (2) tools intended primarily for analysis and prototyping. An example of the latter is the MPC Toolbox for MATLAB (Bemporad, 2004). The MPC Toolbox allows one to program and manipulate MPC controllers as MATLAB objects through a variety of methods and functions for simulation, analysis, and tuning. Linear MPC controllers can be therefore embedded in arbitrarily complex MATLAB programs, with maximum versatility. A SIMULINK library allows the use of MPC objects in simulation models, therefore providing a large versatility in simulating the effects of MPC in complex scenarios.

3. EXPLICIT MPC CONTROL WITH CONTRAINTS

3.1 Introduction

The big drawback of constrained MPC is the on-line computational effort which may limit its applicability to relatively slow and/or small problems. In (Bemporad, 2002), 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 RHC to problems where anti-windup schemes and other ad hoc techniques dominated up to now. Such an explicit form of the controller provides also additional insight for better understanding the control policy of MPC.

There are several advantages obtained by using explicit solutions to RHC problems. The resulting explicit PWL control law allows implementation without real-time optimization software. The implementation 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. There are also some disadvantages of using explicit solutions to RHC problems 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.

3.2 Explicit law computation

The constrained finite time optimal control problems described in *Section 2* can be converted into the multiparametric Quadratic Program (*mp-QP*)

$$\begin{aligned} V(x) &= \min_{U}\left[\frac{1}{2}U^T HU + x^T F^t U\right] \\ \text{subject to}: &\quad GU \le W + Sx \end{aligned} \tag{4}$$

that must be solved for all x 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. Then, the quadratic program can be viewed as a multiparametric quadratic program (*mp-QP*). In (Bemporad, 2002), the authors analyze the properties of *mp-QP*, showing that the optimal solution is a piecewise affine function of the vector of parameters. As a consequence, the MPC controller is a piecewise affine control law 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. Nowadays, there exist very efficient *mq-QP* solvers. The solution is an explicit MPC la$u = f(x)$ that is piecewise affine (PWA) with respect to states:

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

An output feedback constrained optimal controller is obtained by computing the control law as a function of an estimate of the state vector.

An algorithm based on a geometric approach for solving *mp-QP* problems, and therefore obtain explicit RHC controllers, was proposed in (Bemporad, 2002). More recently, in (Tøndel, 2003) the authors proposed a faster algorithm based on an active-set approach.

3.3 Tools for implementing explicit MPC

The Hybrid Toolbox for MATLAB (Bemporad, 2004b) allows one to design explicit MPC control laws. The toolbox can be freely downloaded. It contains among other things various functions for the design, simulation and code generation of MPC controllers in explicit form. In particular, MPC objects developed through the MPC Toolbox can be converted to explicit form through a multiparametric quadratic programming solver based on the algorithm described in (Tøndel, 2003). The Hybrid Toolbox also provides functions for manipulation and visualization of polyhedral objects and polyhedral partitions, and contains SIMULINK blocks to simulate explicit MPC controllers.

4. INCLUDING FAULT TOLERANCE IN MPC CONTROL

4.1 Inclusion of fault tolerance in MPC

Fault-tolerant control is an incipient research area in the automatic control field (Blanke, 2003). One way of achieving fault-tolerance is to employ a fault detection and isolation (FDI) scheme on-line. This system 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. Fault-tolerance against faults can be embedded in MPC it relatively easy (Maciejowski, 2002). This can be done in two ways: (1) Redefining the constraints to represent certain kinds of faults, being this particularly appropriate for actuator fault. For example, in the case that a actuator is stuck at a given position, it can be represented in the optimization program by changing the lower and upper constraints, or if the value at which the actuator is stuck is known, inserting it as both a lower an upper constraint; (2) Changing the control objectives to reflect limitations because of the faulty conditions.

4.2 Inclusion of fault tolerance in Explicit MPC

Easy reconfiguration is traditionally considered one of the advantages of MPC, but reconfiguring an explicit solution may seems that a first glance that will need considerable off-line computation time. However, the use of parametric programming allows to express constrained optimal control problems as parametric program. This allow introducing faults as extra parameters into the parametric program:

$$u(x,f) = \begin{cases} F_1 x + g_1 & \text{if} \quad H_1 \begin{bmatrix} x \\ f \end{bmatrix} \le K_1 \\ \vdots & \\ F_{m+n_f} x + g_{m+n_f} & \text{if} \quad H_{m+n_f} \begin{bmatrix} x \\ f \end{bmatrix} \le K_{m+n_f} \end{cases} \tag{6}$$

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.

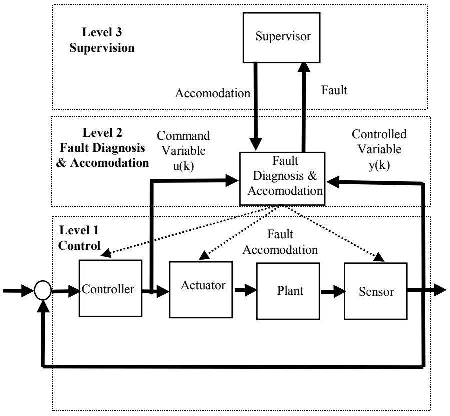

Fig. 1 Schematic diagram of the fuel cell system with auxiliary component included.

5. APPLICATION TO A TEST-BENCH FUEL CELL SYSTEM

5.1 *Fuel-cell based system description*

To test the proposed a approach a known test-bench PEM fuel cell based on the model proposed by (Pukrushpan, 2004b) 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 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 2. This can be done using several control criteria, as is described below. The main characteristics of the fuel cell used in this work are (Pukrushpan, 2004b) : Number of cells=381, Material of the membrane=Nafion 117, Active area=280cm^2, Nominal Stack Voltage =245V, Nominal Stack Current = 191A and Maximum Power= 75kW.

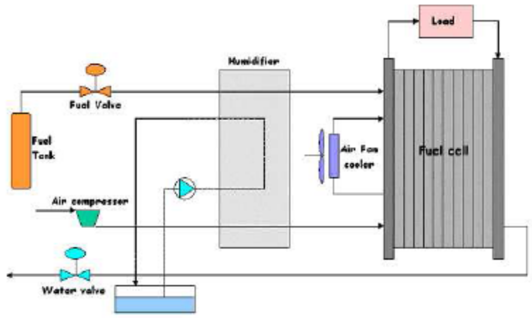

Fig. 2 Schematic diagram of the fuel cell system

The control criteria chosen is the oxygen excess ratio. This variable is used to avoid starvation phenomenon that can deteriorate or even spoil the cell. Therefore a good control performance must be achieved.

5.2 *MPC control for fuel-cell based system*

Model Predictive Control excess ratio control will be implemented using MATLAB MPC Toolbox (Bemporad, 2004a). The Fuel Cell System linear model used to implement the MPC is derived (Pukrushpan, 2004b), through a linearization at operating point: P_{net}=40kW, λ_{O2}=2 and V_{st}=235V in measured variables; I_{st}=191A in measured input disturbances; and V_{cm}=164V in manipulated variable. MPC weights are tuned to desired control goals. Following the linear model proposed by (Pukrushpan, 2004b), there are three measured outputs (Stack Net Power, Oxygen Excess Ratio and Stack Voltage), but only the oxygen excess ratio measurement is controlled by the implement MPC controller. Thus, the weight associated this variable has of a value of 10 for a good performance control, after some "trial and error" experimentation. The air compressor voltage is modelled as a constraint input due to physical limits (maximum compressor voltage cannot exceed 230V, and voltage value is never negative). The oxygen excess ratio is modelled using output constraint (the operating range is between 1.5 and 3) in order to avoid starvation. However, this last restriction can not be implemented because the electrochemical dynamics are much faster than fluid performances. It leads to the physical impossibility to prevent the drastic reduction in oxygen concentration when a step change in current occurs (Bordons 2006). This constraint can only be satisfied when auxiliary components such as batteries or ultracapacitors are used. Nevertheless, as it is shown below, the oxygen concentration transient response afterwards the first reduction, is improved by the control techniques designed. Notice that this output constraint is implemented as a soft constraint in the MPC toolbox in order to prevent the infeasible solution. Figure 3 shows the evolution of the excess ratio. A series of step changes in stack current are applied to the stack. 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.

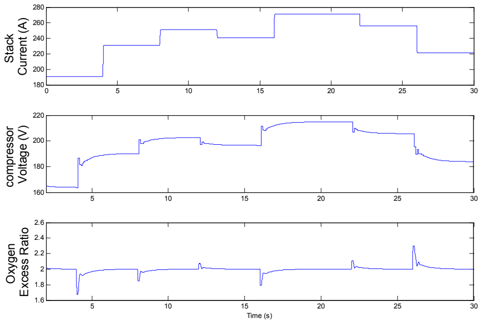

Fig. 3. Simulation results of the fuel cell system model for the oxygen excess ratio control in MPC.

5.3 Fault tolerant MPC control for fuel-cell based system

As explained in previous sections, the MPC formulation allows to easily include fault tolerant control capabilities in the control law. In this paper, faults affecting the compressor range of operation are treated. The FDI module should provide the controller the new limits of compressor voltage in every sample time. A global structure is showed in Figure 4 where the variable *LimVcp* represents the limits range actuator computed by FDI module. The FDI module (drawn in dashed line) is not implemented in this work, assuming it is available.

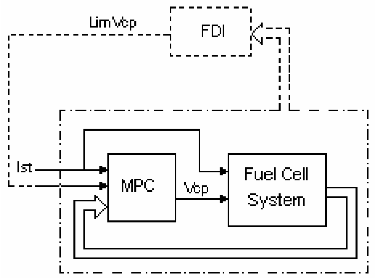

Fig. 4. Fault Tolerant MPC schema for air compressor faults.

In order to take into account changes in the actuator limit, linear model for MPC design is modified by the actuator limit as a new state for fault tolerant control:

$$\begin{bmatrix} \dot{X}_{1..N} \\ \dot{X}_{N+1} \\ \dot{X}_{N+2} \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{1..N} \\ X_{N+1} \\ X_{N+2} \end{bmatrix} + \begin{bmatrix} B \\ 0 \\ 0 \end{bmatrix} \cdot U \qquad (7)$$

$$\begin{bmatrix} Y_{1..M} \\ Y_{M+1} \\ Y_{M+2} \end{bmatrix} = \begin{bmatrix} C & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} X_{1..N} \\ X_{N+1} \\ X_{N+2} \end{bmatrix} + \begin{bmatrix} D \\ -1 \\ -1 \end{bmatrix} \cdot U$$

where A, B, C, D are the system matrices before reconfiguration and the new states are the limits of control variable. Assign to X_{N+1} the upper limit role and X_{N+2} the lower limit role, then the following new constraints in MPC controller are added: $Y_{M+1} \geq 0$ and $Y_{M+2} \leq 0$. This ensures that the controller computes the control variable U into the range specified by theoretical FDI module though Y_{M+1} and Y_{M+2} variables. Notice that from the controller view, $Y_{M+1} = X_{N+1}$ and $Y_{M+2} = X_{N+2}$, thus the unique way to keep the constrains is by modifying the control variable U.

Figures 5, 6 and 7 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 5. Dashed line represents the actuator limit that the theoretical FDI module computes. The control action is showed in Figure 5 when an actuator (air compressor) fault causes the limit range reduction of 0-75%. In this case, the control degradation is minimal as the fault does not affect the control action. In Figure 6, 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 7 the limit range is reduced 0-25%. In this case, the control goal is not achieved once the actuator fault has appeared.

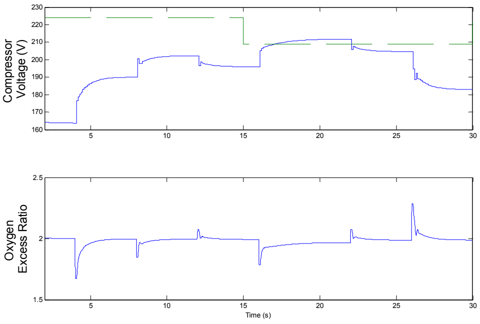

Fig. 5. Fault Tolerant MPC results in case an actuator fault that limits operating range to 0-75%.

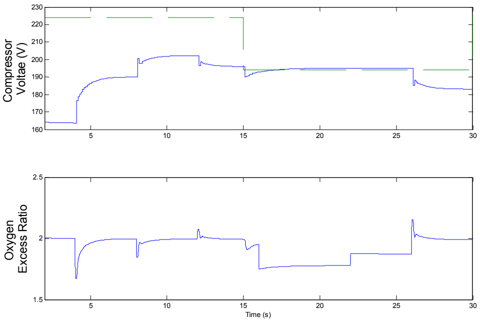

Fig. 6. Fault Tolerant MPC results in case an actuator fault that limits operating range to 0-50%.

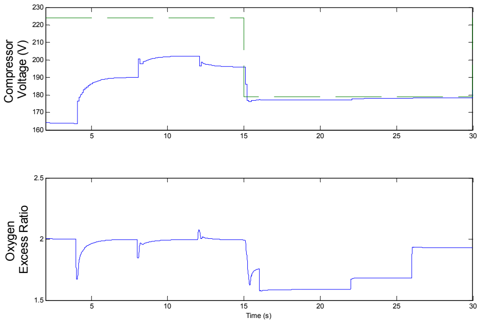

Fig. 7. Fault Tolerant MPC results in case an actuator fault that limits operating range to 0-25%.

5.3 Fault tolerant Explicit Control for fuel-cell based system

Now, Fault Tolerant MPC using Explicit MPC Control with the same properties that Classical MPC but without having to solve optimization problems on-line. The explicit controller is implemented using the Hybrid Toolbox. Extend model given by Eq. (t) is used in order to parametrisize the controller with respect to faults in actuator limits. The result is a PWA affine controller with 79 regions following the structure given by Eq. (6) In Figure 8, a projection of this PWA controller on two variables is presented: the Oxygen Excess Ratio (state variable) and the Upper Limit of the Compressor (fault variable). This allows to visualize how the controller gain changes depending on the size of the fault in the actuator.

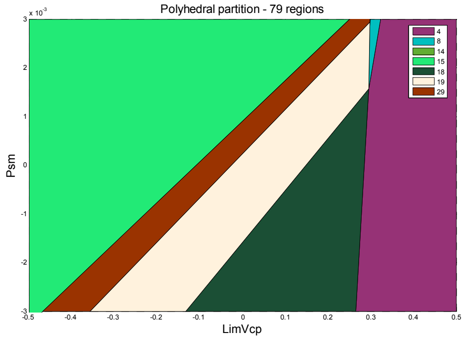

Fig. 8. Projection of PWA explicit controller on output state variable and fault variable

Simulation with explicit controller is showed in Figure 9. This simulation applies a actuator fault at time=*15s*. The model used in this case is the linear model of fuel cell system, thus the magnitudes does not mind. The last graph in this figure has special interest since shows in which region the controller is working.

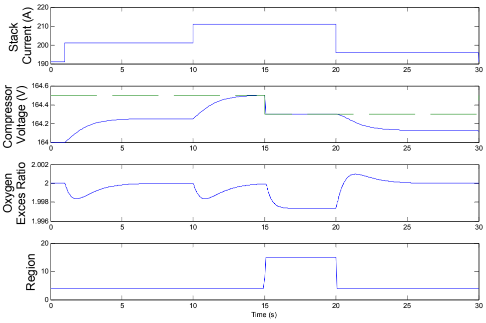

Fig. 9. Results using Fault-tolerant Explicit MPC Controller.

6. CONCLUSIONS

In this paper, fault tolerant constrained MPC control of fuel cells 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 the need of recomputing 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.

ACKNOWLEDGMENTS

The authors wish also to thank the support received by the Research Comission of the Generalitat of Catalunya (Grup SAC ref. 2005SGR00537) and by CICYT (ref. DPI-2005-05415) of Spanish Ministry of Education.

REFERENCES

Almeida, P.E.M. and M. Godoy Simoes, (2005) "Neural optimal control of PEM fuel cells with parametric CMAC networks", IEEE Transactions on Industry Applications. Volume 41 (1), pp. 237 – 245.

Bemporad, A. Morari, M., Dua, V., and E.N. Pistikopoulos, E.N. (2002). The explicit linear quadratic regulator for constrained systems. Automatica, 38(1), pp. 3–20.

Bemporad, A., Morari, M., Ricker, N.L. (2004a) Model Predictive Control Toolbox for Matlab – User's Guide. Mathworks.

Bemporad, A. (2004b) Hybrid Toolbox – User's Guide, http://www.dii.unisi.it/hybrid/toolbox.

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.

Blanke, M., Kinnaert, M., Lunze, J. and Staroswiecki, M. (2003) Diagnosis and Fault-tolerant Control, Springer.

El-Sharkh, M.Y., Rahman, A., Alam, M.S. (2004) "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.

Larmine, J., A. Dicks (2003), "Fuel Cell Systems Explained", John Wiley & Sons, England.

Maciejowski, J.M. (2002), Predictive Control with Constraints. Essex, Prentice Hall, 2002.

Prukushpan, J. T. , Stefanopoulou, A. G., Peng, H. (2004a). "Control of fuel cell Power Systems: Principles, Modeling and Analysis and Feedback Design", series Advances in Industrial Control, 2004, Springer.

Pukrushpan, J., Peng, H., Stefanopoulou, A. (2004b) "Control-oriented modeling and analysis for automotive fuel cell systems," ASME Journal of Dynamic Systems, Measurement and Control., vol. 126, pp. 14–25, 2004.

Qin, S., Badgwell, T. (2003), "A survey of industrial model predictive control technology," Contr. Eng. Practice, vol. 11, no. 7, pp. 733–764, July 2003.

Rawlings, J. (2000), "Tutorial overview of model predictive control," IEEE Control Systems Magazine, pp. 38–52.

Rodatz, P., G. Paganelli, A. Sciarretta, L. Guzzella. (2005) "Optimal power management of an experimental fuel cell/supercapacitor-powered hybrid vehicle.", Control Engineering Practice. Vol 13, p.41-53.

Tøndel, P., Johansen, T.A., Bemporad, A. (2003) "An algorithm for multi-parametric quadratic programming and explicit MPC solutions," Automatica, vol. 39, no. 3, pp. 489–497.

Vahidi, A., Stefanopoulou, A., Peng, H. (2006) "Current Management in a Hybrid Fuel Cell Power System: A Model-Predictive Control Approach". IEEE Transactions on Control Systems Technology, Vol. 14, (6), November.