Performance Evaluation of a Distributed Hybrid MPC Strategy Applied to the Four-Tank System*

Benjamín Lagos * Pablo Karelovic * Aldo Cipriano *

* College of Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile, (e-mail: balagos@uc.cl, pakarel@uc.cl, aciprian@ing.puc.cl)

Abstract: The interest in control for distributed systems has led to the development of strategies that aim to overcome issues related to high computational burden or lack of robustness. Furthermore, the search for more convenient ways of modeling MIMO systems has motivated the development of techniques for the representation and control of hybrid systems that include both continuous and discrete variables.

In this article we compare the robustness of a Distributed Hybrid MPC strategy against a Centralized MPC strategy, applied to the four-tank system. Performance was evaluated by testing random failure in sensors and actuators. Results show that for medium levels of failure, the distributed strategy has a better performance, but it requires more calculation time. The methodology proposed can be used in different applications, however it does not have assured convergence properties.

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Piece-wise affine model, distributed control, hybrid control, predictive control, fault tolerance.

1. INTRODUCTION

Large scale, spatially distributed or networked systems may be hard to control due to communication complexity, high computational burden (Negenborn and Maestre, 2014), or lack of robustness when a single centralized controller is responsible for the entire system (Zhang et al., 2014). Motivated for those issues several non-centralized control alternatives have been developed. The approach used in these schemes is to divide the system in multiple subsystems (Maestre and Negenborn, 2013) controlled by different agents independent of each other (decentralized strategy) or with communication (distributed strategy).

In turn, the search of better modelling methodologies has guided the academic community to study systems which exhibit discontinuities, events or different operating modes modelled by discrete and continuous variables. Different ways to model such kind of systems, named hybrid systems, has been proposed (Bemporad and Morari, 1999; Lunze and Lammah-Lagarrique, 2009), some of them based on identification techniques from operation data (Vasak et al., 2006; Ferrari-Trecate et al., 2001).

The goal of this work is to prove the convenience of the implementation of Distributed MPC strategies using identified Hybrid models (DHMPC, Distributed Hybrid Model Predictive Control) by studying the robustness when random fail-stop or stuck-at faults (Kim et al., 2005) occur in sensors and actuators respectively.

The study was carried out in the Four Tank System benchmark (Alvarado et al., 2011; Johansson, 2000).

Section 2 presents a brief introduction to Hybrid Models and Hybrid Model Identification. Section 3 explains the Distributed MPC strategy implemented. Afterwards, Section 4 presents how the failures are modelled and the performance index used. Then, Section 5 describes the four tank system, the identification process, the MPC strategy and the results obtained. Finally, Section 6 gives the conclusions of this work.

2. HYBRID MODELS AND HMPC

2.1 Hybrid Models

The dynamic evolution of a system is typically modelled by continuous variables, but there are some cases in which the time response is related to logic/discrete variables, e.g., if/else rules, on/off valves, batch process, etc.

One general way to model that kind of system is proposed in Bemporad and Morari (1999) by means of Mixed Logical Dynamical (MLD) systems, where logical relations are represented using equivalent inequalities between auxiliary continuous and discrete variables; and system variables. Another kind of hybrid system is the Piece-Wise Affine (PWA), which can be used to represent multiple linearizations of a nonlinear system. The operating mode is chosen according to the value of the current state-input pair in certain regions of the space of admissible values. Each mode is associated to a sub region represented by n-dimensional convex polytope. In Bemporad and Morari (1999), the equivalence between PWA and MLD models
are demonstrated, using a logic variable for each sub-region.

In this study, the MLD model was used to implement a Hybrid MPC methodology by adding the constraints defined by the MLD model to the conventional constrained MPC formulation. Using this framework, a Mixed Integer Quadratic Program (MIQP) must be solved at each control instant, which can be performed by an optimization solver.

2.2 Hybrid Identification

A hybrid model can be identified from input-output data, as is described in Vasak et al. (2006) and Ferrari-Trecate et al. (2001), even for systems that do not have hybrid behaviour but its non-linear nature can be represented using models with multiple operating modes.

Assuming a data set of a MIMO (Multiple Inputs Multiple Output) system given by \( (y(k), u(k)) \), where \( u(k) \in \mathbb{R}^m \) and \( y(k) \in \mathbb{R}^q \), the regressor is defined as:

\[
\begin{align*}
\mathbf{r}(k) &= [y(k-1)^T \ y(k-2)^T \ ... \ y(k-n_a)^T \ u(k-1)^T \ u(k-2)^T \ ... \ u(k-n_b)^T]^T \\
\mathbf{u}(k) &= [\mathbf{y}(k) \ \mathbf{r}(k)]^T \\
\end{align*}
\]

where \( n_a \) and \( n_b \) correspond to the model orders.

Using an identification algorithm (Vasak et al., 2006; Ferrari-Trecate et al., 2001), model parameters for each operating mode are determined. Additionally, the convex polytopes that represent the region associated with each operating mode are also calculated. The resulting identified PWARX model is:

\[
y(k) = \begin{cases} 
\Theta_1^T \mathbf{r}(k) + \mathbf{f}_1 & \text{if } \mathbf{r}(k) \in \mathcal{R}_1 \\
\vdots & \\
\Theta_s^T \mathbf{r}(k) + \mathbf{f}_s & \text{if } \mathbf{r}(k) \in \mathcal{R}_s 
\end{cases}
\]

where \( \Theta_i \) is a matrix containing the model parameters for mode \( i \), and regions \( \mathcal{R}_s \) form a complete partition of the admissible set \( \mathcal{R} \), where \( s \) is the number of regions.

By reordering the parameters obtained in the identification for each output it is possible to find a state space representation for each region, which is called a PWA model:

\[
x(k+1) = \mathbf{A}_i \mathbf{x}(k) + \mathbf{B}_i \mathbf{u}(k) + \mathbf{f}_i \\
y(k) = \mathbf{C}_i \mathbf{x}(k) + \mathbf{g}_i, \quad \text{for } \mathbf{x}(k) \in \mathcal{R}_i
\]

The PWA model is then translated into an equivalent MLD model. The general structure of a MLD system is:

\[
x(k+1) = \mathbf{A} \mathbf{x}(k) + \mathbf{B}_1 \mathbf{u}(k) + \mathbf{B}_2 \mathbf{\delta}(k) + \mathbf{B}_3 \mathbf{z}(k) \\
y(k) = \mathbf{C} \mathbf{x}(k) + \mathbf{D}_1 \mathbf{u}(k) + \mathbf{D}_2 \mathbf{\delta}(k) + \mathbf{D}_3 \mathbf{z}(k) \\
\mathbf{E}_2 \mathbf{\delta}(k) + \mathbf{E}_3 \mathbf{z}(k) \leq \mathbf{E}_1 \mathbf{u}(k) + \mathbf{E}_4 \mathbf{x}(k) + \mathbf{E}_5 
\]

where \( \mathbf{\delta}(k) \) and \( \mathbf{z}(k) \) are auxiliary variables which are binary and continuous respectively. \( \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4 \) and \( \mathbf{E}_5 \) are matrices which contain the information to transform the logical relations between the variables to inequalities.

The selection of the region of the PWA model is performed by the inequalities of the MLD model, which is suitable to be included in the MIQP problem.

3. FUNDAMENTALS OF DISTRIBUTED MPC (DMPC)

One option for controlling a MIMO system is to divide the problem in multiple subsystems, each one of them controlled by an agent, an entity capable of making decisions using the information received by sensing the environment or communicating with other agents.

If each subsystem is controlled independently from each other (i.e., no communication or cooperation between agents) it’s called a decentralized solution. On the other hand, if coupling between subsystems and communication among agents is considered, it’s called a distributed strategy (Negenborn and Maestre, 2014).

Recently, different Distributed MPC alternatives have been proposed (Maestre and Negenborn, 2013; Bemporad, 2009; Furina and Ferrari-Trecate, 2012; Negenborn et al., 2009; Scattolini, 2009), classified according to its topology (partial or complete), information exchange (iterative or non-iterative) and rationality (cooperative or non-cooperative).

For this work, the methodology described in Stewart et al. (2010) was selected, that is an Iterative Cooperative DMPC strategy. The optimal control problem considers communication between agents and the objective function is global.

Assuming that the system has at least 2 inputs and 2 outputs, it can be divided in \( M \) subsystems each one associated to certain number of inputs and outputs, in such a way that they don’t belong to different subsystems at the same time.

In the selected strategy, at each time step \( k \), each agent solves an optimization problem with all available information (current state and control sequence over a control horizon \( N_c \), communicated by the other agents), minimizing a global objective function, and obtaining the control sequence for its associated subsystem, while considering a constant value for the control sequence of the other agents. Once solved, the solutions are communicated to other agents, weighted (\( \omega_i \)) and averaged with an estimation of it, and used for each agent in the next iteration as a constant value for the control sequence of the rest of the agents. That process is repeated until a maximum number of iterations or a convergence criterion is reached and the first element of the control sequence is applied to the plant, according to the receding horizon concept.

This scheme tends to the centralized solution when increasing the number of iterations.

Details about the implementation could be found in Stewart et al. (2010) and Ferramosca (2013).

4. PERFORMANCE EVALUATION

4.1 Robustness Analysis

Fail-stop faults in sensors and stuck-at faults in actuators (Kim et al., 2005) are applied to the plant. Fail-stop sensor faults \( f_1^s(k) \) were included when the states of the system are estimated and are modelled as a value equal to zero when the failure occurs. Actuators faults \( f_2^a(k) \)
are applied directly to the plant and represent the case where the actuator does not upgrade its control action and keeps applying the last action given by the controller \(u(k) = u(k-1)\).

Given a time step and a total simulation time, there are a total of \(N\) simulation iterations. The random occurrence of faults was introduced by defining the percentage \(%\text{Fault}\), which is equal to the ratio of iterations that present a failure \(n_{\text{fault}}\) to the total number of iterations \(N\). The iterations that present failures were distributed randomly.

\[
\%\text{Fault}_{\text{actuator}} = \frac{n_{\text{actuator}}}{N}, \quad \%\text{Fault}_{\text{sensor}} = \frac{n_{\text{sensor}}}{N}
\]

\[
N \sum_{k=1}^{N} f_i^g(k) = n_{\text{actuator}}, \quad N \sum_{k=1}^{N} f_i^s(k) = n_{\text{sensor}} \quad i = \{1, \ldots, q\}, \quad j = \{1, \ldots, m\}
\]

Three different percentages (0%, 40% and 80%) were considered for \(%\text{Fault}_{\text{sensor}}\), while four (0%, 25%, 50% and 75%) were considered for \(%\text{Fault}_{\text{actuator}}\). All the possible combinations give twelve different fault cases for this study.

4.2 Calculation Time

In order to evaluate the time required for the controllers to find a solution at each control step, the following criteria was implemented.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>MPC</td>
<td>Calculation time for the single agent</td>
</tr>
<tr>
<td>HMPC</td>
<td>Calculation time for the single agent</td>
</tr>
<tr>
<td>DHMPC</td>
<td>For each agent, the time after the last negotiation iteration. Then, the maximum time between agents.</td>
</tr>
</tbody>
</table>

Mean calculation time per time step is used for comparison. Maximum calculation time is used to check the feasibility of the proposed controller (the calculation time should be less than the control sampling period).

4.3 Performance Index

The performance index used for this case is the same used in Alvarado et al. (2011) which includes the reference tracking error and the difference between the input applied and the steady-state input to be applied in order to reach the reference value.

\[
J = \sum_{k=1}^{N} \left( \sum_{i=1}^{q} \alpha_i \left(y_i - y_i^{\text{ref}}\right)^2 + \sum_{j=1}^{m} \beta_j \left(u_j - u_j^*\right)^2 \right)
\]

Parameters \(\alpha_i\) and \(\beta_j\) are suitable weights, chosen to make the different inputs and outputs comparable between them.

5. APPLICATION TO THE FOUR TANK SYSTEM

5.1 Description of the Process

The four tank system, proposed as a benchmark in Johansson (2000), has been used for several works in model predictive control because of its versatility, easy implementation and high coupling between its variables.

The inputs \((u_1, u_2)\) correspond to the flow rate \([\text{m}^3/\text{h}]\) given by two pumps \((q_1, q_2)\) and the outputs \((y_1, y_2)\) correspond to the height in tanks 1 and 2 \((h_1, h_2)\). Additional states of the system are the heights of tanks 3 and 4 \((h_3, h_4)\).

The equations that describe the dynamics of the process are explained comprehensively in Alvarado et al. (2011).

5.2 Identification of a PWA model

A typical input signal used for identification is the Pseudo-Random Binary Sequence (PRBS) (Yun-Tao et al., 2012). For this work, the identification was made using the input sequence, and its corresponding output, showed in Figure 1 inspired in the identification realized in Putz (2014) and obtained by simulation of the system.

![Input-Output data for identification](image)

Fig. 1. Input-Output data used for the identification.

The input data was determined from the constraints. In several moments the states of the system violate the constraints; however the information of the dynamics offered by this collection of data is more important. The constraints were considered in the simulation of the model and in the controller.

From the observation of the plant (Alvarado et al., 2011) it is possible to establish some relations between variables. The identification process considered the following relations:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Regressor</th>
</tr>
</thead>
<tbody>
<tr>
<td>(h_1)</td>
<td>([h_1 h_2 q_1 q_2])</td>
</tr>
<tr>
<td>(h_2)</td>
<td>([h_1 h_2 q_1 q_2])</td>
</tr>
<tr>
<td>(h_3)</td>
<td>([h_3 q_2])</td>
</tr>
<tr>
<td>(h_4)</td>
<td>([h_4 q_1])</td>
</tr>
</tbody>
</table>

Identification was performed using the Hybrid Identification Toolbox (HIT) (Ferrari-Trecate, 2005) with the parameters of Table 3.

Parameter \(c\) is associated to the clustering algorithm. Just one model was identified for \(h_3\) and \(h_4\) in order to reduce the complexity of the problem.

Figure 2 shows the time response when a PRBS is applied as input signal. This validation considers the model constraints in inputs and outputs.

The model identified for \([h_1, h_2]\) shows good agreement with the simulated output. On the other hand, the models...
Table 3. Hybrid identification parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$h_1$</th>
<th>$h_2$</th>
<th>$h_3$</th>
<th>$h_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n_a$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$n_b$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$s$</td>
<td>6</td>
<td>5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$c$</td>
<td>300</td>
<td>300</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>HIT Algorithm</td>
<td>Proximal Support Vector Classifiers (PSVC)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length of data vector</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
</tr>
</tbody>
</table>

Fig. 2. Validation of PWA model.

identified for $[h_3, h_4]$ are not as good as the previous ones but are enough to keep the levels between its constraints, as will be seen in the following section. The model was simulated previously without considering the constraints but the response of the model for $h_3$ and $h_4$ was not satisfactory. There are zones where the mode of operation oscillates between 2 modes. To overcome this issue, the tolerance of the inequality that selects the operating mode was increased.

Each pair of regions ($C_1, C_2$) was intersected according to the procedure described in Section 2.2. In this case, the intersection of the constraints for the regressor space entails the elimination of modes giving 16 modes of operation, less than the 30 modes expected. That result could be observed in Figure 2 where only modes $\{1,3,4,5\}$ and $\{1,2,4,5\}$ appear, for $h_1$ and $h_2$ respectively. Additionally, a 12 modes model, that considers the modes $\{1,3,4,5\}$ and $\{1,4,5\}$ for $h_1$ and $h_2$ respectively, will be evaluated, in order to reduce calculation time.

5.3 MPC Strategy

A conventional MPC strategy was developed, solving a Quadratic Program (QP) at each control instant using a quadratic objective function which minimize the error between the outputs and a reference and minimize the variations of the control action in a linear model subject to the system constraints.

The prediction model used for this controller is one of the linear models identified previously, associated with the operating point defined by levels $[h_1^*, h_2^*] = [0.65, 0.65]$.

5.4 Simulation Results and Analysis

Simulation test were performed over a simulation time of 200 minutes on an Intel Xeon E3-1240 v3 processor. The parameters used for the controllers are displayed in Table 4, where $\omega_1$ and $\omega_2$ are the weights used when the solutions are communicated to other agents. The performance index (6) was computed in the time range between 45 and 200 minutes, with $\alpha_1 = \alpha_2 = 1$ and $\beta_1 = \beta_2 = 0.01$.

Some representative simulation tests are displayed in Figure 3 through Figure 5. The figures also include the reference values for $h_1$ and $h_2$, and the corresponding steady-state input values $q_1$ and $q_2$.

Table 4. Controller parameters.

<table>
<thead>
<tr>
<th></th>
<th>MPC</th>
<th>HMPC</th>
<th>DHMPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction/control horizon N</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Weights for reference tracking error $[\omega_1 \omega_2]$</td>
<td>$[1 \ 1]$</td>
<td>$[1 \ 1]$</td>
<td>$[1 \ 1]$</td>
</tr>
<tr>
<td>Weights for variation in control action</td>
<td>$2 \times 10^7 [1 \ 1]$</td>
<td>$2 \times 10^7 [1 \ 1]$</td>
<td>$2 \times 10^7 [1 \ 1]$</td>
</tr>
<tr>
<td>Coefficients for optimal solution $[\omega_1 \omega_2]$</td>
<td>N/A</td>
<td>N/A</td>
<td>$[0.3 \ 0.7]$</td>
</tr>
<tr>
<td>Number of iterations</td>
<td>N/A</td>
<td>N/A</td>
<td>8</td>
</tr>
<tr>
<td>Controller time step [s]</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>References (every 50 minutes) [m]</td>
<td>$[h_1^* h_2^*]$</td>
<td>$[0.65 \ 0.30 \ 0.50 \ 0.90]$</td>
<td>$[0.65 \ 0.30 \ 0.75 \ 0.75]$</td>
</tr>
</tbody>
</table>

The Root-Mean-Square (RMS) error, performance index $J$ and both mean and maximum computation time are displayed in Figure 6 for the PWA model with 16 modes of operation, and in Figure 7 for the PWA model with 12 modes of operation.

The base case displayed in Figure 3 (no faults) shows the effectivity of the designed controllers. The controllers manage to drive outputs to the reference values while keeping all variables within the predefined operating ranges. A similar response can be seen in Figure 4, with a medium failure level.

Fig. 3. Case 1, no failures.
The most extreme case with the highest failure level, displayed in Figure 5, shows that the conventional MPC strategy has trouble keeping the variable $h_4$ within its desired operating range. It can also be seen in this figure that at some instants, the values of the manipulated variables fail to change even though there are reference changes for the controlled variables. This behaviour can be seen in every case with 80% chance of sensor error (cases 3, 6, 9 and 12), which happen to be the cases with the worst performance.

The explanation for this event is that the controllers are unable to find a solution for the optimization problem, so the optimal solution of the previous time step is repeated. This may be a consequence of the structure of the MLD representation used in this article, which increases the amount of constraints in the optimization problem as more subsystems are considered, making the optimization problem unfeasible. Evidence of this is the fact that the conventional MPC, which does not use MLD models, manages to find a solution in these cases. Furthermore, the computation time increases when more modes of operation are considered, as seen in Figure 6 and Figure 7. This problem may be addressed by using more efficient techniques for the computation of the MPC control problem with MLD models (Wang and Zhao, 2009; Thomas et al., 2006).

Regarding the error in reference tracking and the performance index used, it can be seen that the DHMPC strategy exhibits the best performance in the cases with no faults or a medium level of sensor faults. Nonetheless, its performance worsens in comparison to the conventional MPC strategy with increasing fault levels.

Even though the DHMPC strategy exhibits a better performance for lower fault levels, the computation time for this scheme is higher. For the PWA model with 16 modes of operation, the mean computation time does not exceed the sample time. Nonetheless, for some reference changes the computation time may exceed the sample time. To address this issue, the PWA model with 12 modes of operations was implemented, which considerably lowered computation times. A different way of lowering computation time is by performing less iterations for the computation of the individual agents solution. A drawback of this approach is that the optimal solution will worsen as less iterations are performed. Despite this problem, the inclusion of a MLD model results in a better performance when compared to a conventional MPC strategy, due to the better representation of the systems dynamics.

A final observation is the impact of the chosen values for the coefficients $\omega_1 (0.3)$ and $\omega_2 (0.7)$. The higher value for the solution calculated by agent 2 was chosen by
simulating different combinations until the performance index was minimized.

The results obtained support the hypothesis that distributed solutions are more robust than centralized solutions in the presence of faults. On the other hand, it is expected that distributed solutions also lower the required computation time, which is not seen in this study. This is a result of the iterative nature of the chosen strategy, combined with the low amount of manipulated variables, which lowers the impact of distributing the computation of the optimal solution between different agents.

Finally, the methodology used in this article can be used in different applications since the distributed strategy only uses the calculated values for the manipulated variables so the model used and the solution of the control problem at each iteration is independent. However, the convergence properties of the DMPC strategy applied wasn’t implemented and needs to be extended for the case of hybrid systems in order to assure its robustness.

6. CONCLUSION

This paper presents the study of the performance of a DHMPC strategy in presence of faults in sensors and actuators. A PWA model, identified from simulation data, was used for the development of the controllers.

The performance of the proposed solution was compared to a centralized HMPC strategy and a conventional MPC strategy. The DHMPC exhibits a better performance for low to medium fault levels. Additionally, sensor faults have a higher impact in performance than actuator faults.

As a trade-off for the increase of robustness, the DHMPC strategy requires higher computation times. This behaviour is greatly influenced by the low amount of variables in the studied system, as mentioned previously.

The MLD representation used for the proposed strategy does not guarantee the convergence of the distributed optimal solution to the centralized optimal solution. This topic will be addressed in future studies.

Future work will focus on applying the methodology to real processes, such as mineral froth flotation (Putz, 2014).

REFERENCES


