2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Evaluation of Constrained Multivariable EPSAC Predictive Control Methodologies*

Robinson Medina [1,2], Andres Hernandez[1,2], Clara Ionescu [1] and Robin De Keyser[1]

Abstract— **This paper investigates the performance of two Multivariable Model Predictive Control (MPC) strategies: selfish and solidary. These strategies are based on the main ideas developed in the EPSAC (Extended Prediction Self-Adaptive Control) approach to MPC. A two degree of freedom (2DOF) helicopter simulation has been chosen to illustrate these concepts, as it represents a complicated and challenging problem due to strong intercoupling effects, non-linear dynamics and uncertainties in the system model. The performance obtained with a Linear Quadratic Regulator (LQR) is also included as a reference to the performance of the multivariable MPC strategies. In this contribution, the performance of these multivariable MPC strategies is analyzed, providing more insight about the behavior of these controllers.**

I. Introduction

Model Predictive Control (MPC) is a general designation for controllers that make explicit use of a model of the plant to obtain the control signal by minimizing an objective function over a time horizon. This strategy allows to deal with multivariable and non-linear processes, as well as unusual behavior of processes. Some examples of MPC strategies described in [1] are: Generalized Predictive Control (GPC), Dynamic Matrix Control (DMC), Model Algorithmic Control (MAC) and Extended Prediction Self-Adaptive Control (EPSAC).

During the last decades, MPC has become an important, distinctive part of control theory and application. A great interest has been shown for this methodology resulting in many interesting reviews and books [2], [3], [4], [5]. The reason for its success can be attributed to the fact that Model Predictive Control is the most general way of posing the process control problem in the time domain. MPC formulation integrates optimal control, stochastic control, control of processes with dead time and multivariable control to mention a few ones.

A particularly challenging and interesting process to test the performance of MPC is a Helicopter. This process presents severe nonlinearities and significant cross-coupling between its inputs and outputs, which make the control of such multiple-input multiple-output (MIMO) system a challenging task.

Conventional approaches to helicopter flight control involve linearization about an equilibrium point, allowing the use of single-input single-output (SISO) techniques [6], [7]. The main drawback of this approach is that interactions of the MIMO system decreases the controller performance as it assumes a decoupled system, besides the fact that these multiloop controllers are inflexible and difficult to tune. Therefore, the MIMO controller design approaches have received more and more attention. Successful implementations of Linear Quadratic Regulator (LQR) and Sliding Mode Control are presented in [8] and [9], respectively. Constrained MPC is presented in [10] in which a twin rotor MIMO system was used.

In this study two alternatives to calculate the optimal control solution for MIMO MPC are evaluated: solidary and selfish control [2]. Initially, the effectiveness of these techniques is probed, by comparing the performance of the solidary MPC with another multivariable strategy, namely the Linear Quadratic Regulator (LQR). During this first test, the controllers are tuned to produce the fastest response without overshoot and then the control effort is just constrained into the minimal and maximal possible value of the actuator (this intuitive strategy to deal with input constraints will be referred as 'clipping'). Next, a more convenient formulation to deal with constraints is done, by including the constraints in the multivariable MPC control problem 'constrained control'. Simulation experiments in a strongly coupled 2DOF helicopter shows that the main differences between selfish and solidary control arise at the moment that the constraints are active, providing insight in the understanding of their behavior.

The content of this paper is as follows. Section II presents in detail the methodology for Multivariable MPC, with 2 design solutions based on different control criteria: *solidary* and *selfish* control. Next in section II-C constraints are included in the control problem. A comparison between the 2 proposed multivariable MPC methodologies and a LQR is presented in section III. A conclusion section summarizes the main outcome of this investigation.

II. Multivariable EPSAC Method

This section briefly summarizes the extension of the EPSAC predictive control to multivariable systems, both 'solidary' and 'selfish' approaches [2]:

A. Principle of MIMO EPSAC

The analysis of the method is described considering the case of a n_u inputs and n_y outputs process (Fig. 1):

$$y_i(t) = x_i(t) + n_i(t),\ i = 1,2,\dots,n_y \tag{1}$$

where,

*This work has been obtained as results of the bilateral agreement between Ghent University and Universidad de Ibague, in the framework of the Master in Control Engineering

[1] Department of Electrical energy, Systems & Automation, Ghent University, Belgium. Andres.Hernandez@UGent.be

[2] Department of Automation and Industrial Control, D+TEC research group, Universidad de Ibague, Colombia.

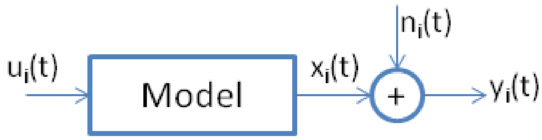

Fig. 1. Process Model

- $y_i(t)$:(measured) process outputs
- $u_j(t)$: process intputs
- $x_i(t)$: model outputs
- $n_i(t)$: process/model disturbances

An essential aspect in the MPC methodology consists of the prediction of the process outputs, these are calculated with:

$$y_i(t+k|t) = x_i(t+k|t) + n_i(t+k|t), \; i = 1,2,\ldots,n_y \tag{2}$$

for $k = N_{1i},\ldots,N_{2i}$ where N_{1i} and N_{2i} are the minimum and the maximum prediction horizons for each *i-output* of the process. Our problem resides now on finding $x_i(t+k|t)$ and $n_i(t+k|t)$. The first multi-step prediction problem is solved by recursion of the process models, while the second is solved using filtering techniques on the noise model. A detailed description is given in [2].

The future response of the process is considered to be the result of two effects:

$$y_i(t+k|t) = y_{ibase}(t+k|t) + y_{iopt}(t+k|t), \; i = 1,2,\ldots,n_y \tag{3}$$

The two contributions have the following origins:

$\boxed{y_{ibase}(t+k|t):}$

- effect of past controls and of the basic future control scenario, called $u_{jbase}(t+k|t)$, for $k = 0\ldots,N_{uj}-1$ (N_u being the control horizon), and for $j = 1\ldots n_u$. For linear systems the choice of this scenario is irrelevant, a simple choice being $u_{jbase}(t+k|t) \equiv 0, k \geq 0$
- effect of future (predicted) disturbances $n_i(t+k|t)$.

.

$\boxed{y_{iopt}(t+k|t):}$

- effect of the optimizing future control actions: $\delta u_j(t+k|t) = u_j(t+k|t) - u_{jbase}(t+k|t)$,$k = 0\ldots N_u - 1$. Where $u_j(t+k|t)$ are the desired optimal control actions. The optimizing control actions δu_j can be considered as a series of impulses h^{ij} and a final step g^{ij} of input j to output i.
- the prediction horizons N_{2i} could be different for the n_y *outputs*.
- the control horizons N_{uj} could be different for the n_u *inputs*.

In brief the **key EPSAC-MPC equations** for the MIMO case (3), can be expressed in matrix notation:

$$\mathbf{Y_i} = \mathbf{Y_{iBase}} + \mathbf{Y_{iopt}} = \mathbf{\overline{Y}_i} + \sum_{\mathbf{j=1}}^{\mathbf{n_u}} \mathbf{G_{ij}U_j} \tag{4}$$

where for $i = 1,2,\ldots,n_y$ and $j = 1,2,\ldots,n_u$:

$$\mathbf{Y_i} = [y_i(t+N_{1i}|t)\ldots y_i(t+N_{2i}|t)]^T$$
$$\mathbf{\overline{Y}_i} = [y_{ibase}(t+N_{1i}|t)\ldots y_{ibase}(t+N_{2i}|t)]^T$$
$$\mathbf{U_j} = [\delta u_j(t|t)\ldots \delta u_j(t+N_{uj}-1|t)]^T$$
$$\mathbf{G_{ij}} = \begin{bmatrix} h^{ij}_{N_{1i}} & h^{ij}_{N_{1i}-1} & \ldots & h^{ij}_{N_{1i}-N_{uj}+2} & g^{ij}_{N_{1i}-N_{uj}+1} \\ h^{ij}_{N_{1i}+1} & h^{ij}_{N_{1i}} & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ h^{ij}_{N_{2i}} & h^{ij}_{N_{2i}-1} & \ldots & h^{ij}_{N_{2i}-N_{uj}+2} & g^{11}_{N_{2i}-N_{uj}+1} \end{bmatrix}$$

B. The Control Objective

For MIMO-systems, the calculation of the optimal control vector $\mathbf{U_j}$ can be approached in 2 different ways, depending on the choice of the control criteria. Without affecting the generality of the problem will be considered in the sequel a system with 2 inputs and 2 outputs.

Solidary Control

The objective is to find the optimal control vectors $\mathbf{U_1^*}$ and $\mathbf{U_2^*}$ which minimize the cost function

$$\begin{aligned} J(\mathbf{U}) = & \sum_{k=N_{11}}^{N_{21}} [r_1(t+k|t) - y_1(t+k|t)]^2 + \\ & \sum_{k=N_{12}}^{N_{22}} [r_2(t+k|t) - y_2(t+k|t)]^2 \end{aligned} \tag{5}$$

suject to $u_1(t+k|t) = u_1(t+N_{u1}-1|t)$ for $k \geq N_{u1}$ and $u_2(t+k|t) = u_2(t+N_{u2}-1|t)$ for $k \geq N_{u2}$.

With this strategy the predicted control errors *summed over all process outputs* are minimized. Notice that the control error for a specific variable y_1 can possibly and *deliberately* be increased, with the pupose of reducing the control error for another variable y_2. The objective is thus to minimize the *total* control error of *all* partners together, and not just to minimize the individual control error of each partner separately; hence, the choice for the name *solidary* control.

Defining compound matrices $\mathbf{G_1} = [\; \mathbf{G_{11}} \;\; \mathbf{G_{12}} \;]$ and $\mathbf{G_2} = [\; \mathbf{G_{21}} \;\; \mathbf{G_{22}} \;]$ and the compound vector $\mathbf{U} = [\; \mathbf{U_1} \;\; \mathbf{U_2} \;]^T$, it is possible to represent (5) in a quadratic cost index in $\mathbf{U}$:

$$J(\mathbf{U}) = \mathbf{U^T H U} + \mathbf{2f^T U} + c \tag{6}$$

with,

$$\begin{aligned} &\mathbf{H} = \mathbf{G_1^T G_1} \quad \mathbf{f} = [-\mathbf{G_1^T}(\mathbf{R_1} - \mathbf{\overline{Y}_1}) + \mathbf{G_2^T}(\mathbf{R_2} - \mathbf{\overline{Y}_2})] \\ &c = (\mathbf{R_1} - \mathbf{\overline{Y}_1})^{\mathbf{T}}(\mathbf{R_1} - \mathbf{\overline{Y}_1}) + (\mathbf{R_2} - \mathbf{\overline{Y}_2})^{\mathbf{T}}(\mathbf{R_2} - \mathbf{\overline{Y}_2}) \end{aligned} \tag{7}$$

Selfish Control

In this case the objective is to find the optimal control vectors $\mathbf{U_1^*}$ and $\mathbf{U_2^*}$ which minimize the cost function:

$$
\begin{aligned}
J(\mathbf{U_1}) &= \sum_{k=N_{11}}^{N_{21}} [r_1(t+k|t) - y_1(t+k|t)]^2 \\
J(\mathbf{U_2}) &= \sum_{k=N_{12}}^{N_{22}} [r_2(t+k|t) - y_2(t+k|t)]^2
\end{aligned}
\tag{8}
$$

subject to $u_1(t+k|t) = u_1(t+N_{u1}-1|t)$ for $k \geq N_{u1}$, and $u_2(t+k|t) = u_2(t+N_{u2}-1|t)$ for $k \geq N_{u2}$.

Although at first sight the above mentioned objectives (8) might give the (false) impression of a degenerate MIMO controller - consisting of 2 independent SISO controllers, it is important to realize that is certainly not the case. Contrary to the solidary control strategy, the objective of this selfish control strategy is not to minimize the total control error of *all partners together*, but just to minimize the individual control error of *each player separately*. However, taking into account *the effect of control actions of all other - possibly competing - players*.

This makes the strategy a *multivariable* control approach, with internal cross compensation of the dynamic interactions in the multivariable process. In fact, in the unconstrained case, experience shows that both solidary and selfish lead to nearly identical control performance.

Using again the compound matrix notation, the cost functions of selfish control can be written as a quadratic form in $\mathbf{U_1}$ and $\mathbf{U_2}$:

$$
\begin{aligned}
J_1(\mathbf{U_1}) &= \mathbf{U_1^T H_1 U_1} + \mathbf{2f_1^T U_1} + c_1 \\
J_2(\mathbf{U_2}) &= \mathbf{U_2^T H_2 U_1} + \mathbf{2f_2^T U_2} + c_2
\end{aligned}
\tag{9}
$$

with,

$$
\begin{aligned}
\mathbf{H_1} = \mathbf{G_{11}^T G_{11}} \quad \mathbf{f_1} &= -\mathbf{G_{11}^T}(\mathbf{R_1} - \mathbf{\overline{Y}_1} - \mathbf{G_{12}U_2}) \\
c_1 = (\mathbf{R_1} - \mathbf{\overline{Y}_1} - \mathbf{G_{12}U_2})^{\mathbf{T}} &(\mathbf{R_1} - \mathbf{\overline{Y}_1} - \mathbf{G_{12}U_2})
\end{aligned}
\tag{10}
$$

$\mathbf{H_2}, \mathbf{f_2}$ and $\mathbf{c_2}$ can be easily defined similarly to (10).

C. Constrained MIMO EPSAC

In practice all processes are subject to constraints, because actuators have a limited range of action and a maximum slew rate. Fortunately, Model Predictive Control offers a straightforward approach to deal with constraints, making of it a desirable strategy to be applied in industry. For the case of limits in the actuators range (*input constraints*), two approaches are available: *clipping* (leading to suboptimal results and being usual approach, e.g. in PID control) and *constrained control* (leading to optimal results and particulary being one of the main advantages of MPC).

Clipping is the simplest approach as the control is calculated assuming the actuator has unlimited range. Once the action is calculated, it is then hard-limited to keep the resulting values within the specified range.

Minimizing $J(\mathbf{U})$ for both *selfish* and *solidary* controllers with respect to $\mathbf{U}$, leads to the optimal (unconstrained) solution:

$$
\mathbf{U_i^*} = -\mathbf{H_i^{-1} f_i}, \quad for\ i = 1, 2. \tag{11}
$$

Clipping approach will take the unconstrained solution (11) into a minimum and maximum value allowed, e.g. ($min \leq U_i^* \leq max$).

In **Constrained control**, constraints are taken into account a priori, thus leading to the best solution that is possible within the specified limits. In MPC, the calculation of these constrained control actions is approached as a constrained optimization problem:

$$
\begin{aligned}
\min_{\mathbf{U}} \quad & J(\mathbf{U}) = \mathbf{U^T H U} + \mathbf{2f^T U} + c \\
\text{subject to} \quad & \mathbf{AU} \leq \mathbf{b}
\end{aligned}
\tag{12}
$$

with $\mathbf{A}$ a specified matrix and $\mathbf{b}$ a specified vector (both depending on the type of constraints).

Above problem is a standard, well-known optimization problem called quadratic programming (quadratic cost function with linear inequality constraints). The differences between the two constrained approaches will be addressed in next section through a simulation example, in which at the same time the conceptual differences of *solidary* and *selfish* control are also highlighted.

III. Mechatronic Application

Although the MIMO EPSAC algorithm can be applied to any multivariable process, we will focus on the field of mechatronic systems, by applying it to a 2 DOF Quanser helicopter (Fig. 2).

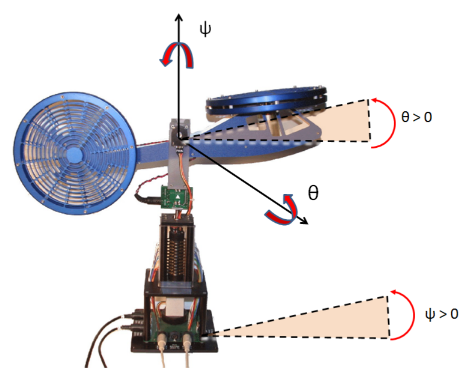

Fig. 2. Quanser 2 DOF Helicopter

A. Process Description

While the description of the helicopter is beyond the scope of this paper, a brief description of it is provided in this subsection; for further information about it, the original documentation in [8] is recommended. The Quanser 2 DOF Helicopter experiment consists of a helicopter model mounted on a fixed base with two propellers that are driven by DC motors. The front propeller controls the elevation of the helicopter nose about the pitch axis and the back propeller controls the side to side motions of the helicopter about the yaw axis. The pitch and yaw angles are measured using high-resolution encoders [8].

The two degrees of freedom helicopter pivots about the pitch axis by angle θ and about the yaw axis by angle ψ. The pitch is defined positive when the nose of the helicopter goes up and the yaw is defined positive for a clockwise rotation.

B. Modelling

Using the kinematics of the center of mass it is possible to find the potential and kinetic energies involved in the helicopter system, and using the Euler-Lagrange method to derive the nonlinear equations of motion [8]. Two nonlinear differential equations of second order are obtained; the values of lengths, masses and moment of inertias associated with the helicopter model are presented in the table I.

$$\ddot{\theta}(J_{eqp}+m_{heli}l_{cm}^2) = K_{pp}V_{mp}+K_{py}V_{my}-B_p\dot{\theta} \\ -m_{heli}.l_{cm}^2\dot{\psi}^2\cos\theta\sin\theta - m_{heli}.g.l_{cm}\cos\theta \tag{13}$$

$$\ddot{\psi}(J_{eqy}+m_{heli}l_{cm}^2\cos\theta^2) = K_{yp}V_{mp}+K_{yy}V_{my}-B_y\dot{\psi} \\ +2m_{heli}.l_{cm}^2\cos\theta\sin\theta\dot{\psi}\dot{\theta} \tag{14}$$

TABLE I

NOTATION AND UNITS USED IN THE HELICOPTER MODEL

Symbol	*Description*	*Value*	*Unit*
J_{eqp}	Total moment of inertia in θ	0.0384	Kg.m^2
J_{epy}	Total moment of inertia in ψ	0.0432	Kg.m^2
m_{heli}	Total moving mass of the helicopter	1.3872	Kg
l_{cm}	Center of mass about pitch axis	0.186	cm
K_{pp}	Thrust torque on θ from pitch motor	0.204	N.m/V
K_{py}	Thrust torque on θ from yaw motor	0.0068	N.m/V
K_{yp}	Thrust torque on ψ from pitch motor	0.0219	N.m/V
K_{yy}	Thrust torque on ψ from yaw motor	0.072	N.m/V
V_{mp}	Voltage applied to pitch motor	$[0,20]$	V
V_{my}	Voltage applied to yaw motor	$[-10,10]$	V
B_y	Equivalent viscous damping about ψ	0.318	N/V
B_p	Equivalent viscous damping about θ	0.8	N/V
g	Gravitational acceleration constant	9.81	m/s^2

The linear model of the process represented in transfer functions is presented in 15. This model is obtained by linearization of equations (13) and (14) around the operating point described by the outputs ($\theta=0$, $\psi=0$) and the inputs ($V_{mp}=18.05$, $V_{my}=-7.5$). A suitable sampling time, able to capture the dynamics of the system is $T_s=0.01$ s.

$$\begin{bmatrix} \theta(s) \\ \psi(s) \end{bmatrix} = \begin{bmatrix} \frac{1.657}{s(s+9.275)} & \frac{0.07898}{s(s+9.275)} \\ \frac{0.2542}{s(s+3.496)} & \frac{0.6121}{s(s+3.496)} \end{bmatrix} \begin{bmatrix} V_{mp}(s) \\ V_{my}(s) \end{bmatrix} \tag{15}$$

From the matrix transfer function (15) it is possible to observe that voltage in pitch motor V_{mp} has strong influence on pitch angle θ, while voltage in yaw motor V_{my} has strong influence in yaw angle ψ. Moreover, a positive voltage applied to V_{mp} generates an increment of pitch angle, but also a positive change in the yaw angle. A similar conclusion can be stated for voltage V_{my}, although, the cross-coupling effect in this case is smaller but not negligible.

C. Clipping: LQR vs. Solidary EPSAC

It is important to highlight that all controllers were tested using as plant the nonlinear model of the helicopter (13),(14).

The first simulation consists in comparing the performance of a Linear Quadratic Regulator (LQR) and solidary EPSAC controller, both using clipping as strategy to deal with constraints given in Table I. LQR was designed including integral action to avoid steady state error, according to the cost function:

$$J=\int_0^\infty (X^TQX+U^TRU)dt \tag{16}$$

where the penalizing matrices Q and R were tuned to achieve the fastest response possible without overshoot giving as results:

$$Q = 1000*diag(100,100,1,1,400,500) \\ R = diag(0.05,0.05) \tag{17}$$

Following the same approach as for LQR (fastest response without overshoot), the unconstrained EPSAC solidary controller was tuned, giving as result: $N_{11}=N_{12}=1$, $N_{u1}=N_{u2}=1$ and $N_{21}=N_{22}=10$ for $T_s=0.01$ s.

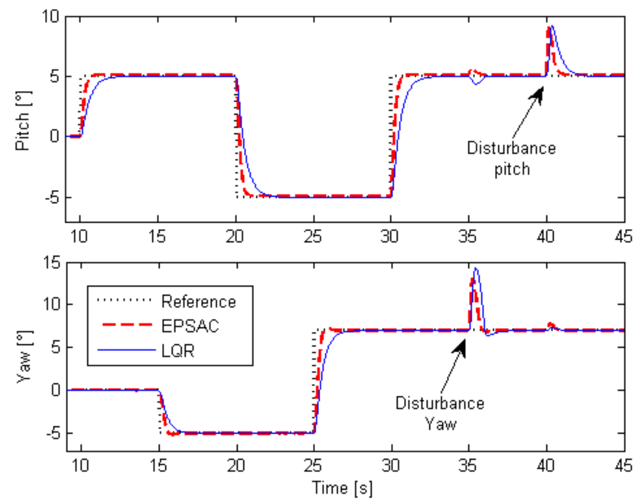

Fig. 3. LQR vs Clipping EPSAC

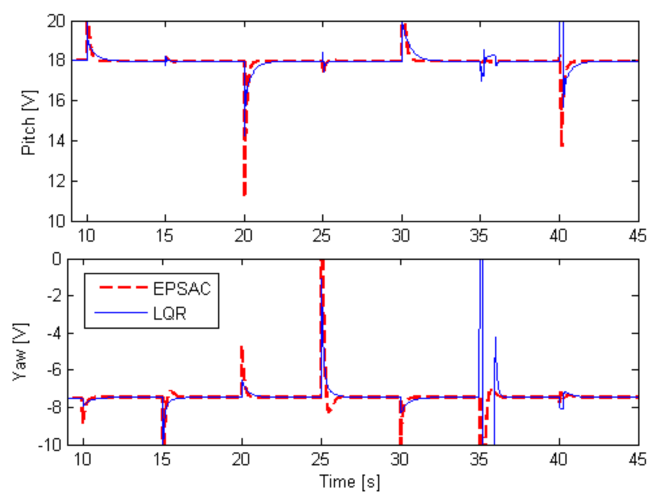

Fig. 4. Control effort LQR vs Clipping EPSAC

Another important aspect to tune the predictive controller is the noise filter as illustrated in [11]. A 'smart' choice for these filters being: $\frac{1}{(1-q^{-1})(1-q^{-1})}$ for pitch and $\frac{1}{(1-q^{-1})}$ for yaw, as it leads to have zero steady-state error

The comparison between LQR and the clipping solidary EPSAC is depicted in Fig. 3, in which it is observed how the MPC outperforms the LQR both in terms of overshoot and settling time. MPC presents a more agressive control action compared to the LQR (Fig. 4).

In Fig. 3 it can be noticed how applying a step change in the reference or applying a disturbance to one of the axis produce a deviation in the other one. More specifically, at time 25 seconds a change in the reference is applied to yaw axis, the MPC controller applies a big control action compared to the LQR, as result the system has a smaller settling time. More interesting is to observe that pitch angle did not change significantly, owing to the fact that the influence from pitch to yaw is not so strong. Furthermore, the MPC compensates the effect of yaw on pitch by decreasing the voltage in pitch, thus dynamically compensating the interactions. Opposite to the MPC behavior, the LQR applies a smaller control action in yaw and almost no action in pitch, resulting in a slower response to track the yaw reference and a small overshoot in pitch.

At time 35 seconds a disturbance is applied to yaw axis, to wich the MPC presents a better disturbance rejection. The explanation is again that it better compensates for the dynamic interactions, meaning in this particular case to avoid decreasing the voltage in pitch as this has a big influence in yaw. As observed, in the control actions made by the LQR it decreases the voltage in both motors producing an undesired behavior over pitch, meanwhile the MPC did not apply any voltage in pitch motor, thus minimizing the interaction effect.

Even though clipping EPSAC has exhibit a satisfactory performance, it can be improved by considering input constraints in the MPC formulation as evaluated in next section.

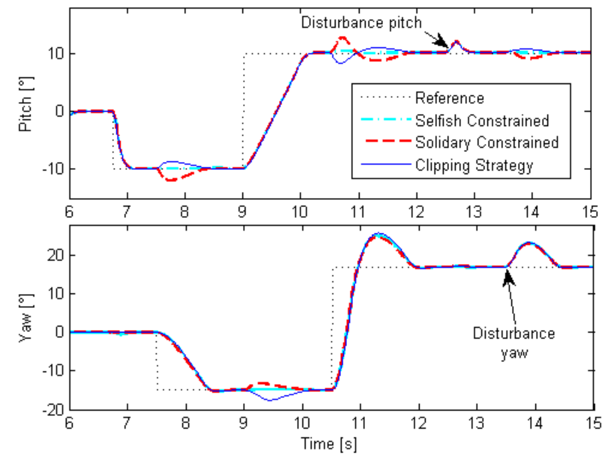

Fig. 5. Constrained Selfish and Solidary vs Clipping EPSAC

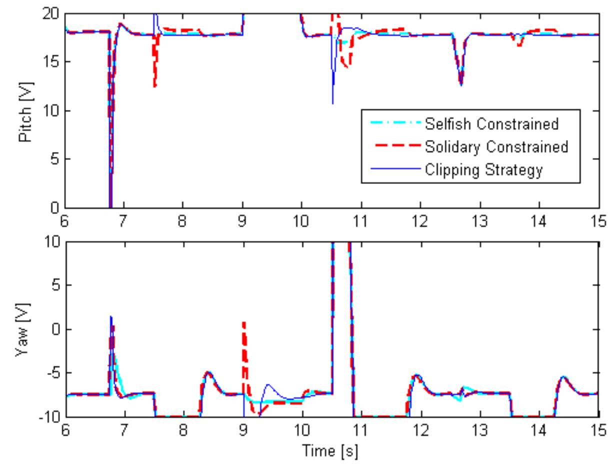

Fig. 6. Control effort constrained vs Clipping EPSAC

D. *Constrained Control: Selfish vs. Solidary*

Although, originally the helicopter has a larger operation range, during this study we have constrained it even more to illustrate the effectiveness of MPC to deal with input constraints. The operating input range was thus fixed to: $0 \leq V_{mp} \leq 20$ and $-10 \leq V_{my} \leq 10$. Aditionally, in order to highlight the differences between clipping and constrained MPC, the simulation time was reduced and bigger changes in the reference were applied compared to the LQR simulation.

Fig. 5 shows the multivariable *clipping* strategy (the same analized above) compared to both selfish and solidary *constrained* strategies. Observing in detail, it is noticeable how each time there is a big change in the setpoint in one axis, the selfish strategy compensates the cross interaction such that the opposite axis remains as close as possible to the setpoint (as expected from theory!). On the other hand, the solidary strategy tries to use the interaction to achieve a faster response even at expenses of creating a bigger error in the opposite axis to which the change on the setpoint was applied. In Fig. 6 it is observed the control effort for this particular experiment.

The mentioned effect can be clearer seen on Fig. 7, which corresponds to a zoom of Fig. 5 at time 10.5 seconds when a step change was applied to yaw. As result, the following analysis for the control strategies is shown:

- The *clipping* EPSAC asks for an unachievable value of $V_{my} = 120$ V (Fig. 8). Therefore, the controller will try to compensate for the possible increment in the pitch angle due to the interaction decreasing the V_{mp} to about 11 V. Finally, the voltage applied to yaw is not 120 V as request by the controller but restricted to 10 V, consequently, pitch decreases as it was expecting a bigger interaction.
- In the *constrained selfish* strategy, the controller output in the yaw axis is within the constraints of the physical system, as consequence, the interaction is well compensated avoiding significant changes in the pitch angle. This statement is in agreement with theory as constrained selfish controller will try to keep the error in pitch as low as possible, i.e. it will act as a good decoupler.
- In the *constrained solidary* strategy, the controller output is also within the constraints of the system. However, in this case, the controller produces a positive change in pitch in order to help the controller to reach

the setpoint in yaw faster (a positive change in pitch generates a positive change in yaw angle).

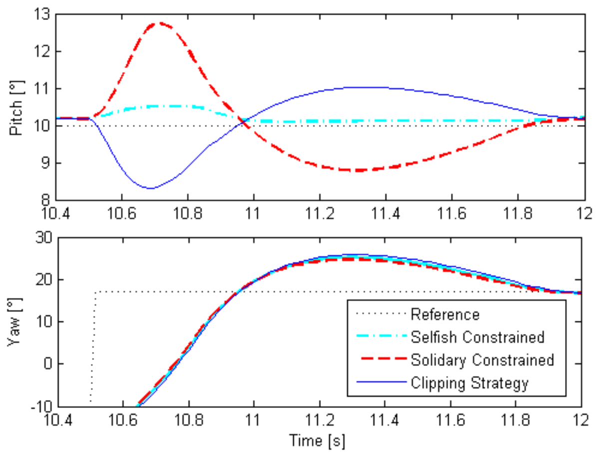

Fig. 7. Zoom Constrained vs Clipping EPSAC

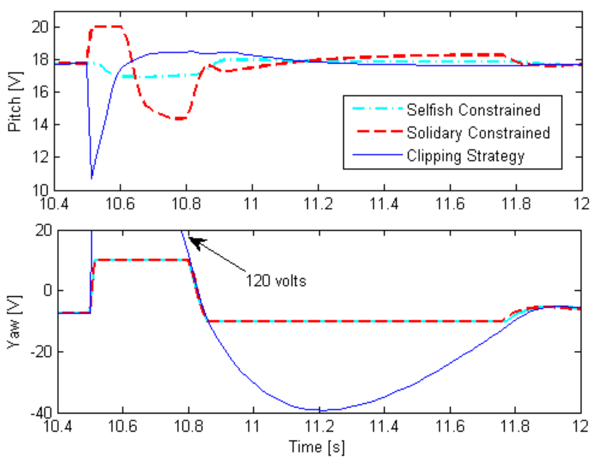

Fig. 8. Zoom Control effort Constrained vs Clipping EPSAC

The performance of the controllers is compared (in the time range of the zoomed area) by using the well known performance indexes *ISE*, *IAE* and *ITAE* as shown in table II. It is possible to observe that the best performance is achieved by the constrained controllers, but specially the constrained selfish. The main reason for this result is that it does not deviate significantly in pitch while still keeps a similar settling time and overshoot in yaw compared to the solidary constrained.

TABLE II
PERFORMANCE INDEX FOR MIMO EPSAC STRATEGIES

Controller	*ISE*	*IAE*	*ITAE*
Solidary Clipping	4.9713	17.4560	191.3585
Selfish Constrained	4.7258	15.7619	172.2839
Solidary Constrained	4.5815	16.3843	179.0239

IV. CONCLUSIONS

Two multivariable predictive control strategies were presented and evaluated by means of a 2 DOF helicopter case. It has been illustrated that the main differences of these controllers appear at the moment the constraints are active. The results obtained suggest that depending on the control target, following setpoint changes as fast as possible (tracking) or avoiding that the change of one reference affects the other (decoupling), one might choose the solidary or selfish control respectively. However, if the cross-coupling effect between the inputs and outputs is not strong enough, using a solidary MPC for tracking might not be meaningful. Instead, a selfish MPC control should be used, as it follows equally well the reference changes without disturbing other outputs, as observed in the studied case of this paper.

ACKNOWLEDGMENT

This study was performed under the basis of an agreement of cooperation between the Universidad de Ibague (Colombia) and Universiteit Gent (Belgium).

The results presented in this paper have been obtained within the frame of the IWT SBO-110006 project The Next Generation Organic Rankine Cycles (www.orcnext.be), funded by the Institute for the Promotion and Innovation by Science and Technology in Flanders. This financial support is gratefully acknowledged.

REFERENCES

[1] Camacho, E.F., and Bordons, C., *Model Predictive Control*. ISBN-13:978-1852336943, Springer, 2nd Ed., 2007.

[2] De Keyser R., (2003). Model Based Predictive Control. *Invited Chapter in UNESCO Encyclopaedia of Life Support Systems (EoLSS)*. Article contribution 6.43.16.1, Eolss Publishers Co Ltd, Oxford, ISBN 0 9542 989 18-26-34 (www.eolss.net), 30p.

[3] Qin, S.J., and Badgwell, T.A., A survey of Industrial model predictive control technology. *Control engineering practice*, 11(7), 733-764, 2003

[4] Garcia, C.E., Prett, D.M., and Morari, M.,Model predictive control: Theory and Practice a survey, *Automatica*, 25, no. 3, pp. 1753-1758, 1989.

[5] Maciejowski, J.M., Predictive control with constraints, *Prentice hall*, 2002.

[6] Garcia R.A., Rubio F.R., and Ortega M.G., "Robust PID Control of the Quadrotor Helicopter", In IFAC Conference on Advances in PID Control PID'12 Brescia (Italy), March 28-30, 2012

[7] Velagic J., and Osmic N., 'Identification and Control of 2DOF Nonlinear Helicopter Model Using Intelligent Methods", *IEEE Inter. Conf. on Systems, Man and Cybernetics (SMC 2010)*, Istanbul, TURKEY, pp., 2267 - 2275, 2010

[8] Quanser, '2-DOF Helicopter reference manual - onlne version", *Quanser Consulting Inc.*, 2006.

[9] G.R. Yu, and H.T. Liu, 'Sliding mode control of a two-degree-of-freedom helicopter via linear quadratic regulator", *IEEE Int. conf. on Systems, Man and Cybernetics 4*, pp.3299-3304, 2005.

[10] Rahideh A., and Shaheed M.H., 'Constrained output feedback model predictive control for nonlinear systems", *J. Control engineering practice* 20, pp.413-443, 2012.

[11] De Keyser R., and Ionescu C., "The disturbance Model in Model Based Predictive Control", in *Proc. of the IEEE conference on Control applications*, Istanbul, Turkey, CD paper CF-001472, 6p, 2003.