2014 International Conference on Unmanned Aircraft Systems (ICUAS)
May 27-30, 2014. Orlando, FL, USA

Automatic Landing of a UAV using Model Predictive Control for the surveillance of Internal Autopilot's Controls

Juliano A. de Bonfim Gripp[1], Ulisses P. Sampaio[2]

***Abstract*— Recent advancements in processing capabilities have enabled the possibility of use of new computationally costly control methods like the Model Predictive Control (MPC) on real-time applications. On this paper, a framework is proposed for the design of a supervising MPC to be implemented over a simpler, more reliable internal loop PID Autopilot without making any modifications to its structure. The proposed system was able to add new capabilities to the internal loop by making little corrections to its control demands, ensuring the compliance with restrictions. At the end, the system was able to ensure safe landing over many demanding conditions by effectively expanding the internal autopilots states and by enforcing altitude restrictions.**

I. Introduction

Model Predictive Control (MPC) has become an attractive strategy to control systems, finding wide application in several fields, from industrial processes to dynamic moving systems. One of the main advantages of MPC is the possibility to consider explicitly system's physical and operational constraints during the design of the control loop. This property makes it useful for the operation of physically constrained systems as it allows the enforcement of hard limits on variables. If the model adopted for the plant is linear and the constraints involve upper and lower limits for the states and controls, the control signal to be applied to the plant at a given moment involves the solution of a Quadratic Programming problem, for which efficient algorithms exist, as described in Maciejowski [1]. This process is usually repeated every sampling instant, based on information acquired from sensors, and can employ a strategy of receding horizon and terminal condition.

This paper presents an overview of a control system used as proof of concept for automatic landing of a UAV, using a predictive controller in a supervisory loop to actuate and protect a supervised loop of constraints violation.

The supervised loop consists of the aircraft plant with its actuators and a linear longitudinal autopilot for the tracking of the flight-path angle γ and the total forward velocity V_T. The designed MPC controller analyzes system variables (inputs, outputs, control signals and states) and predicts its behavior some iterations ahead (prediction horizon N) to ensure that the supervised loop avoids the violation of constraints . Physical limits of actuators and the landing altitude are examples of monitored variables by MPC controller.

*This work was not supported by any organization

[1]Juliano A. de Bonfim Gripp, Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil juliano.gripp@gmail.com

[2]Ulisses P. Sampaio, Instituto Tecnológico de Aeronáutica, São José dos Campos, Brazil ulisses.sampaio@gmail.com

The assembly explored in this paper (Figure 1) has two great advantages: it adds the possibility to impose constraints in the internal loop's envelope (like making sure the aircraft's altitude stays higher or equal to the runway's level); and enables the insertion of a terminal output condition to be achieved (the aircraft must land at the runway's altitude) at a finite number of steps in the future. This architecture does not require any modifications of the supervised loop's configuration but inserts new means of correcting its control signal and gives it advantageous new capabilities. Furthermore, the MPC will remain inactive as long as the aircraft is not close to breaking any constraints or it has not engaged the landing mode (landing mode will be activated when the UAV has an altitude close to the runway's altitude).

The Figure 1 presents a high level diagram of the proposed system. The control loop to be supervised by the MPC is inside dashed red box. Based on reference vector (r), plant's measured *Outputs* and the internal states of longitudinal autopilot (*AP states*), the MPC provides a contribution (v) to the control signal (u) resultant from AP when constraints are about to be violated. Otherwise, the MPC provides no contribution to control ($v = 0$), having only a constraints supervisor role.

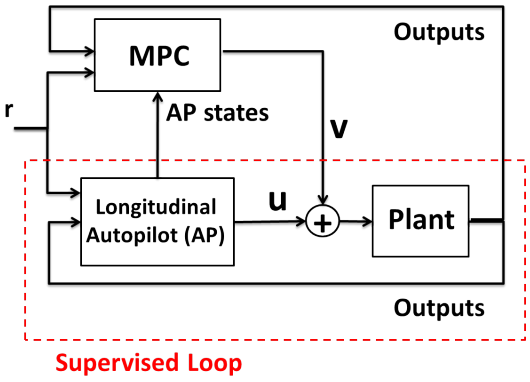

Fig. 1. Overview of MPC and supervising loop.

This paper is organized as follows: Section II presents an overview of the supervised loop (dashed red box in Figure 1); Section III describes the predictive controller (MPC block in Figure 1) and the approach adopted to apply it as a surveillance controller. Finally, Section IV summarizes the main results obtained from this design by comparing performance of the system in landing mode for different conditions using the predictive controller.

978-1-4799-2376-2/14/$31.00 ©2014 IEEE

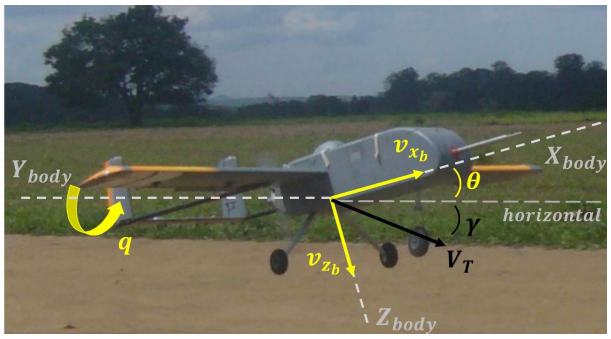

Fig. 2. Vector-P UAV and its variables of interest

II. Supervised Loop: Plant, Actuators and Longitudinal Autopilot

Vector-P UAV, the plant of this design (Figure 2), is an Unmanned Aerial Vehicle (UAV) designed by Intellitech Microsystems. This aircraft has been used at Technological Institute of Aeronautics (ITA) for research and development purposes. Santos and Goes [2] describe details about aircraft system identification and rigid-body dynamics.

From the aircraft's longitudinal non-linear dynamics a linear model can be obtained and represented in a matrix form. Let the plant linear dynamics be described by Equation 1, with state vector x_p and applied control vector u. Additionally, a measured plant output vector y_p and a *performance output* vector z can be defined.

$$\dot{x}_p = A_p x_p + B_p u \quad y_p = C_p x_p \quad z = H_p x_p \qquad (1)$$

In this case, matrices A_p and B_p represent the linear dynamics of the longitudinal state vector $x_p = [v_{x_b}\ v_{z_b}\ q\ \theta\ \delta_t\ \delta_e]^T$ with respect to the insertion of the control vector $u = [u_t\ u_e]^T$. The state variables v_{x_b} and v_{z_b} are the aircraft's velocity components in body-fixed axis, θ is the pitch angle and q is the pitch rate, as seen in Figure 2. δ_t is the applied throttle [%] and δ_e is the elevator deflection.

Simplified dynamics of actuators are considered (Equations 2), with u_t and u_e representing throttle and elevator commands to be inserted by the longitudinal autopilot, with constraints: $0.1 < \delta_t < 1$ and $-15° < \delta_e < 15°$.

$$\delta_t = \left(\frac{4}{s+4}\right) u_t \qquad \delta_e = \left(\frac{10}{s+10}\right) u_e \qquad (2)$$

A. Controller Dynamics

For satisfactory reference tracking, the automatic flight control system must have the following features:

- Output feedback to promote stability augmentation into loop. This design presents a Pitch Damper as Stability Augmentation System (*SAS*) by using q and θ in feedback, thus $y_p = [q\ \ \theta]^T$ is selected;
- Dynamic Controllers for variable tracking, by using Proportional-Integral (PI) and *lead-lag* compensators. V_T and γ tracking was chosen (reference vector $r = [V_{T_c}\ \ \gamma_c]^T$) using $z = [V_T\ \ \gamma]^T$.

The design of tracking control system follows the approach of Stevens and Lewis [3]. This structure includes a outer loop that feeds the performance output z back and subtracts it from the reference command r. It defines a tracker error e which should be kept small (Figure 3).

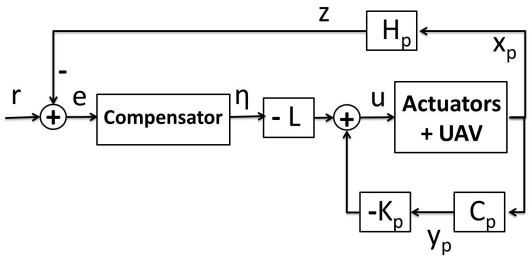

Fig. 3. Plant with compensator of desired structure - Tracker Design.

The dynamic compensator in Figure 3 has the form presented by Equations 3.

$$\dot{\omega} = F_c\omega + Ge \quad \eta = D\omega + Je \quad e = r - z \qquad (3)$$

The compensator's chosen structure is shown in Figure 4.

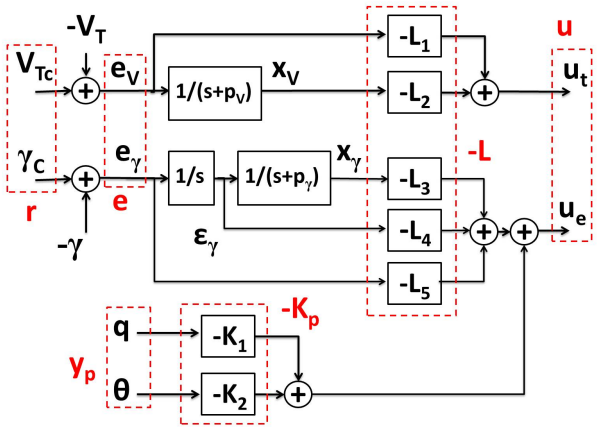

Fig. 4. Compensator structure with its gains (V_T and γ tracking) and pitch damper SAS.

State vector (ω), output (η) and tracking error (e) related to the compensator may be expressed by Equations 4, where subscripts are used for variables related to *commanded* value ("c"), total forward velocity ("V") and flight path angle ("γ").

$$\begin{aligned} \omega &= [x_V\ \ x_\gamma\ \ \epsilon_\gamma]^T \quad \eta = [x_V\ \ e_V\ \ x_\gamma\ \ \epsilon_\gamma\ \ e_\gamma]^T \\ e &= [e_V\ \ e_\gamma]^T \quad r = [V_{T_c}\ \ \gamma_c]^T \quad z = [V_T\ \ \gamma]^T \end{aligned} \qquad (4)$$

From aircraft and compensator dynamics, using the configuration presented in Figure 3, control signal assumes the format $u = -K_p y_p - L\eta$, i. e., control signal has a feedback component $-L\eta$ responsible for output tracking and other component $-K_p y_p$ to stabilize the system (*SAS*).

Matrices F_c, G, D, J are directly obtained from structure in Figure 4 regarding to variables in Equation 4. Moreover,

the design parameters of F_c are chosen as $p_V = 3/T$ and $p_\gamma = 7/T$, where T is the sampling time used for system discretization.

B. Complete Dynamics of Actuator and Plant

Complete dynamics of system may be written as presented in Equation 5:

$$\begin{aligned}
\underbrace{\begin{pmatrix}\dot{x_p}\\ \dot{\omega}\end{pmatrix}}_{\dot{x}_s} &= \underbrace{\begin{pmatrix}A_p & 0\\ -GH_p & F_c\end{pmatrix}}_{A_s}\underbrace{\begin{pmatrix}x_p\\ \omega\end{pmatrix}}_{x_s} + \underbrace{\begin{pmatrix}B_p\\ 0\end{pmatrix}}_{B_s}u + \underbrace{\begin{pmatrix}0\\ G\end{pmatrix}}_{E}r\\
\underbrace{\begin{pmatrix}y_p\\ \eta\end{pmatrix}}_{y} &= \underbrace{\begin{pmatrix}C_p & 0\\ -JH_p & D\end{pmatrix}}_{C_s}\underbrace{\begin{pmatrix}x_p\\ \omega\end{pmatrix}}_{x_s} + \underbrace{\begin{pmatrix}0\\ J\end{pmatrix}}_{F}r\\
z &= \underbrace{[H_p\ \ 0]}_{H_s}\underbrace{\begin{pmatrix}x_p\\ \omega\end{pmatrix}}_{x_s} \qquad u = -\underbrace{[K_p\ \ L]}_{K}\underbrace{\begin{pmatrix}y_p\\ \eta\end{pmatrix}}_{y}
\end{aligned} \tag{5}$$

for

$$K = \begin{pmatrix}0 & 0 & L_2 & L_1 & 0 & 0 & 0\\ K_1 & K_2 & 0 & 0 & L_3 & L_4 & L_5\end{pmatrix} \tag{6}$$

Vectors and matrices of Equation 5 may be written as follows (redefining matrices):

$$\begin{aligned}
\dot{x}_s &= A_s x_s + B_s u + Er \qquad & y &= C_s x_s + Fr\\
z &= H_s x_s & u &= -Ky
\end{aligned} \tag{7}$$

The resultant dynamics with closed-loop state-space matrices A_c and B_c is given by Equation 8. This equation represents the linear dynamics of the supervised loop inside the dashed red box in Figure 1.

$$\dot{x}_s = \underbrace{(A_s - B_s K C_s)}_{A_c} x_s + \underbrace{(E - B_s K F)}_{B_c} r \tag{8}$$

Therefore, considering expanded dynamics of the plant with its compensator, the state of the feedback system can be rewritten as:

$$x_s = [v_{x_b}\ v_{z_b}\ q\ \theta\ \delta_t\ \delta_e\ x_V\ x_\gamma\ e_\gamma]^T \tag{9}$$

Also, matrices A_c, B_c and C_s can be used to obtain the steady-state values of the vectors x_s, y and u for a given reference. Those vectors are obtained with Equation 10 and will later be used as the reference vectors for the MPC construction:

$$\begin{cases}
x_{ref} &= -(A_c^{-1}B_c)r\\
y_{ref} &= (F - C_s A_c^{-1} B_c)r\\
u_{ref} &= -K y_{ref}
\end{cases} \tag{10}$$

C. Gain Matrix Synthesis

The method employed for synthesis of the K controller gain involves Linear Quadratic Control. This is an advantageous approach since it allows direct synthesis of dynamic controllers, avoiding individual loop closing.

Given a cost function J (Performance Index), the problem of designing a system based on state feedback controller becomes to determine the gain K that minimizes such Performance Index.

The considered cost-function must penalize tracking error, in both transient and steady-state. In order to guarantee observability of (H_s, A_s), tracking error is weighted in this design by t^2. Thus, the optimal cost-function is computed by a chain of Lyapunov's equations, as described by Stevens and Lewis [3]. The format of the chosen cost function is shown in Equation 11, where $Q = {H_s}^T H_s$ (positive semi-definite symmetric matrix) penalizes states, R (positive definite symmetric matrix) penalizes control effort, matrix W_e penalizes tracking error, and e_{ss} is steady-state tracking error.

$$J = \frac{1}{2}\left(\int_0^\infty (t^2 x^T Q x + x^T C^T K^T R K C x)dt + e_{ss}^T W_e e_{ss}\right) \tag{11}$$

III. MPC Concept and Loop Surveillance

Model Predictive Control (MPC) is a general designation to a set of sophisticated and efficient discrete control methods that analyzes the optimum control that must be applied to certain system at each iteration while predicting system response at many sample times ahead. The MPC can check if the system will provide a desired outcome and, if not, can act preemptively to change it. The proposed design in this paper uses the current output/state measurement, the plant's linear model and performance requirements to enforce variables constraints (states, controls, or outputs). The MPC can also impose terminal conditions to system's variables, making the system's outputs/states reach desired values at the end of a given number of iterations.

A. Expanded State

The MPC must be designed to ensure a safe landing. It must be noted that the longitudinal autopilot alone is not able to meet this requirement because it does not monitor aircraft's altitude and cannot deal with hard constraints by itself.

Besides enforcing the constraints on the state vector $x_p = [v_{x_b}\ v_{z_b}\ q\ \theta\ \delta_t\ \delta_e]^T$, the MPC must be able to impose altitude constraints. For satisfactory actuation, the controller must predict with good accuracy how the altitude (h) is affected by the dynamics of aircraft, actuators and autopilot.

Then, for the MPC design, taking into account the aircraft's altitude constraints, the UAV's open loop dynamics can be linearized considering the new augmented state vector $x_a = [v_{x_b}\ v_{z_b}\ q\ \theta\ x_D\ \delta_t\ \delta_e]^T$, where $x_D = -h$ represents vertical position in the $North - East - Down$ coordinate

system. An augmented version of Equation 1 is obtained with new A_a, B_a, C_a and H_a matrices (obtained by numeric linearization of the non-linear space-state):

$$\dot{x}_a = A_a x_a + B_a u \quad y_p = C_a x_a \quad z = H_a x_a \tag{12}$$

It must be noted that the augmented A_a, B_a, C_a and H_a matrices can be used to obtain the matrices A_p, B_p, C_p and H_p from last section: just remove the rows and columns that are related to x_D (the fifth state of x_a).

For the prediction of the future states, the MPC also needs to take into account the autopilot's internal states x_V, x_γ and e_γ. Considering the already established dynamics of the supervised loop, the same F_c, G, D, J, K from last section are used with A_a, B_a, C_a, H_a on Equations 5 to obtain a new closed loop representation:

$$\begin{aligned} \dot{x} &= Ax + Bu + Er \qquad y = Cx + Fr \\ z &= Hx \qquad\qquad\quad u = -Ky \end{aligned} \tag{13}$$

Thus, the state vector to be supervised by MPC is:

$$x = [v_{x_b}\ v_{z_b}\ q\ \theta\ \ x_D\ \ \delta_t\ \delta_e\ x_V\ x_\gamma\ e_\gamma]^T \tag{14}$$

It can be noted that the new x vector, despite having x_D as an internal state, describes exactly the same dynamics as the vector x_s (Equation 9). Figure 5 summarizes how each state vector is obtained.

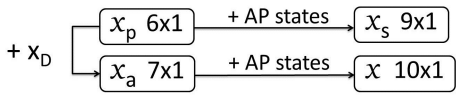

Fig. 5. State vector augmentation.

B. Steady-State Deviation

The state vector x describes the complete aircraft's linear dynamics taking into consideration the autopilot's closed loop. This system can be re-written as a function of the deviation with respect to a steady-state condition. Thus, the following variables are defined:

$$\begin{cases} \delta x = x - x_{ref,mod} & \Rightarrow x = \delta x + x_{ref,mod} \\ \delta y = y - y_{ref} & \Rightarrow y = \delta y + y_{ref} \\ \delta u = u - u_{ref} & \Rightarrow u = \delta u + u_{ref} \end{cases} \tag{15}$$

Vectors y_{ref} and u_{ref} are obtained from Equation 10. The vector $x_{ref,mod}$ represents the 10 states (including x_D) on the steady-state regime. It is obtained from x_{ref} (also from Equation 10) by adding a reference value of x_D on the vector's fifth position. The value $x_{D,ref} = -h_L$ (where h_L is the runways altitude) is chosen to serve as reference. It must be noted that since x_D is not controllable, the closed-loop matrix $(A - BKC)$ is singular, prohibiting its use for the direct determination of $x_{ref,mod}$.

Using Equations 15 on Equations 13 and using the steady-state relations, the following set of equations is obtained:

$$\dot{\delta x} = A(\delta x) + B(\delta u) \quad \delta y = C(\delta x) \quad \delta u = -K(\delta y) \tag{16}$$

The discretization of Equations 16 results Equations 17. The subscript "d" indicates discrete-time state-space matrices and $\hat{}$ symbol indicates the future (and modifiable) elements that are predicted based on the information of the k instant.

$$\begin{cases} \hat{\delta x}(k+1|k) & = A_d(\delta x(k)) + B_d(\hat{\delta u}(k|k)) \\ \hat{\delta y}(k) & = C(\delta x(k)) \\ \hat{\delta u}(k|k) & = -K(\delta y(k)) \end{cases} \tag{17}$$

C. Supervising MPC

As shown in reference [1], Equations 17 can be used to design a supervising MPC: a configuration where the MPC acts as supervisor for a previously defined loop, constantly monitoring it's response in order to predict it's future states and outputs. This configuration can be seen on Figure 1.

The supervising MPC controller must minimize the following Cost Function:

$$\begin{aligned} J_{MPC} = \sum_{i=1}^{N} & \hat{\delta x}(k+i|k)^T Q_{MPC} \hat{\delta x}(k+i|k) + \\ & \hat{v}(k+i-1|k)^T R_{MPC} \hat{v}(k+i-1|k) \end{aligned} \tag{18}$$

Subject to the following constraint:

$$\delta x_{min} \leq \hat{\delta x}(k+i|k) \leq \delta x_{max} \tag{19}$$

where $\hat{\delta x}(k+i|k)$ and $\hat{v}(k+i|k)$ describe the predicted value of δx and the predicted MPC control vector v at the $k+i$ instant given the values of the states and outputs at the k instant. The weighting matrix for states is $Q_{MPC} \in \Re^{10\times10}$ and for MPC control is $R_{MPC} \in \Re^{2\times2}$. The vectors δx_{min} and δx_{max} contain lower and upper bounds of states. Moreover, the states ignored by MPC have their range set to $[-\infty, \infty]$.

Also, when the UAV's altitude is close to the runway's, the following terminal condition is inserted on the system:

$$\delta x_D(k+N|k) = 0 \tag{20}$$

By design choice, the Q_{MPC} matrix is set to zero ($0_{10\times10}$). With this configuration, the MPC will not penalize state deviations from the reference/equilibrium point. The applied $v(k)$ control will be nonzero only when constraint violations are predicted or when the terminal condition is active. When that is not the case, the predictive controller will keep monitoring the aircraft's restrictions and set $v(k) = 0$.

D. MPC's Automatic Landing Logic

In this design, the constraints and terminal condition were imposed by the MPC as follows:

1) The altitude shall not fall below the landing altitude h_L ($\delta x_{D,min} = -\infty$; $\delta x_{D,max} = 0$; $x_{D,ref} = -h_L$).
2) The aircraft shall engage landing mode when the difference between its current altitude and landing

altitude is less than a threshold ($h - h_L < threshold$). Then, the MPC shall activate the terminal condition (Equation 20) to ensure that the UAV reaches the landing altitude: $h(k+N|k) = h_L$. On this mode, the prediction horizon (N) will start to decrease by one unit in each iteration until it reaches a minimum value N_f. This strategy of receding horizon is applied to ensure that the terminal condition will be reached in defined number of iterations.

The first rule imposes a physical lower bound for the aircraft's altitude. In practical terms, it prevents the aircraft from making a hard crash with the landing course.

The second rule ensures that the aircraft touches the ground. The receding horizon will ensure that the aircraft does not take too long to land.

Because the first and second rules have concurrent objectives ($h > h_L$ and $h = h_L$, respectively), when landing mode is engaged, the first rule shall be disabled when the receding prediction horizon reaches half its original value ($N/2$).

IV. Results

A. Supervised Loop Gain

Based on development of previous sections, the UAV's control scheme can be implemented for its automatic landing. Results were obtained performing simulations with software MATLAB/SIMULINK. Linear models of the aircraft were obtained around trim flight conditions in wing-level flight (reference [2]): Velocity $V_{T_e} = 33m/s$, Altitude $h_e = 650m$, and flight-path angle $\gamma_e = 0^o$, (subscript "e" means equilibrium). A, B and C matrices from Equation 13 are displayed in Equation 22.

The chosen landing altitude for the simulation is $h_L = 600m$, which corresponds to the altitude of a runway situated on the average altitude of São José dos Campos city.

Following considerations of Section II and using the diagonal matrices $R = diag[10^5 \ 1.5 \times 10^5]$ and $W_e = diag[16 \ 10^4]$ as penalization matrices, the calculated optimum control K that minimizes Equation 11 is:

$$K = \begin{bmatrix} 0.0000 & -0.0000 & -0.3347 & -0.0293 & -0.0000 & -0.0000 & 0.0000 \\ -0.0279 & -0.5212 & -0.0000 & -0.0000 & 1.1953 & 0.2862 & 0.1914 \end{bmatrix} \quad (21)$$

B. Automatic Landing of the Aircraft

Once the supervised loop's autopilot is implemented (Equation 8), the supervising MPC can be designed.

The chosen sampling period is $T = 0.1s$: a time interval that should be large enough to enable the processing of the MPC's calculations on a real-time application while still having enough speed of response to properly control the aircraft. This sample time T is also applied in discretization by using structure with zero-order holder.

For the prediction horizon, it was chosen an initial value of $N = 30$. When the aircraft engages the landing mode ($h - h_L < threshold$), this horizon recedes to the final value of $N_f = 5$. The first rule of the supervising MPC ($h > h_L$) shall be disabled when the horizon recedes to $N = 15$, leaving only the second rule active.

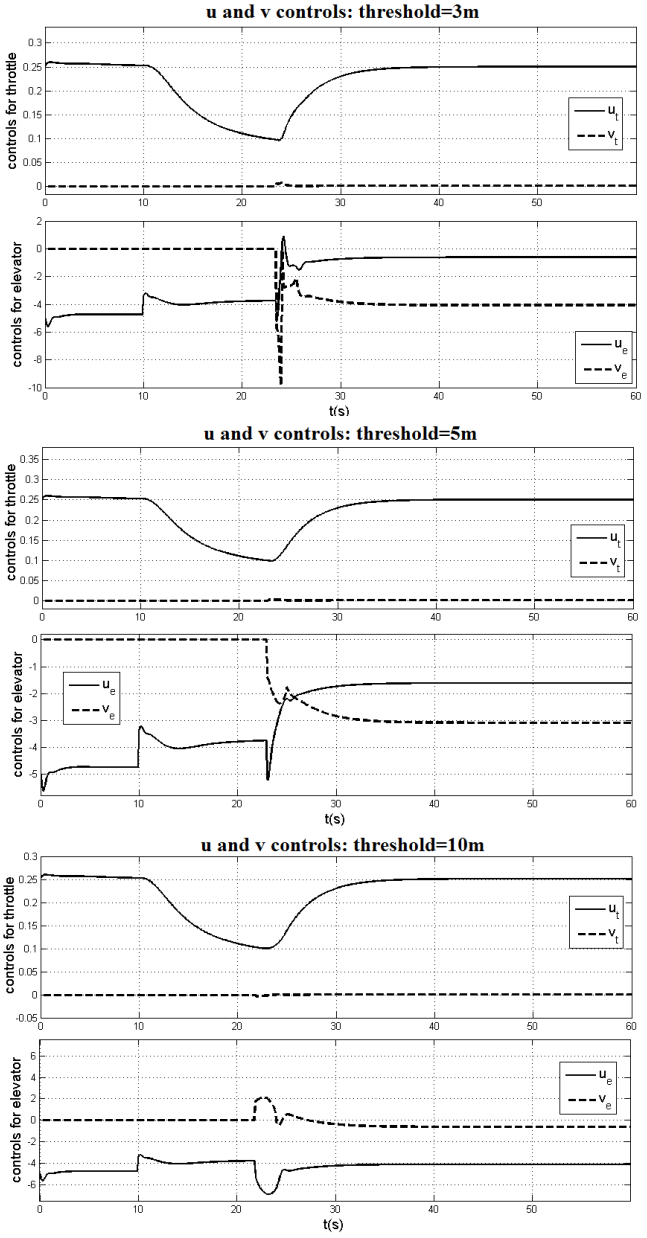

Fig. 6. u and v controls for different thresholds and -7^o glide slope.

Simulations of landing maneuver with the proposed design were performed with initial reference for flight-path angle $\gamma_c = 0^o$. At the time $t = 10s$, γ reference is switched to $\gamma_c = -7^o$ (gliding slope). The reference for velocity remains $V_c = 33m/s$ during the complete simulation. The initial altitude is $h = 650m$ and the desired landing altitude is $h_L = 600m$.

When the difference between the aircraft's altitude and the landing altitude is less than a certain threshold, the aircraft engages the landing mode. At this moment, the MPC's second rule becomes active and the γ reference is again switched to $\gamma_c = 0^o$.

The plots in Figure 6 present the contribution of the MPC ($v = [v_t \quad v_e]^T$) and contribution of the supervised loop ($u =$

$$A = \begin{pmatrix} -0.127 & 0.160 & 0.151 & -9.807 & -0.000 & 5.465 & -1.289 & 0 & 0 & 0 \\ -0.605 & -2.385 & 33.000 & 0.045 & -0.001 & 0 & -21.860 & 0 & 0 & 0 \\ -0.003 & -0.648 & -7.176 & 0 & 0.000 & 0 & -38.739 & 0 & 0 & 0 \\ 0 & 0 & 1.000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.005 & 1.000 & 0 & -33.000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -10 & 0 & 0 & 0 \\ -1.000 & 0.005 & 0 & 0 & 0 & 0 & 0 & -30 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -70 & 1 \\ 1e-4 & 0.030 & 0 & -1.000 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 4 & 0 \\ 0 & 10 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \quad (22)$$

$$C = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -1.0000 & 0.0046 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \\ 0.0001 & 0.0303 & 0 & -1.0000 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$[u_t \quad u_e]^T$) in the control signal applied to throttle ($v_t + u_t$) and elevator ($v_e + u_e$).

Figure 6 shows that in cases with threshold equal to $3m$ and $5m$ the MPC pushes the elevator up (convention: movement of elevator down for positive deflection). This happens because, since the threshold is low, the MPC must make the aircraft rise to avoid crashing. For threshold equal to $10m$ happened the opposite: the aircraft is too high for a successful landing. Therefore, the MPC pushes the elevator down to drive the aircraft down to the desired altitude. It should be noted that even for small thresholds (as for example $3m$) and steep gliding slope (-7^o) the supervising MPC prevents constraints violation applying enough control to actuate on the supervised loop.

Figure 7 shows the results of landing obtained for the three values of threshold, demonstrating efficiency of design to comply defined requirements.

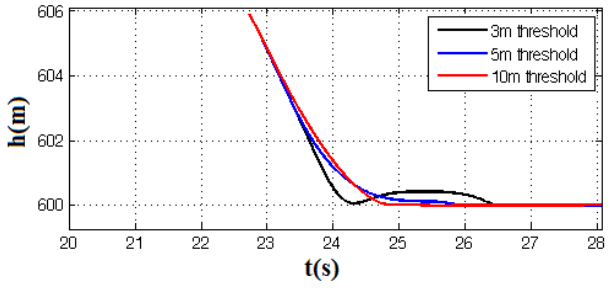

Fig. 7. Automatic landing for different thresholds and -7^o glide slope.

V. CONCLUSIONS

This project illustrates a powerful application of the MPC: to supervise a pre-existent control loop, making sure it achieves given design requirements that it otherwise would not be able to comply. An overview with the theory involving this kind of MPC implementation was given, covering all the steps necessary to provide to a simple velocity and γ autopilot the capability to monitor the aircraft's altitude for enforcement of two kinds of restrictions, culminating on a safe landing despite the harsh scenarios.

It can also be noted that the MPC was responsible for a virtual expansion of the supervised loop's states without having to change its structure. The designed controller proves its efficiency to comply tracking requirements and constraints. Thus, this paper presents an efficient approach for combining the reliability of a simple internal autopilot with the safety enforcement proprieties of a more complex MPC controller that only needs to actuate to correct or to help the internal loop in critical circumstances.

For future research, it would be interesting to include constraints monitoring of other relevant variables such as the Aircraft's Load Factor, Angle of Attack or the physical limits and rates of its actuators.

REFERENCES

[1] J. M. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, 2002.

[2] J. S. Santos and L. C.S. Goes, *System identification based on flight test data and autopilot design for longitudinal motion of the Vector-P UAV*, 22nd International Congress of Mechanical Engineering (COBEM 2013), November 3-7, 2013, Ribeirão Preto, SP, Brazil, 2013.

[3] B. L. Stevens and F. L. Lewis, *Aircraft flight control design using output feedback*, Journal of Guidance, 1992.

[4] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation*, John Wiley and Sons, 1992.

[5] F. A. de Almeida, *Aerodynamic model identification of a fighter trainer using partial orthogonal least squares*, AIAA Aerospace Sciences Meeting, Nashville, Proceedings, 2012.

[6] F. A. de Almeida, *Reference Management for Fault-Tolerant Model Predictive Control*, Journal of Guidance, Control, and Dynamics, Vol. 34, No.1, January-February 2011. doi: 10.2514/1.50938

[7] R. J.M. Afonso and R. K.H. Galvão, *Predictive Control of a Helicopter Model with Tolerance to Actuator Faults*, Conference on Control and Fault Tolerant Systems, Nice, France, October 6-8, 2010.