Autonomous landing of an UAV on a moving platform using Model Predictive Control

José Alfredo Macés-Hernández, François Defay, Corentin Chauffaut

Autonomous landing of an UAV on a moving platform using Model Predictive Control

José A. Macés-Hernández, François Defaÿ and Corentin Chauffaut

***Abstract*— This paper proposes a linear Model Predictive Control (MPC) for high-accuracy tracking of a mobile robotic platform, in an indoor environment. The mission is divided into three main phases: target detection, target tracking and autonomous lading. These phases were coded into a so-called guidance function, which allows the system to safely switch between different reference signals according to the state of the mission. In order to ease the control loop implementation in the experimental framework available at the facilities of ISAE Supaero, some existing optimisation tools were employed. In addition, some simulations and real tests were carried out to demonstrate the performance of the controller against other control techniques developed for the same experimental system.**

I. Introduction

Quadrotors have been widely used in the automatic control research field, these systems are ideal to test control algorithms given its non-linear dynamics, size and relative low price; but not less important, they have great number of applications in different fields; such as remote inspection, monitoring, aerial mapping and target recognition.

In [1], a control loop based on the Backstepping approach was formulated and implemented for the same UAV and experimental framework. Regarding the application of the reference tracking problem using visual feedback, [2], [3], and [4], dealt with the problem using PIDs, [5] developed a Backstepping control and [6] proposed a time delay robust decentralised controller.

Some researchers have also suggested a solution based on MPC; for instance, [7] proposes an integral predictive control for the path tracking problem, [8] and [9] address the solution using MPC for position stabilisation. [10] presents a linear model predictive control and provides some experimental results.

Concerning the target recognition, a tracking algorithm for ground robots was developed and successfully tested by [11]. This algorithm was developed for ground robots, but it can easily be applied to a moving platform like the one used in this investigation. On the other hand, in [12], [13] and [2], an on-board target detection algorithm is proposed, obtaining successful results in all cases.

II. System modelling

An inertial frame $\mathcal{F}_i = \{\mathcal{O}_i, X_i, Y_i, Z_i\}$ and a body frame $\mathcal{F}_b = \{\mathcal{O}_b, X_b, Y_b, Z_b\}$ are defined to fully represent the attitude and position dynamics of the quadrotor. The origin of the inertial frame can be placed at any point on the Earth's surface, while the body frame origin matches with the quadrotor's center of mass (see Figure 1). X_i- and Y_i-axes point towards Earth's geographical north and east, respectively; while the Z_i-axis points in the opposite direction of the gravity vector. In addition, the body frame X_b-axis crosses between the motors M_1 and M_2, and Y_b-axis is placed in the middle of motors M_2 and M_3, Z_b-axis completes an orthonormal system.

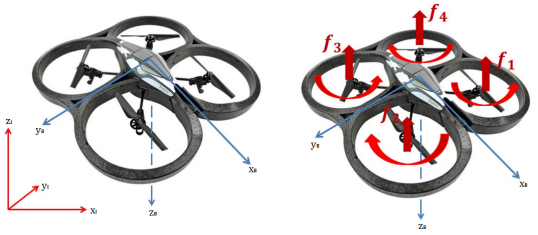

Fig. 1. Inertial and body frames

Given these reference frames, vectors

$$ {}^{i}\xi = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \text{and} \quad {}^{i}\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{1} $$

will be used to represent the UAV's center of gravity position and velocity with respect to the origin of the inertial frame; respectively. Vector

$$ {}^{b}\mathcal{V} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = {}^{b}\mathcal{R}_i \, {}^{i}\dot{\xi} = $$

$$ = \begin{bmatrix} c(\psi)c(\theta) & s(\psi)c(\theta) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi) & c(\theta)s(\phi) \\ c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}, \tag{2} $$

represents the linear velocity directly measured on the body frame.

Regarding the quadrotor's attitude, vectors

$$ {}^{i}\eta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad \text{and} \quad {}^{i}\dot{\eta} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \tag{3} $$

provide information about the angles and angular velocities of the body axis with respect to the inertial frame, and

$$ {}^{b}\Omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \mathcal{W} \, {}^{i}\dot{\eta} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & cos(\phi) & cos(\theta)\sin(\phi) \\ 0 & -\sin(\phi) & cos(\phi)cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}, \tag{4} $$

are the angular rates measured on the body frame.

As seen in Figure 1, the UAV is composed by a cross structure with a motor in the extremities, usually the structure

José A. Macés-Hernández, François Defaÿ and Corentin Chauffaut are with the Institut Supérieur de l'Aéronautique et de l'Espace (ISAE), 31055 Toulouse, France (e-mails: jose-alfredo.maces-hernandez@student.isae-supaero.fr, francois.defay@isae.fr and corentin.chauffaut@isae.fr).

is built of light material. Motors M_1 and M_3 rotate in clockwise direction, while M_2 and M_4 rotate in anticlockwise; this configuration induces opposite torques in the XY_b plane. The rotation of the motors produces a punctual effect in the UAV dynamics, which can either be a thrust force

$$f_i = k_f \, {\omega_i}^2, \tag{5}$$

or a torque

$$\tau_i = k_\tau \, {\omega_i}^2. \tag{6}$$

The combination of these small effects in $\mathcal{F}_b$ is translated into a total thrust

$$u = f_1 + f_2 + f_3 + f_4, \tag{7}$$

a torque in roll

$$\tau_\phi = l(f_1 - f_2 - f_3 + f_4), \tag{8}$$

a torque in pitch

$$\tau_\theta = l(f_1 + f_2 - f_3 - f_4), \tag{9}$$

and a torque in yaw

$$\tau_\psi = \tau_1 - \tau_2 + \tau_3 - \tau_4. \tag{10}$$

Therefore, a positive roll movement is produced if $\omega_2 + \omega_3 > \omega_1 + \omega_4$; in the same way, when $\omega_1 + \omega_2 > \omega_3 + \omega_4$, the UAV tilts positively in pitch; and a positive yaw results from $\omega_1 + \omega_3 > \omega_2 + \omega_4$.

Those equations can be rewritten in matrix form as

$$\begin{bmatrix} u \\ \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ k_a & -k_a & -k_a & k_a \\ k_a & k_a & -k_a & -k_a \\ k_c & -k_c & k_c & -k_c \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}, \tag{11}$$

where $k_c = \frac{k_\tau}{k_f}$ and $k_a = l$, k_τ and k_f are called torque and force constants, respectively; and l is the the distance from M_i to $\mathcal{O}_b$.

As proposed by [14], the dynamics of a quadrotor is obtained by using the rigid body equations, which are

$$\begin{array}{rcl} m \, {}^b\dot{\mathcal{V}} + {}^b\Omega \times (m \, {}^b\mathcal{V}) & = & {}^bF + {}^bG \, , \\ \mathbb{I} \, {}^b\dot{\Omega} + {}^b\Omega \times (\mathbb{I} \, {}^b\Omega) & = & {}^b\tau \, . \end{array} \tag{12}$$

For measurement and experimental purposes it easier to refer the translational dynamics to $\mathcal{F}_i$, while the attitude dynamics is referred to the body frame. Thus, equation (12) can be rewritten as

$$\begin{array}{rcl} m \, {}^i\mathcal{R}_b \, {}^b\dot{\mathcal{V}} & = & {}^i\mathcal{R}_b \, ({}^bF + {}^bG) \, , \\ \mathbb{I} \, {}^b\dot{\Omega} + {}^b\Omega \times (\mathbb{I} \, {}^b\Omega) & = & {}^b\tau \, , \end{array} \tag{13}$$

where

$$\mathbb{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \tag{14}$$

is the moment of inertia matrix, m the quadrotor's mass and ${}^iG$ the gravity vector. The Coriolis effect no longer appears in this equation since it does not affect the system's behaviour when the inertial frame is used as reference. After developing some algebra, the resulting equations are

$${}^b\dot{\Omega} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{I_{yy} - I_{zz}}{I_{xx}} q \, r + \frac{\tau_\phi}{I_{xx}} \\ \frac{I_{zz} - I_{xx}}{I_{yy}} p \, r + \frac{\tau_\theta}{I_{yy}} \\ \frac{I_{xx} - I_{yy}}{I_{I_{zz}}} p \, q + \frac{\tau_\psi}{I_{zz}} \end{bmatrix}, \tag{15}$$

for the position dynamics, and

$${}^i\ddot{\xi} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} [\sin(\phi)\sin(\psi) + cos(\phi)\sin(\theta)cos(\psi)] \frac{u}{m} \\ [-\sin(\phi)cos(\psi) + cos(\phi)sin(\theta)sin(\psi)] \frac{u}{m} \\ -g + cos(\phi)cos(\theta)\frac{u}{m} \end{bmatrix}, \tag{16}$$

for the attitude dynamics.

III. Control system design

A. System discretisation

For the design of the linear Model Predictive Controller, the system should be represented in a discrete state-space form. Therefore, an state augmentation using equation (16), followed by a linearisation assuming an equilibrium point in $\begin{bmatrix} \phi(0) & \theta(0) & \psi(0) \end{bmatrix}^T = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$, is performed. In addition, the sampling time for the discrete system was set to be $T_s = 0.02 \; s$ (this is the overall sampling time of the experimental system, which will be explained in the respective section). This evaluations result in

$$\begin{bmatrix} \dot{x}_d \\ \dot{y}_d \\ \dot{z}_d \\ \ddot{x}_d \\ \ddot{y}_d \\ \ddot{z}_d \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0.02 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.02 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.02 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0.001962 & 0 \\ 0 & -0.001962 & 0 & 0 \\ 0.0003717 & 0 & 0 & 0 \\ 0 & 0 & 0.1962 & 0 \\ 0 & -0.1962 & 0 & 0 \\ 0.03717 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ \phi \\ \theta \\ \psi \end{bmatrix}. \tag{17}$$

The physical parameters of the quadrotor $\left(m = 0.5380 \; kg \text{ and } g = 9.81 \; m \cdot s^{-2}\right)$ were substituted for simplification purposes. The inputs of the control system will be the thrust and the euler angles, they will be sent to the inner attitude control loop, which is assumed to have a quick response time ($< 0.6 \; s$). The inner loop has been tuned in the PX4 board (multirotor attitude of Pixhawk is used as default).

B. Linear predictive model

The objective of this control formulation is to make the system's output $y(t)$, track a reference signal $r(t)$ with the smallest possible error. For this purpose, it is necessary to reformulate the state space representation, starting from

$$\begin{cases} x(k+1) = A \, x(k) + B \, u(k) \, , \\ \quad y(k) = C \, x(k) \, , \end{cases} \tag{18}$$

and assuming that $\Delta u(k) = u(k) - u(k-1)$, the control inputs vector $u(k)$ in equation (18) can be rewritten as

$$\begin{cases} x(k+1) = A\,x(k) + B\,u(k-1) + B\Delta u(t)\,, \\ u(k) = u(k-1) + \Delta u(k)\,. \end{cases} \tag{19}$$

If Δu is considered as the vector of control inputs, and $u(k-1)$ is added to the states vector [15]. The system can be reordered as

$$\begin{cases} \begin{bmatrix} x(k+1) \\ u(t) \end{bmatrix} &= \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(k)\,, \\ y(k) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ u(k-1) \end{bmatrix}. \end{cases} \tag{20}$$

For simplification purposes the new state vector will be $\mathbf{x}(k) = \begin{bmatrix} x(k) & u(k-1) \end{bmatrix}^T$. Consequently, the system can be represented in a more compact way as

$$\begin{cases} \mathbf{x}(k+1) &= \mathcal{A}\,\mathbf{x}(k) + \mathcal{B}\,\Delta u(k)\,, \\ y(k) &= \mathcal{C}\,\mathbf{x}(k)\,. \end{cases} \tag{21}$$

The current time k is always available from the measurements, $\mathbf{x}(k)$ provides the current value of the plant states. In the same manner, the future control trajectory is denoted by

$$\Delta u = \begin{bmatrix} \Delta u(k)^T & \Delta u(k+1)^T & \dots & \Delta u(k+n_c-1)^T \end{bmatrix}^T, \tag{22}$$

where n_c is called control horizon and it represents the number of parameters used to capture the future control trajectory. In addition, given the model of the system (21), $\mathbf{x}(k)$ can be computed for a number n_p of samples; n_p is known as the prediction horizon, and it is used to compute the effects of the current control input in the future. It is necessary to remark that the control horizon should always be lower or equal than the prediction horizon [16].

From the state-space representation of equation (21), the future value of the state variables

$$\begin{aligned} \mathbf{x}(k) \quad = \quad & \begin{bmatrix} \mathbf{x}(k+1|k)^T & \mathbf{x}(k+2|k)^T & \dots \\ & \mathbf{x}(k+m|k)^T & \dots & \mathbf{x}(k+n_p|k)^T \end{bmatrix}^T, \end{aligned}$$

can be computed using the future values of the control inputs as

$$\begin{aligned} \mathbf{x}(k+1|k) &= \mathcal{A}\,\mathbf{x}(k) + \mathcal{B}\Delta u(k), \\ \mathbf{x}(k+2|k) &= \mathcal{A}\,\mathbf{x}(k+1|k) + \mathcal{B}\Delta u(k+1), \\ &= \mathcal{A}^2\mathbf{x}(k) + \mathcal{A}\,\mathcal{B}\Delta u(k) + \mathcal{B}\Delta u(k+1), \\ &\vdots \\ \mathbf{x}(k+n_p|k) &= \mathcal{A}^{n_p}\,\mathbf{x}(k) + \mathcal{A}^{n_p-1}\mathcal{B}\Delta u(k) \\ &\quad + \mathcal{A}^{n_p-2}\mathcal{B}\Delta u(k+1) + \dots \\ &\quad + \mathcal{A}^{n_p-n_c}\mathcal{B}\Delta u(k+n_c-1). \end{aligned} \tag{23}$$

In the same way, the future value of the outputs

$$y(k) = \begin{bmatrix} y(k+1|k)^T & y(k+2|k) & \dots & y(k+n_p|k)^T \end{bmatrix}^T, \tag{24}$$

can be estimated by using the predicted state previously defined, leading to

$$\begin{aligned} y(k+1|k) &= \mathcal{C}\mathcal{A}\,\mathbf{x}(k) + \mathcal{C}\mathcal{B}\Delta u(k), \\ y(k+2|k) &= \mathcal{C}\mathcal{A}^2\mathbf{x}(k) + \mathcal{C}\mathcal{A}\mathcal{B}\Delta u(k) + \mathcal{C}\mathcal{B}\Delta u(k+1), \\ y(k+3|k) &= \mathcal{C}\mathcal{A}^3\mathbf{x}(k) + \mathcal{C}\mathcal{A}^2\mathcal{B}\Delta u(k) \\ &\quad + \mathcal{C}\mathcal{A}\mathcal{B}\Delta u(k+1) + \mathcal{C}\mathcal{B}\Delta u(k+2), \\ &\vdots \\ y(k+n_p|k) &= \mathcal{C}\mathcal{A}^{n_p}\,\mathbf{x}(k) + \mathcal{C}\mathcal{A}^{n_p-1}\mathcal{B}\Delta u(k) \\ &\quad + \mathcal{C}\mathcal{A}^{n_p-2}\mathcal{B}\Delta u(k+1) + \dots \\ &\quad + \mathcal{C}\mathcal{A}^{n_p-n_c}\mathcal{B}\Delta u(k+n_c-1). \end{aligned} \tag{25}$$

From vectors (23) and (25), a so-called estimation model of the form (18) can be obtained as follows:

$$y(k) = F\mathbf{x}(k) + \Phi\Delta u\,, \tag{26}$$

where

$$F = \begin{bmatrix} \mathcal{C}\mathcal{A} & \mathcal{C}\mathcal{A}^2 & \dots & \mathcal{C}\mathcal{A}^{n_p} \end{bmatrix}^T,$$

and

$$\Phi = \begin{bmatrix} \mathcal{C}\mathcal{B} & 0 & 0 & \dots & 0 \\ \mathcal{C}\mathcal{A}\mathcal{B} & \mathcal{C}\mathcal{B} & 0 & \dots & 0 \\ \mathcal{C}\mathcal{A}^2\mathcal{B} & \mathcal{C}\mathcal{A}\mathcal{B} & \mathcal{C}\mathcal{B} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{C}\mathcal{A}^{n_p-1}\mathcal{B} & \mathcal{C}\mathcal{A}^{n_p-2}\mathcal{B} & \mathcal{C}\mathcal{A}^{n_p-3}\mathcal{B} & \dots & \mathcal{C}\mathcal{A}^{n_p-n_c}\mathcal{B} \end{bmatrix}.$$

C. Optimisation: Tracking Formulation

The objective of the position controller proposed in this work is to bring the system outputs $y(k)$ as close as possible to the reference $r(k)$. Therefore, a performance index function

$$J(k) = \sum_{k=0}^{n_p-1} \left(\|\mathcal{W}_y(y(k) - r(k))\|^2 + \|\mathcal{W}_{\Delta u}\Delta u(k)\|^2 \right), \tag{27}$$

is defined, where

$$\begin{aligned} \mathcal{W}_y &= diag(w_{y_0}, w_{y_1}, \dots, w_{y_n})\,, \\ \mathcal{W}_{\Delta u} &= diag(w_{\Delta u_0}, w_{\Delta u_1}, \dots, w_{\Delta u_n}), \end{aligned}$$

are called weighting matrices. Given this formulation, the new problem is to find the best control input Δu such that the tracking error is minimised. Therefore, equation (27) can be rewritten in a matrix form as

$$J = (y(k) - r(k))^T {\mathcal{W}_y}^2 (y(k) - r(k)) + \Delta u^T {\mathcal{W}_{\Delta u}}^2 \Delta u\,. \tag{28}$$

Then, the substitution of (26) in the previous equation leads to

$$\begin{aligned} J \quad = \quad & (F\mathbf{x}(k) - r(k))^T {\mathcal{W}_y}^2 (F\mathbf{x}(k) - r(k)) \\ & + 2\Delta u^T \Phi^T {\mathcal{W}_y}^2 (F\mathbf{x}(k) - r(k)) \\ & + \Delta u^T (\Phi^T {\mathcal{W}_y}^2 \Phi + {\mathcal{W}_{\Delta u}}^2)\Delta u\,, \end{aligned} \tag{29}$$

which minimal value can be obtained from its derivative with respect to the control input Δu as

$$\frac{\partial J}{\partial \Delta u} = 2\Phi^T {\mathcal{W}_y}^2 (F\mathbf{x}((k) - r(k)) + 2(\Phi^T {\mathcal{W}_y}^2 \Phi + {\mathcal{W}_{\Delta u}}^2)\Delta u = 0, \tag{30}$$

therefore, the resulting controller will have the form

$$\Delta u = -(\Phi^T {\mathcal{W}_y}^2 \Phi + {\mathcal{W}_{\Delta u}}^2)^{-1} \Phi^T {\mathcal{W}_y}^2 (F\mathbf{x}(k) - r(k)). \tag{31}$$

It can be seen that the final controller is directly linked to the reference signal and the system states, so it should be computed in real time at any new sample.

IV. Target detection and landing algorithm

For this investigation, not only a MPC controller was developed and proposed, but also a guidance law. The ground robot to be used as tracking target is presented in Figure 2.

Fig. 2. Mobile ground robot

The final guidance law has four states:

1) Take-off: The UAV turns-on the motors, consequently, it takes-off from either the ground level $z(0) = 0$ or the mobile robot $z(0) = h_{rob}$. The controller will stabilise the UAV at a given altitude $z_r(k) = z_{track}$, while $x_r(k)$ and $y_r(k)$ will remain constant. The system will immediately switch to the next state.

$$\left[x_r, y_r, z_r, v_{x_r}, v_{y_r}, v_{z_r}\right] = \left[x(0), y(0), z_{track}, 0, 0, v_{z_{track}}\right].$$

2) Patrol mode: The UAV starts moving within the flying arena until the camera system detects the robot in a near position. The selected tracking pattern is a spiral on XY_i plane with variable amplitude and frequency. The condition to switch to the next state is the detection of the robot.

$$\left[x_r, y_r, z_r, v_{x_r}, v_{y_r}, v_{z_r}\right] = \left[x_{pat}, y_{pat}, z_{track}, v_{x_{pat}}, v_{y_{pat}}, v_{z_{track}}\right].$$

3) Tracking mode: As soon as the target is detected, the quadrotor starts tracking it at the given $z_r(k) = z_{track}$, while the position (x_{rob} and y_{rob}) of the robot measured by the camera is sent to the control loop as reference. The robot velocity is estimated by means of a discrete derivative with a second order low-pass filter since the camera system is only able to provide position measurements. If the robot is lost at any time during this phase, the state machine will switch back to the previous state.

$$\left[x_r, y_r, z_r, v_{x_r}, v_{y_r}, v_{z_r}\right] = \left[x_{rob}, y_{rob}, z_{track}, v_{x_{rob}}, v_{y_{rob}}, v_{z_{track}}\right].$$

4) Landing mode: The UAV descends if the velocity of the robot is smaller that $x_{rob_{max}}$ and $y_{rob_{max}}$, after reaching the platform the motors are switched-off. If the robot is lost during the descending phase, the state machine will return to the second state.

$$\left[x_r, y_r, z_r, v_{x_r}, v_{y_r}, v_{z_r}\right] = \left[x_{rob}, y_{rob}, -z(k) + 1, v_{x_r}, v_{y_r}, \dot{z}_{ref}\right].$$

All the phases previously presented are shown in a more illustrative way in Figure IV, where also the change conditions are defined.

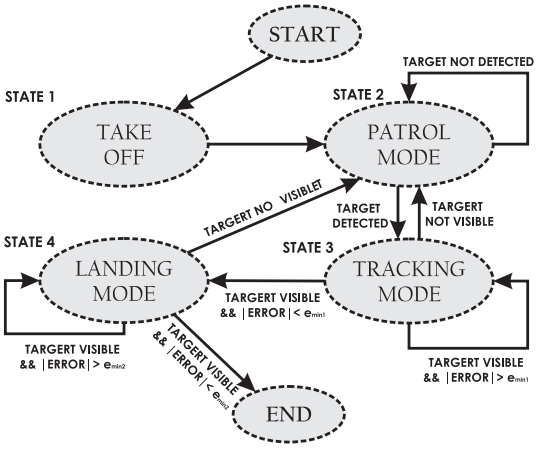

Fig. 3. State machine for the guidance law

V. Experimental framework

At ISAE Supaero, a complete experimental framework has been developed along the years by the researchers and students. This system allows one to test high performance control laws within a Simulink interface. The experimental environment is presented in Figure 4, the main source of measurements is an Optitrack motion capture system. For this investigation, an ARdrone2.0 frame is used, on which the embedded system has been replaced by a PX4 board for the attitude control loop, and a Gumstix board for the guidance and high-level control.

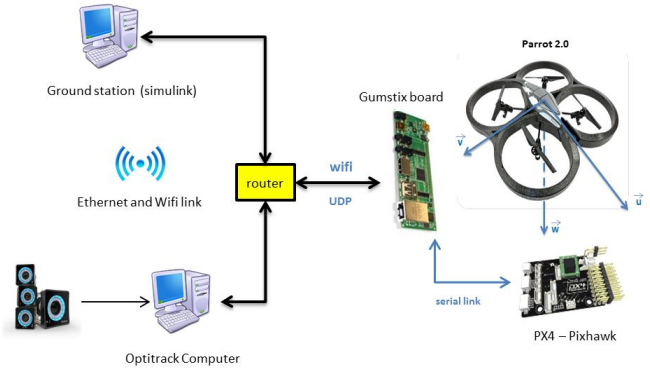

Fig. 4. Experimental environment

The Simulink Project called SimoDrones allows one to choose the quadrotor configuration for the experimentation, either a real time experiment (with Simulink running in real time withing a Windows kernel) or simulation can be selected. SimoDrones includes many non-linear models (brushless motor dynamics, communication delays, measurement noise, quadrotor dynamics and aerodynamic drag) in order to validate advanced control laws first in simulations, and then in real experimentation. The results that are presented later on where directly extracted from SimoDrones after performing reals tests.

The final cascade architecture control and its corresponding Simulink model are presented in Figures 5 and 6, respectively.

For the tuning of the controller the following parameters an constraints were selected:

- n_p = 10,

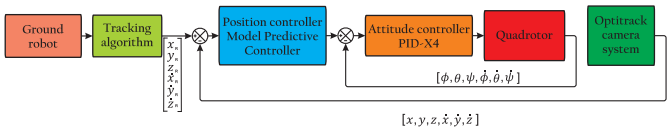

Fig. 5. Final control architecture

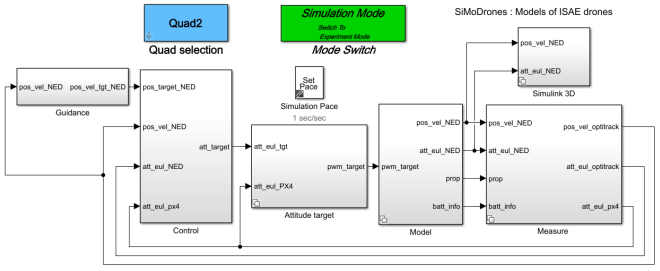

Fig. 6. SimoDrones project

- n_c = 10,
- $\mathcal{W}_y = diag(8,\ 8,\ 6,\ 2,\ 2,\ 2)$,
- $\mathcal{W}_{\Delta u} = diag(0.8,\ 0.5,\ 0.1,\ 0.1)$,
- $u_{max} = \begin{bmatrix} 0.5\,mg & 45° & 45° & 180° \end{bmatrix}$,
- $u_{min} = \begin{bmatrix} -0.5\,mg & -45° & -45° & -180° \end{bmatrix}$.

VI. Experimental Results

A. Comparison: MPC vs LQR, PID and Backstepping

For the performance analysis, the controller proposed in this paper was compared to some other control techniques available at ISAE Supaero drones library, they are: PID, LQR and Backstepping[1]. The results are presented in Figure 7, during this experimental phase, the reference signal is a set of way-points where the guidance function switches to the next way-point in function of the steady state error value and the the time response of each controller.

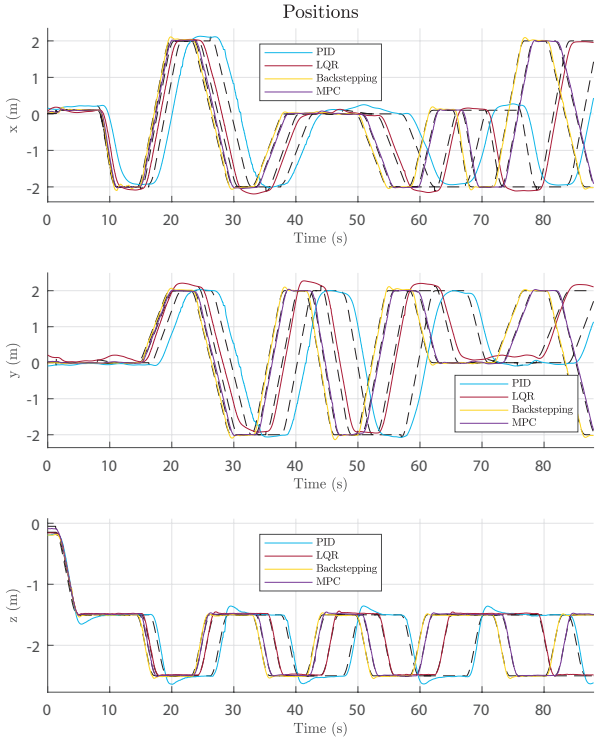

Fig. 7. Controllers comparison

Concerning the results, the Backstepping and the MPC have the better response, it can be observed that PID and LQR need a longer time to minimise the steady state error; consequently, to finish the full set of way-points, a longer time is required. The Backstepping technique offers the shortest delay between the reference signal and the output, and the MPC has a slightly slower response, which for the experimental purposes is almost negligible ($\approx 0.1\ s$). In terms of power consumption and aggressiveness of the control input, the MPC has shown to be a better alternative. When the tracking error is analysed, both controllers have a similar response; however, the overshot provided by the MPC is a bit smaller, and therefore a better alternative for the target tracking problem.

B. Landing on the moving target

After proving that the MPC minimised faster the steady state error with the minimal amount oscillations, and its overshot was the smallest among the different techniques, some tests using the new guidance algorithm proposed in section IV were carried out. Figure 8 is composed of four plots; the first three are the position measurements, and the last one shows the transitions of the state machine. Regarding the position in X_i- and Y_i- axes, there are three signals in each plot, in purple the real position of the robot, in blue the patrol reference signal, and finally the real position of the quadrotor in red.

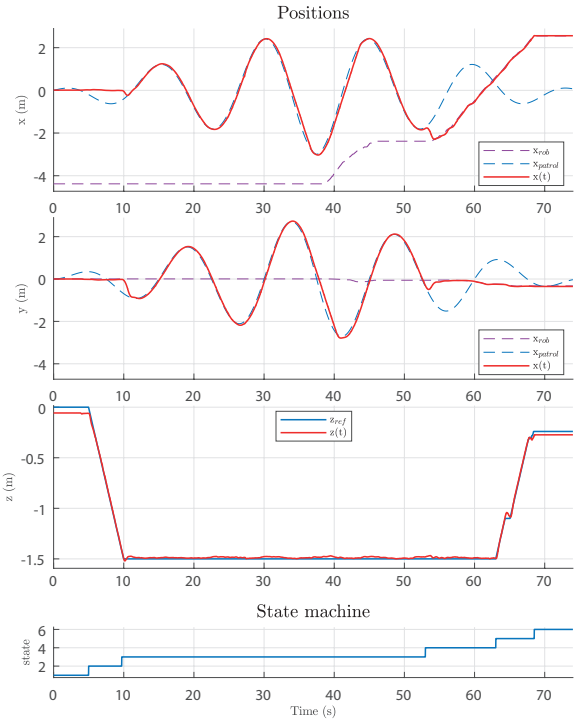

Fig. 8. Landing sequence: states and references

The mission profile has the following sequence: at 5 s the quadrotor takes-off, when the desired z_{track} (at around 10 s) is reached, the new reference will be the patrol spiral. As soon as the camera detects the robot within the vision range, the reference is switched to the measurement of the robot position. Subsequently, the UAV starts descending at around 63 s. At some point during this phase, the velocity of the robot was too high, therefore the quadrotor waited until it decreased to keep descending. Finally, the motors are switched-off as

soon as the UAV touches the platform. The whole mission takes around 65 s, however this time can be improved given that some protection flags are used. The video of the mission can be retrieved from [17].

VII. Conclusions

In this paper, it has been shown the validity of using MPC for accurate target tracking. The MPC was successfully implemented in the experimental framework and many simulations and experimental tests were carried out in order to validate this proposition. The predictive model model developed for this investigation has correctly described the position dynamics of the UAV. This model has also proven to be accurate for the estimation of the future control inputs and states in the linear predictive model.

By comparing the developed controller against the other available control techniques, the one propose in this paper has demonstrated to be the best solution for a mission where it is important to have an accurate tracking. However, the controller had a short delay ($\approx 0.1\,s$) due to the computations of the control input at every sampling time, which for experimental purposes did not compromise the response of the system since this value is very short.

The state machine based guidance law proposed in this paper has also proven to have an excellent behaviour in the experimental framework. All the transitions between states were optimised for the safety and integrity of the UAV. After performing many tests, the results show that the guidance function can perfectly model any UAV mission, and some more complex functions can be included in the future.

Many test were carried out, which demonstrated the validity of both solutions working together in the same experimental system. The resulting functions were added to the ISAE Supaero drones library.

Further studies would include the substitution of the internal attitude control loop by a full position and attitude MPC controller, using the same experimental framework.

References

[1] M. Lecointe, C. P. C. Chanel, and F. Defay, "Backstepping control law application to path tracking with an indoor quadrotor," in *Proceedings of European Aerospace Guidance Navigation and Control Conference (EuroGNC)*, (Toulouse, FR), pp. pp.1–19, April 2015.

[2] J. Ajmera, S. PR, R. K. M., G. Vasan, N. Balaji, and V. Sankaranarayanan, "Autonomous visual tracking and landing of a quadrotor on a moving platform," in *2015 Third International Conference on Image Information Processing (ICIIP)*, pp. 342–347, Dec 2015.

[3] K. Ling, D. Chow, A. Das, and S. L. Waslander, "Autonomous Maritime Landings for Low-Cost VTOL Aerial Vehicles," in *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pp. 32–39, May 2014.

[4] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic Take Off, Tracking and Landing of a Miniature UAV on a Moving Carrier Vehicle," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1, pp. 221–238, 2011.

[5] D. Lee, T. Ryan, and H. J. Kim, "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 971–976, May 2012.

[6] J. M. Daly, Y. Ma, and S. L. Waslander, "Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays," *Autonomous Robots*, vol. 38, no. 2, pp. 179–191, 2015.

[7] G. V. Raffo, M. G. Ortega, and F. R. Rubio, "An Integral Predictive/Nonlinear H_∞ Control Structure for a Quadrotor Helicopter," *Automatica*, vol. 46, pp. 29–39, Jan. 2010.

[8] P. Bouffard, "On-board Model Predictive Control of a Quadrotor Helicopter: Design, Implementation, and Experiments," Master's thesis, EECS Department, University of California, Berkeley, 2012.

[9] M. W. Mueller and R. D'Andrea, "A model predictive controller for quadrocopter state interception," in *Control Conference (ECC), 2013 European*, pp. 1383–1389, July 2013.

[10] M. Bangura and R. Mahony, "Real-time Model Predictive Control for Quadrotors," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11773 – 11780, 2014. 19th IFAC World Congress.

[11] W. Ren, J.-S. Sun, R. W. Beard, and T. W. McLain, "Nonlinear tracking control for non-holonomic mobile robots with input constraints: an experimental study," in *Proceedings of the 2005, American Control Conference, 2005.*, pp. 4923–4928 vol. 7, June 2005.

[12] C. Teulire, L. Eck, and E. Marchand, "Chasing a moving target from a flying UAV," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4929–4934, Sept 2011.

[13] K. Ling, "Precision Landing of a Quadrotor UAV on a Moving Target Using Low-Cost Sensors," Master's thesis, University of Waterloo, Ontario, Canada, 2014.

[14] S. Bouabdallah, *Design and Control of Quadrotors with Applicatioin to Autonomous Flying*. PhD thesis, École Polytechnique Fédárale de Lausanne, Laboratoire de systémes autonomes, 2007.

[15] J. Maciejowski, *Predictive Control with Constraints*. UK: Prentice-Hall International, 2002.

[16] B. Kouvaritakis and M. Cannon, *Model Predicitive Control: Classical, Robust and Stochastic*. London, UK: Springer International, 2015.

[17] J. A. Macés-Hernández, F. Defaÿ, and C. Chauffaut, "Autonomous landing of an UAV on a moving platform." `https://video.isae.fr/videos/?video=MEDIA170719172018432`, 2017.