Memorias del XVI Congreso Latinoamericano
de Control Automático, CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México

Model Predictive Path-Following Control of an AR.Drone Quadrotor [⋆]

Andres Hernandez [*,**], **Harold Murcia** [**], **Cosmin Copot** [*] and **Robin De Keyser** [*]

[*] *Department of Electrical energy, Systems and Automation, Ghent University, Belgium* (Andres.Hernandez,Cosmin.Copot}@UGent.be).
[**] *Department of Automation and Industrial Control, D+TEC research group, Universidad de Ibague, Colombia.*

Abstract:
This paper addresses the design and implementation of the Extended Prediction Self-Adaptive Control (EPSAC) approach to Model Predictive Control (MPC) for path-following. Special attention is paid to the closed-loop performance in order to achieve a fast response without overshoot, which are the necessary conditions to ensure the desired tracking performance in confined or indoor environments. The performance of the proposed MPC strategy is compared to the one achieved using PD controllers. Experimental results using the low-cost quadrotor AR.Drone 2.0 validate the reliability of the proposed strategy for 2D and 3D movements.

Keywords: Autonomous vehicles, Closed-loop control, path-following, flight control, Model Predictive Control

1. INTRODUCTION

In recent years, a big interest has emerged in the use of Unmanned Aerial Vehicles (UAVs) on applications such as aerial photogrammetry (Mokhtar et al. (2012)), agriculture (Berni et al. (2009)), habitat mapping (Dijkshoorn (2012)), and military tasks (Bednowitz et al. (2014)). One type of aerial vehicle which can accomplish this, is a quadrotor. The quadrotor is a micro UAV with four rotating blades which enable flight in a way similar to that of a helicopter. Movement is attained by varying the speeds of each blade thereby creating different thrust forces.

In order to accomplish the above mentioned missions without constant supervision of human operators, the UAV must autonomously follow predefined paths in 2D or 3D space. Usually, the problems of motion control for a single autonomous vehicle are roughly classified into three groups. Namely, point stabilization, where the goal is to stabilize a vehicle about a given target point with a desired orientation; trajectory tracking, where the vehicle is required to track a time parametrized reference; and path-following, where the quadrotor is required to follow a desired geometric path, implying a constraint in space, but not in time. Thus, the time it takes the quadrotor to reach the target position does not matter here.

In the literature, several authors present studies about path-following and reference tracking using UAVs. In (Xargay et al. (2012)) a cooperative control strategy for path-following of multiple autonomous vehicles is presented. The stability and convergence of the control strategy is addressed in Aguiar and Hespanha (2003) using Lyapunov-based design techniques. A more recent work (Alessandretti et al. (2013)) presents simulation results of a MPC strategy applied to 2-D and to 3-D moving vehicles.

In this study, we propose the use and real-life implementation of the Extended Prediction Self-Adaptive Control (EPSAC) approach to Model Predictive Control (MPC) for path-following control, as a continuation of the authors' previous work (Vlas et al. (2013)), where the first steps towards identification and position control of an AR.Drone 2.0 was performed. This quadrotor available to the mass market, was chosen thanks to its simple structure, sufficient sensory equipment and ease of maintenance, at a very low price. These features prove the quadrotor to be a good subject for both study purposes and practical applications. The quadrotor can fly both indoor and outdoor and is able to perform aggressive aerial maneuvers and to establish wireless communication to a ground station using Wi-Fi (Bristeau et al. (2011)).

The performance of the proposed MPC-EPSAC strategy for path-following is compared to the one achieved using PD controllers, for the tuning specifications of fast tracking without overshoot. In order to qualitatively compare the performance, the tracking error is used as evaluation criteria, using the well-known performance index ISE, IAE and ITAE.

The paper is organized as follows: Section 2 presents the model predictive control (MPC) architecture. In section 3 the AR.Drone 2.0 quadrotor is fully described. The control design of the path-following strategies is presented in section 4. Next in section 5 the experimental results are

[⋆] 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.
Andres Hernandez acknowledges the financial support provided by the Institute for the Promotion and Innovation by Science and Technology in Flanders (IWT SBO-110006).

presented, followed by a conclusion section where the main outcome of this work is summarized.

2. MODEL PREDICTIVE CONTROL

Model Predictive Control (MPC) is a general designation for controllers that make an explicit use of a model of the plant to obtain the control signal by minimizing an objective function over a time horizon in the future. In this contribution we will make use of the Extended Prediction Self-Adaptive Control (EPSAC) approach to MPC. This methodology proposed by (De Keyser (2003)) is briefly described as follows:

In the EPSAC algorithm, the model output $x(t)$ represents the effect of the control input $u(t)$ on the process output $y(t)$. It can be described by the following equation:

$$x(t) = f\left[x(t-1), x(t-2), \ldots, u(t-1), u(t-2), \ldots\right] \quad (1)$$

Notice that $x(t)$ represents here the model output, not the state vector. Also important is the fact that f can be either a linear or a nonlinear function. The generic model of the EPSAC algorithm is:

$$y(t) = x(t) + n(t) \quad (2)$$

where $y(t)$ is the measured output of the process, $x(t)$ is the model output and $n(t)$ represents model/process disturbance, all at discrete-time index t. The disturbance $n(t)$ can be modeled as colored noise through a filter with the transfer function

$$n(t) = \frac{C(q^{-1})}{D(q^{-1})} e(t) \quad (3)$$

with $e(t)$ uncorrelated (white) noise with zero-mean and C, D monic polynomials in the backward shift operator q^{-1}. The disturbance model allows to achieve robustness of the control loop against unmeasured disturbances and model errors. A 'default' choice to remove steady-state control offsets is $n(t) = \frac{1}{1-q^{-1}} e(t)$ (Maciejowski. (2002)).

A fundamental step in the MPC methodology is the prediction. Using the generic process model (2), the predicted values of the output are described by

$$y(t+k|t) = x(t+k|t) + n(t+k|t)$$

for $k = N_1, N_1+1, \ldots, N_2 | N_1, N_2 \in \Re$, where N_1 and N_2 are the minimum and the maximum prediction horizons. The prediction of the process output is based on the measurements available at the sampling instant t, $\{y(t), y(t-1), \ldots, u(t-1), u(t-2), \ldots\}$ and future (postulated) values of the input signal $\{u(t|t), u(t+1|t), \ldots\}$. The future response can then be expressed as

$$y(t+k|t) = y_{base}(t+k|t) + y_{opt}(t+k|t)\,, \quad (4)$$

where each of the contribution terms is understood as:

- $y_{base}(t+k|t)$ is the effect of the past inputs $u(t-1), u(t-2)\ldots$, a future base control sequence $u_{base}(t+k|t)$ that can be the last used input and the predicted disturbance $n(t+k|t)$.
- $y_{opt}(t+k|t)$ is the effect of the optimizing control actions $\delta u(t|t), \ldots, \delta u(t+N_u-1|t)$ with $\delta u(t+k|t) = u(t+k|t) - u_{base}(t+k|t)$, in a control horizon N_u.

The optimized output $y_{opt}(k)\,\forall\, k = [1, 2, \ldots, N_2]$ can be expressed as the discrete time convolution of the unit impulse response coefficients $h_1, \ldots, h_{N_2}$ and unit step response coefficients $g_1, \ldots, g_{N_2}$ of the system as.

$$y_{opt}(t+k|t) = h_k \delta u(t|t) + h_{k-1}\delta u(t+1|t) + \ldots$$
$$+ g_{k-N_u+1}\delta u(t+N_u-1|t) \quad (5)$$

Using (4) and (5), the key EPSAC-MPC formulation becomes

$$\mathbf{Y} = \overline{\mathbf{Y}} + \mathbf{GU} \quad (6)$$

where

$$\mathbf{Y} = [y(t+N_1|t) \ldots y(t+N_2|t)]^T$$
$$\overline{\mathbf{Y}} = [y_{base}(t+N_1|t) \ldots y_{base}(t+N_2|t)]^T$$
$$\mathbf{U} = [\delta u(t|t) \ldots \delta u(t+N_u-1|t)]^T$$

$$\mathbf{G} = \begin{bmatrix} h_{N_1} & h_{N_1-1} & \cdots & g_{N_1-N_u+1} \\ h_{N_1+1} & h_{N_1} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ h_{N_2} & h_{N_2-1} & \cdots & g_{N_2-N_u+1} \end{bmatrix} \quad (7)$$

Then, the control signal U is optimized by minimizing the cost function:

$$J = \sum_{k=N_1}^{N_2} [r(t+k|t) - y(t+k|t)]^2 \quad (8)$$

Note that the controller cost function (8) can be easily extended to many alternative cost functions (similar to the approach in optimal control theory) as described in (De Keyser (2003)). The horizons N_1, N_2 and N_u are design parameters and

$$r(t+k|t) = \alpha r(t+k-1|t) + (1-\alpha) w(t+k|t)$$

is the desired *reference trajectory*, where a 1st-order trajectory was chosen for $k = 1, \ldots, N_2$ with initialization $r(t|t) = y(t)$.The signal $w(t)$ represents the setpoint and alpha (α) is a design parameter to tune the MPC performance (Sanchez and Rodellar. (1996)).

The cost function (8) can be represented in its compact matrix notation as follows:

$$(\mathbf{R}-\mathbf{Y})^{\mathbf{T}}(\mathbf{R}-\mathbf{Y}) = [(\mathbf{R}-\overline{\mathbf{Y}}) - \mathbf{GU}]^{\mathbf{T}}[(\mathbf{R}-\overline{\mathbf{Y}}) - \mathbf{GU}]$$

where $\mathbf{R} = [r(t+N_1|t) \ldots r(t+N_2|t)]^T$.

The previous expression can be easily transformed into the standard quadratic cost index:

$$J(\mathbf{U}) = \mathbf{U^T H U} + \mathbf{2fU} + c. \quad (9)$$

with,

$$\mathbf{H} = \mathbf{G^T G} \quad \mathbf{f} = -\mathbf{G^T}(\mathbf{R}-\overline{\mathbf{Y}})$$
$$c = (\mathbf{R}-\overline{\mathbf{Y}})^{\mathbf{T}}(\mathbf{R}-\overline{\mathbf{Y}}) \quad (10)$$

where $[G^T G] \in \Re^{N_u \times N_u}$. The solution of minimizing (9) is:

$$\mathbf{U}^* = [\mathbf{G^T G}]^{-1}[\mathbf{G^T}(\mathbf{R}-\overline{\mathbf{Y}})] \quad (11)$$

Finally, the feedback characteristic of MPC is given as the first optimal control input $u^*(t) = u_{base}(t|t) + \delta u(t|t) = u_{base}(t|t) + U^*(1)$ is applied to the plant and then the whole procedure is repeated again at the next sampling instant $t+1$.

3. UAV: THE AR.DRONE 2.0 QUADROTOR

3.1 Plant Description

The AR.Drone 2.0 is a commercial and low-cost micro Unmanned Aerial Vehicle. The quadrotor comes with internal in-flight controllers and emergency features making it stable and safe to fly (Bristeau et al. (2011)). The only downside would be that access to the internal controller of the quadrotor is restricted. The internal software is black-box and the parameters that refer to control, motors and other calibrations are undocumented. There are 4 brushless DC motors powered with 14.5 W each from the 3 element 1000 mA/H LiPo rechargeable battery which gives an approximate flight autonomy of 10-15 minutes. Two video cameras are mounted on the central hull. The front camera resolution is 1280x720 and the bottom one is 640x360 with a video stream rate of 30 FPS and 60 FPS for front and bottom cameras.

The sensors are located below the central hull and consist of a 3-axis accelerometer, a 2-axis gyroscope and 1-axis gyroscope which together form the Inertial Measurement Unit (IMU). There is one ultrasonic sensor and one pressure sensor for altitude estimation. A 3-axis magnetometer gives the orientation of the quadrotor with respect to the command station. Communication between quadrotor and command station is done via Wi-Fi connection within a range of 30 m to 100 m for indoor and outdoor environment, respectively. The AR.Drone creates a Wi-Fi network, self-allocates a free IP address to grant access to client devices that wish to connect. For more details about internal structure of this quadrotor, check (Bristeau et al. (2011)).

A C++ application in Visual Studio establishes access to all AR.Drone communication channels, enabling functions to send commands or set configurations and also receive and store data from sensors and video stream. Thus, data can be interpreted off- or on-line for the purpose of identification, modeling and control of the quadrotor.

AR.Drone electronics execute an operative system to read the sensors, manipulate the speed of the motors, and to control the quadrotor in four degrees of freedom. We refer to this black-box on-board system as the low layer. The path-following controller located in the higher layer, sends references to the low-layer internal controllers through Wi-Fi communication. Figure 1 describes the two layers structure which characterize the system.

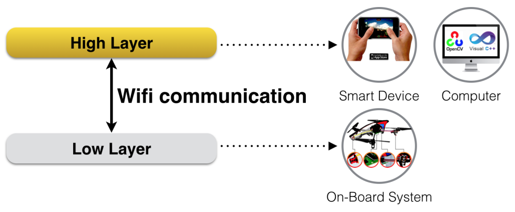

Fig. 1. Quadrotor layers: The low layer represents the electronic assistance and the embedded operative system on the AR.Drone, the high layer represents the pilot (natively a smart device i.e iPhone)

Movement is achieved by giving reference values as input to the internal, black-box controllers. The input and output relations will be discussed in the following subsection.

3.2 Coordinates System

The quadrotor aerial movements are similar to those of a conventional helicopter. The difference is that movement is achieved by varying each of the motor speeds to obtain the desired effect. Figure 2 depicts the movement axes of the quadrotor. The 4 Degrees Of Freedom (DOF) of the AR.Drone give attitude and position. Movements are thus achieved on:
• Pitch - By rotational movement along transverse axis y, translational movement on x axis is made.
• Roll - By rotational movement along longitudinal axis x, translational movement on y axis is made.
• Yaw - Rotational movement along z axis.
• Throttle - Translational movement on z axis.

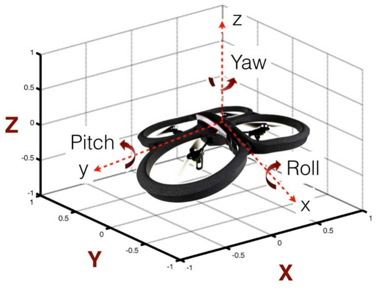

Fig. 2. Movement axes of the quadrotor in the space

3.3 Identified Dynamics

The control parameters given to the internal controllers are floating point values between [-1,1] and represent the percentage of the minimum or maximum configured value for the respective movement. We denote $\{\phi, \theta, \dot{z}, \dot{\psi}\}$ the roll angle reference value, pitch angle reference, vertical speed reference and yaw angular speed reference. The roll, pitch and yaw angles are given in radians, altitude in meters and linear velocities on longitudinal and transverse axes in m/s.

Due to the low layer internal control, the quadrotor behaves as a Linear Time-Invariant System. Making possible to perform for each degree of freedom a parametric identification using the prediction error method (Ljung (2007)). A Pseudo-Random Binary Signal (PRBS) is used to identify the dynamics of the quadrotor. A sampling time of 66 ms is chosen based on the analysis of dynamics performed on previous work (Vlas et al. (2013)). The obtained transfer functions are given by:

$$\begin{aligned} H_x(s) &= \frac{x(s)}{u_x(s)} = \frac{7.27}{s(1.05s+1)} \\ H_y(s) &= \frac{y(s)}{u_y(s)} = \frac{7.27}{s(1.05s+1)} \\ H_z(s) &= \frac{z(s)}{\dot{z}(s)} = \frac{0.72}{s(0.23s+1)} \\ H_{yaw}(s) &= \frac{\psi(s)}{\dot{\psi}(s)} = \frac{2.94}{s(0.031s+1)} \end{aligned} \tag{12}$$

4. CONTROL DESIGN

This section describes the path-following control strategy, which corresponds to the position control located in the higher layer. This controller will send the setpoints to the black-box internal controller in the low-layer. It is important to notice that due to this internal control, each degree of freedom in the quadrotor behaves independently, thus making possible to tune SISO controllers. The tuning procedure for the two considered strategies (i.e. the MPC and PD controllers) is described below.

4.1 MPC-EPSAC tuning

The MPC-EPSAC strategy is implemented in simulation for all degrees of freedom. The main specification was to achieve a fast response without overshoot. A combination of long prediction horizon (N_2) and short control horizon (N_u) was also considered in order to introduce a higher robustness in the controller (De Keyser (2003)). The tuned EPSAC parameters are summarized in table 1.

Table 1. EPSAC controller parameters.

SISO System	N_1	N_2	N_u	α	Noise Filter: C/D
x, y	1	15	1	0	$\frac{1}{1-q^{-1}}$
z	1	30	1	0	
yaw	1	10	1	0	

4.2 PD tuning

The performance of the MPC strategy is compared to the one achieved using PD controllers. In order for a PD controller to be practically usable, its derivative action must be filtered, given that a derivative action without a filter can produce undesired noise-effects on the system. The PD controller is represented in (13), where: N is the derivative gain limit; usually an integer number.

$$C_{(s)} = K_p + K_p \frac{T_d s}{\frac{T_d}{N} s + 1} \tag{13}$$

PD controllers with filter action (13) can be represented as one gain, one zero and one pole (14); this compensator can be designed with any CAD tool. In our case we made use of the Frequency Response (FRTool) (De Keyser and Ionescu (2006)) as it provides an intuitive graphical interface based on the Nichols plot, to tune a compensator based on design specifications such as: robustness, settling time, phase margin and/or gain margin.

$$C_{(s)} = K_p(N+1) \frac{s + \frac{N}{T_d(N+1)}}{s + \frac{N}{T_d}} \tag{14}$$

Once the controller is exported from FRTool in zpk form (15):

$$C_{(s)} = K \frac{(s+z_1)}{(s+p_1)} \tag{15}$$

The PD parameters K_p,T_d and N can be calculated by defining a system of equations from (14) and (15).

$$\begin{aligned} K &= K_p(N+1) \\ z_1 &= \frac{N}{T_d(N+1)} \\ p_1 &= \frac{N}{T_d} \end{aligned} \tag{16}$$

At this point, the controller is still in the continuous-time domain, therefore a discretization step is required. Proportional action is the same in continuous time and discrete time, however the derivative action have several approximations, in this work the Tustin approximation is considered.

$$F_D(z) = \frac{2NT_d}{2T_d + NT_s} \frac{1 - z^{-1}}{1 - \frac{2T_d - NT_s}{2T_d + NT_s} z^{-1}} \tag{17}$$

The digital form of the PD controller with filter and sample period T_s is given in (18), representing an implementable form of the PD controller.

$$C(z) = \frac{u(z)}{e(z)} = K_p + K_p \frac{2NT_d}{2T_d + NT_s} \frac{1 - z^{-1}}{1 - \frac{2T_d - NT_s}{2T_d + NT_s} z^{-1}} \tag{18}$$

The design specifications: robustness (Ro), settling time (T_{set}), overshoot percent ($\%OS$) and gains of the tuned PD controllers are presented in table 2.

Table 2. Design parameters for the PD controllers

Controller	Ro	T_{set}	$\%OS$	K_p	T_d	N
x,y	≥ 0.7	≤ 5 s	$\leq 3\%$	0.15	0.96	1
z	≥ 0.7	≤ 5 s	$\leq 3\%$	1.6	0.46	1
ψ	≥ 0.7	≤ 5 s	$\leq 3\%$	1.52	0.08	1

4.3 Simulation Results

In this subsection a tracking experiment is performed to test the capabilities of the controller to follow a setpoint. Special attention is paid to the settling time and overshoot, as this will limit the control performance for path-following. A faster controller allows to perform more aggressive maneuvers, whilst a smaller overshoot allows to more accurately follow the trajectory in confined spaces.

The performance of the combination of the controllers is depicted in Fig. 3, where the quadrotor is requested to follow four setpoints in the 3D space. The task consists in sequentially following the waypoints, starting at point 0. The MPC-EPSAC is able to follow more accurately the path without large deviations. In order to better understand the performance of the controllers, each one is analyzed separately.

For the case of orientation (Yaw) both controllers provide a desired behavior without overshoot, although MPC-EPSAC provides a faster response as observed in fig. 4. For the case of altitude it was observed no difference between the two control strategies (Fig. 5).

On the other hand for the case of translational movements over X and Y axis, a more distinguishable difference in the control performance of PD and MPC-EPSAC controllers is observed. The MPC-EPSAC reacts faster and without

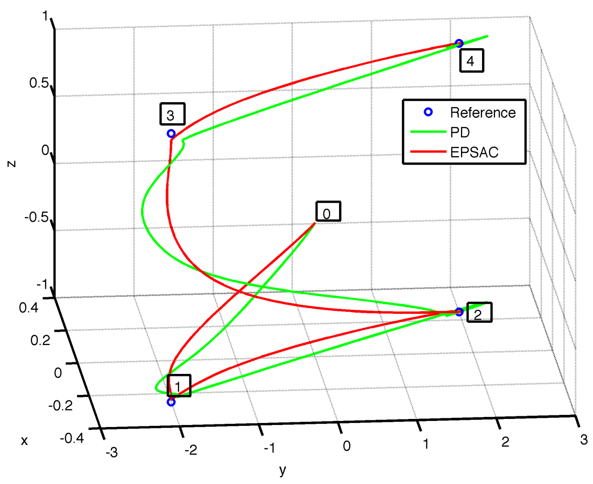

Fig. 3. 3D response for path-following of the AR.Drone 2.0. The markers represent the sequential waypoints for the quadrotor in the space.

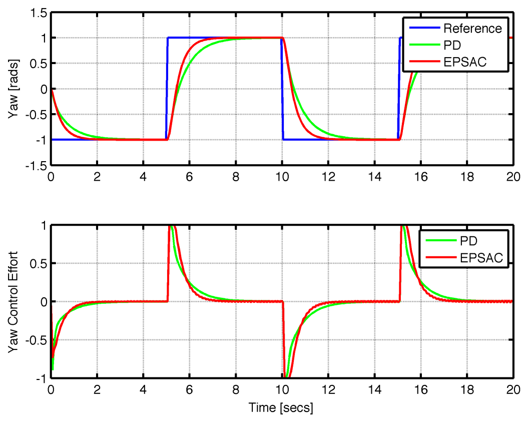

Fig. 4. Results Yaw controller

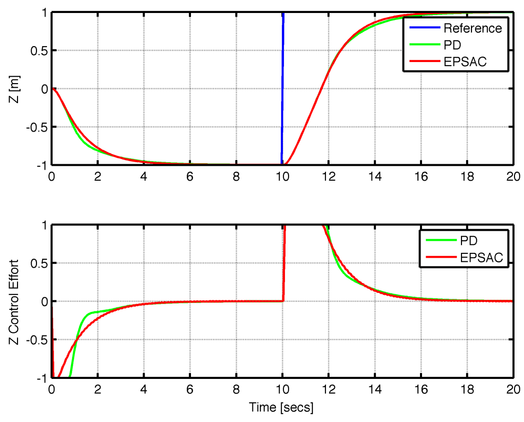

Fig. 5. Results controllers in Z axis

overshoot compared to the PD controllers, although at expenses of a slightly higher control effort as depicted in Fig. 6 and Fig. 7.

5. EXPERIMENTAL RESULTS

In this section the experimental results obtained for the proposed path-following control strategies are presented. The experiment consisting in following a trajectory in the two dimensional space is depicted in Fig. 8 for the case of the PD controllers and in Fig. 9 for the case of the MPC.

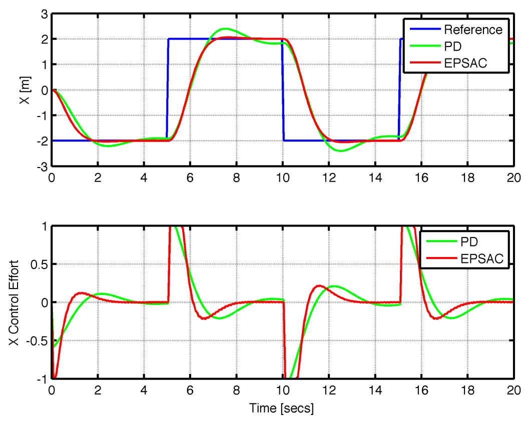

Fig. 6. Results controller X axis

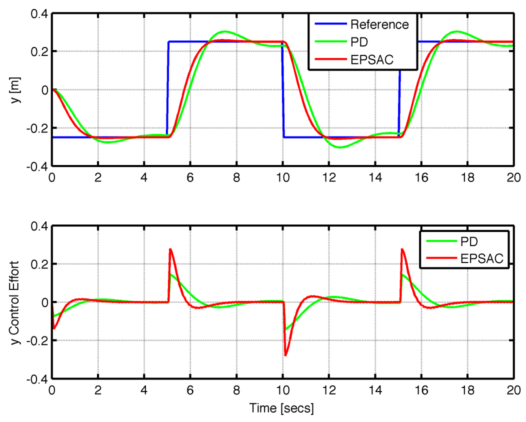

Fig. 7. Results controllers Y axis

In the case of the MPC-EPSAC path-following controller, the quadrotor experiences less deviations to the desired trajectory, specially at the moment of performing sharp bends as in the middle of the trajectory. The performance of the controllers is further compared using the well-known performance index ISE, IAE, and ITAE; which are used in this example to compute the error between the reference and the real trajectory described by the quadrotor. As expected the errors are less for the proposed MPC strategy as summarized in table. 3.

Table 3. Performance index for Path-following strategies

Controller	ISE	IAE	ITAE
PD X	118.12	159.99	1862.7
EPSAC X	**100.81**	**134.2**	**1279.6**
PD Y	87.28	164.65	2360.5
EPSAC Y	**68.89**	**131.68**	**1990.9**

6. CONCLUSION

We have presented in this work the application of Model Predictive path-following control using a low-cost commercial quadrotor. The strategy presents a desired performance as it is able to quickly follow the trajectory with little or no overshoot (compared to PD controllers),

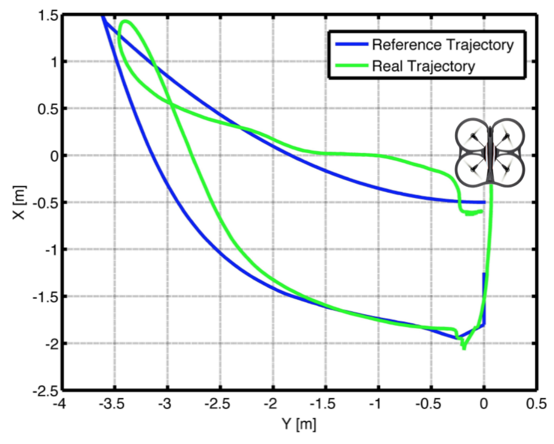

Fig. 8. Results obtained for the PD controllers following a trajectory in a two dimensional space.

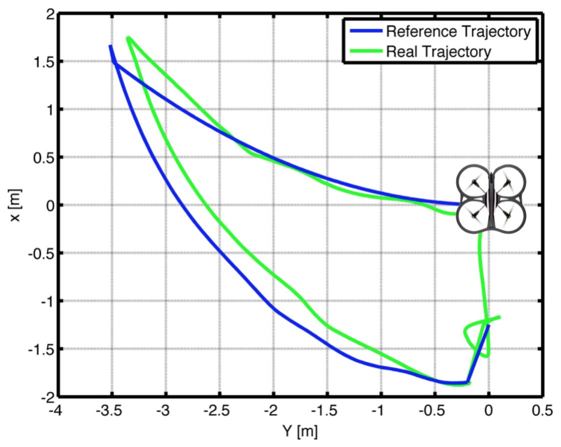

Fig. 9. Results obtained for the EPSAC controllers following a trajectory in a two dimensional space.

which makes it desired for the case of working in indoor or confined environments.

Future work includes a better position estimation (e.g. Kalman filter) and the implementation of multiple UAVs in a formation control structure.

REFERENCES

Aguiar, A. and Hespanha, J. (2003). Position tracking of underactuated vehicles. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, 1988–1993 vol.3. doi:10.1109/ACC.2003.1243366.

Alessandretti, A., Aguiar, A., and Jones, C. (2013). Trajectory-tracking and path-following controllers for constrained underactuated vehicles using model predictive control. In *European Control Conference (ECC13)*, 1371–1376.

Bednowitz, N., Batta, R., and Nagi., R. (2014). Dispatching and loitering policies for unmanned aerial vehicles under dynamically arriving multiple priority targets. *Journal of simulation*, vol:8 iss:1(ISSN: 1747-7778), pp. 9 –24.

Berni, J., Zarco-Tejada, P., Suarez, L., and Fereres, E. (2009). Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *IEEE Transactions on Geoscience and Remote Sensing*, vol.47, no.3, pp.722,738.

Bristeau, P., Callou, F., Vissiere, D., and Petit, N. (2011). The navigation and control technology inside the ar.drone micro uav. In *18th IFAC World Congress*, 1477–1484. Milano.

De Keyser, R. (2003). *Model based Predictive Control for Linear Systems*, chapter invited in UNESCO Encyclopaedia of Life Support Systems (EoLSS). UNESCO Encyclopaedia of Life Support Systems http://www.eolss.net Article contribution 6.43.16.1 (available online at: http://www.eolss.net/sample-chapters/c18/e6-43-16-01.pdf) Eolss Publishers Co Ltd, Oxford, 35 pages.

De Keyser, R. and Ionescu, C. (2006). A frequency response tool for cacsd in matlab. In *IEEE International Symposium on Computer Aided Control Systems Design*, 2275–2280. Munich.

Dijkshoorn, N. (2012). *Simultaneous localization and mapping with the AR.Drone*. Master's thesis, University of Amsterdam.

Ljung, L. (2007). *System identification: theory for the user.* Prentice-Hall.

Maciejowski., J. (2002). *Predictive Control: With Constraints.* Pearson Education. Prentice Hall.

Mokhtar, M., Matori, A., Yusof, H., Chandio, I., Viet, D., and Lawal, D. (2012). A study of unmanned aerial vehicle photogrammetry for environment mapping: Preliminary observation. In *Advanced Materials Engineering and Technology.*, volume Vol. 626 of *Advanced Materials Research*, p. 440 –444.

Sanchez, M. and Rodellar., J. (1996). *Adaptive Predictive Control.* ISBN 0135148618. Prentice Hall London.

Vlas, T., Hernandez, A., Copot, C., , Nascu, I., and De Keyser, R. (2013). Identification and path following control of an ar.drone quadrotor. In *System Theory, Control and Computing (ICSTCC), 2013 17th International Conference*, 583–588. doi: 10.1109/ICSTCC.2013.6689022.

Xargay, E., Dobrokhodov, V., Kaminer, I., Pascoal, A., Hovakimyan, N., and Cao, C. (2012). Time-critical cooperative control of multiple autonomous vehicles: Robust distributed strategies for path-following control and time-coordination over dynamic communications networks. *Control Systems, IEEE*, 32(5), 49–73. doi: 10.1109/MCS.2012.2205477.