

Identification and control of a quadrotor from experimental data

M.A. Lotufo, C. Perez-Montegro, L. Colangelo, E. Canuto, C. Novara

Abstract—The paper presents experimental results related to attitude control of a quadrotor rotating about a single axis. The Embedded Model Control (EMC) methodology is used for control design. Indeed, this methodology can be very effective in applications involving relevant disturbances, modeling errors and unknown nonlinearities, since it allows a cancellation of all these sources of uncertainty. EMC control design is performed from a detailed quadrotor model, where the actuator dynamics is identified from experimental data. The designed EMC controller is compared with a standard proportional-integral-derivative (PID) controller in several tests, carried out on a laboratory testbench.

I. INTRODUCTION

The work presented in this paper has been carried out within the Borea Project (see [1], [2]). This project aims to emulate, using an Unmanned Aerial Vehicle (UAV), guidance, navigation and control strategies to be employed in space applications. A key role in this kind of applications is played by the control algorithms, which should be properly designed to allow a realistic emulation of the spacecraft behavior.

A real UAV (more precisely, a quadrotor) was already constructed within the Borea project, see [1]. Preliminary studies and simulations, finalized at the design of the quadrotor control unit were also carried out, see [1], [2]. Based on these preliminary activities, three main contributions are provided in the present paper. The first one is a model identified from experimental data of the involved quadrotor actuators; this model is crucial for control design. The second one is an algorithm for the quadrotor single axial attitude control; the algorithm is based on the Embedded Model Control (EMC) methodology, [3], [4]. The third contribution consists in the presentation of the EMC control results obtained in several experimental tests, carried out on a laboratory single axial testbench, see Fig. 1.

To describe these contributions more in detail, consider that a quadrotor has four actuators. Each actuator is composed of three main elements (see also Fig. 2): an electronic speed control (ESC), a motor, a propeller. The motor makes the propeller rotate. This rotation transfers a certain amount of mechanical energy to the air, producing the thrust and torque which are used to drive the quadrotor. The ESC regulates the motor angular rate according to an angular rate reference. Overall, an actuator is a dynamic system with main input the angular rate reference, and main output the propeller thrust. Note that such a system has a nonlinear behavior, making control a quite challenging task, especially if difficult manoeuvres have to be executed with high precision.

The Authors are with the Dept. of Control and Computer Engineering, Politecnico di Torino, Italy. mauricio.lotufo@polito.it

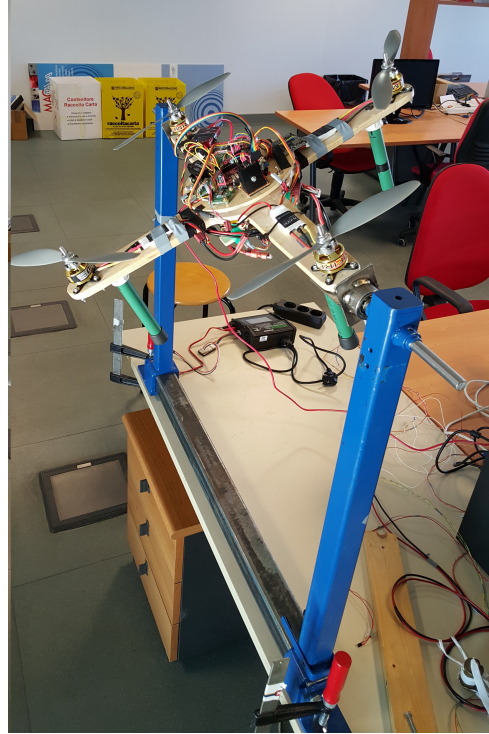


Fig. 1. Borea Quadrotor on testbench for single axial testing.

The first contribution of this paper is an actuator model, identified from data collected in experimental tests performed on a single actuator (under the reasonable assumption that all the actuators are similar). As it will be discussed in details later, the actuator model was identified considering a classical Hammerstein-Wiener structure, with a dynamic linear time invariant (LTI) system placed between two static nonlinear blocks.

The second contribution is a single axial attitude control algorithm, designed using the EMC methodology and the identified actuator model. EMC design involves the design of 1) an extended observer, finalized at predicting state values from gyroscope measurement; 2) a control law, able to perform active disturbance rejection. Thanks to these features, EMC can be very effective in situations involving relevant disturbances, modeling errors and unknown nonlinearities, allowing an efficient cancellation of these sources on uncertainty, [3], [4]. Note that the quadrotor control has been addressed using standard approaches in several papers, see, e.g., [5],[6]. However, for the above reasons, we believe that EMC may provide a better performance than standard methods, since a quadrotor is a system affected by several

uncertainties and nonlinearities.

The third contribution consists in presenting and discussing the experimental results of the EMC controller applied to the quadrotor mounted on the testbench. These results are also compared with those obtained by a standard proportional-integral-derivative (PID) control algorithm controller. The comparison shows that better results can be obtained using the EMC methodology, in terms of precision, disturbance rejection and oscillation attenuation.

II. ATTITUDE MODEL

In the considered experimental setting, the quadrotor body is free of rotating about only one axis. The command torque is provided by two actuators mounted on the two quadrotor's arms orthogonal to the axis of rotation. The arms are assumed rigid for simplicity. The distance between the two propeller axes is denoted by d and the overall moment of inertia around the axis of rotation is denoted by J . The numerical values of these parameters were estimated by means of a 3D CAD software (see Table IV). The thrust generated by the two actuators are indicated with $f_{p1}(t)$ and $f_{p2}(t)$, respectively. On the basis of these definitions, the quadrotor state equations are the following:

$$\begin{aligned} \dot{\omega}_b(t) &= \frac{d}{J} (f_{p2}(t) - f_{p1}(t)) \\ \dot{\theta}(t) &= \omega_b(t) \\ y(t) &= \theta(t) \end{aligned} \quad (1)$$

where θ and ω_b are the angular position and rate of the quadrotor, respectively. The actuator system, generating the thrusts f_{p1} and f_{p2} , was identified from experimental data, as described in the following section. Since all the actuators are identical, the index indicating a particular actuator will be omitted when not necessary, and the actuator thrust will be denoted simply by f_p .

III. ACTUATOR MODEL

A quadrotor actuator is composed of three main elements (see Fig. 2), which are responsible of the thrust generation necessary to make the vehicle fly. These components are an electronic speed controller (ESC), an electrical motor and a propeller. The motor makes the propeller rotate. This rotation produces the thrust and torque which are used to drive the quadrotor. The ESC regulates the motor angular rate according to a digital input reference.

The output of the actuator is the propeller thrust f_p . The command input is the angular rate reference, denoted with u . In order to take into account the battery influence, the battery voltage v_b was considered as an additional input of the actuator. All the quadrotor actuators were assumed identical.

From the available physical information, the actuator is known to have a Hammerstein-Wiener structure, with a dynamic LTI system placed between two static nonlinear blocks. As shown later, this model structure is also validated

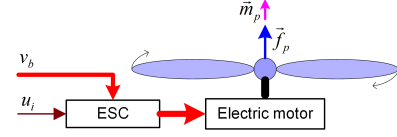


Fig. 2. Actuator main components scheme

by the experimental data. The actuator model equations are

$$\begin{aligned} z(t) &= g(u(t)) + k_V(v_{bmax} - v_b(t)) \\ \dot{\omega}_p(t) &= -p\omega_p(t) + pz(t) \\ f_p(t) &= f(\omega_p(t)) \end{aligned} \quad (2)$$

where the second equation defines the dynamic LTI system, while the first and third equations define the two static nonlinear blocks. As mentioned above, u is the command input, v_b is an additional input, and f_p is the output. Other quantities appearing in (2) are the following: z is an intermediate signal between the first nonlinear block and the LTI system (as it will be shown later, this variable can be interpreted as the steady-state propeller angular rate), ω_b is the propeller angular rate, $g(\cdot)$ and $f(\cdot)$ are static nonlinear functions, p is the pole of the LTI block, v_{bmax} is the full charge battery voltage, k_V is a parameter representing the battery voltage dependency.

The function $f(\cdot)$ is known, since it was studied and validated on experimental data in [7], see Fig. 4. The parameter v_{bmax} is known from the battery manufacturer and it is reported in Table IV. The function $g(\cdot)$, and the parameters k_V and p are not known and were identified from measurements of u , v_b and ω_b (see the next subsections). In other words, we identified the subsystem defined by the first two equations in (2). This is a classical Hammerstein system, formed by the cascade connection of a static nonlinearity and a dynamic LTI block. Note that, as it typically happens for this kind of systems, the intermediate signal z is not measured.

A. Static nonlinearity model

Following a standard approach, several experiments were carried out, where several constant command inputs u were considered. The command values were chosen within a sufficient wide range around the hovering point corresponding to a satisfactory quadrotor flight performance level. In each experiment, a constant command $u(t) = \bar{u}$, $\forall t$ was applied to the actuator and a set of measurements

$$\{\omega_p(kT_s), v_b(kT_s)\}_{k=1}^{1000} \quad (3)$$

was collected in steady-state conditions, using a sampling time $T_s = 0.02$ s.

To overcome the problem that z was not measured, the following equation was considered:

$$\omega_p(t) \underset{t \rightarrow \infty}{=} g(\bar{u}) + k_V(v_{bmax} - v_b(t)). \quad (4)$$

This equation holds since the LTI system in (2) has a unitary stationary gain, implying that, in steady-state conditions,

TABLE I
IDENTIFIED COEFFICIENTS

Coefficient	Value	Units
c_{g1}	447.1	rad/s
c_{g2}	1124.5	rad/s
k_v	40	rad/sV

$\omega_p(t) = z(t)$. From (4), it follows that the data (3) can be described as

$$\omega_p(kT_s) = g(\bar{u}) + k_v(v_{bmax} - v_b(kT_s)) + e(k) \quad (5)$$

where $e(k)$ accounts for possible noises, measurement errors and modeling errors. Defining $\Delta\omega_p(k) = \omega_p(kT_s) - \omega_p(T_s)$, $\Delta v_p(k) = v_b(kT_s) - v_b(T_s)$ and $\Delta e(k) = e(k) - e(1)$, we obtain the set of equations

$$\Delta\omega_p(k) = k_v \Delta v_p(k) + \Delta e(k), \quad k = 1, \dots, 3000. \quad (6)$$

An estimate of k_v was obtained from (6) by means of standard least squares. This operation was repeated for each experiment, resulting in a set of estimated values of k_v . The average of these values was taken as the final estimate of k_v (note anyway that the values of k_v estimated for different constant commands were very similar). The obtained value is reported in Table I.

The next step was the identification of the function $g(\cdot)$ in (2). To this aim, a polynomial form was assumed for this function. Several polynomial degrees were considered and the best trade-off between complexity and accuracy was achieved by a second order polynomial of the form

$$g(u) = c_{g1}u^2 + c_{g2}u. \quad (7)$$

To identify the parameters c_{g1} and c_{g2} , an experiment was performed, where a command signal generated as a sequence of positive and negative ramps was applied to the actuator. A set of measurements

$$\{\omega_p(kT_s), v_b(kT_s), u(kT_s)\}_{k=1}^{1000} \quad (8)$$

was collected, using a sampling time $T_s = 0.02$ s. The slope of the ramps was chosen to have the actuator working in steady-state conditions, so that equation (4) holds. Using (4), the data (8) can be described as

$$y_g(k) = c_{g1}u(kT_s)^2 + c_{g2}u(kT_s) + e_g(k) \quad (9)$$

where $y_g(k) = \omega_p(kT_s) - k_v(v_{bmax} - v_b(kT_s))$ and $e_g(k)$ accounts for possible noises, measurement errors and modeling errors.

The parameters c_{g1} and c_{g2} were estimated from (9) using standard least squares. The obtained values are reported in Table I.

The model was validated using as command input a sequence of ramps with slope different from that used to generate the identification data. The validation results are illustrated in Fig. 3.

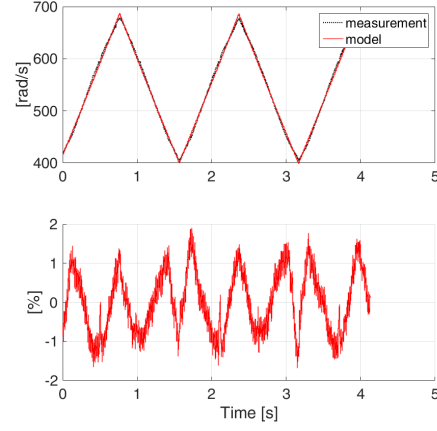


Fig. 3. Propeller angular rate measurement (top) and percentage simulation error (bottom)

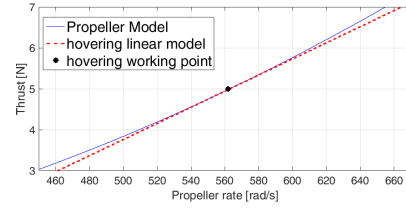


Fig. 4. Propeller thrust function

B. LTI system model

To identify the LTI system model, an experiment was carried out, where a pseudorandom binary sequence (PRBS) command input of duration 2400 s was applied to the actuator. A PRBS is easily generated and exhibits a statistical behavior similar to a truly-random sequence. The use of this sequence allowed us to excite the actuator up to the Nyquist frequency. From this experiment, a set of data

$$\{\omega_p(kT_s), z(kT_s)\}_{k=1}^{120000} \quad (10)$$

was collected, using a sampling time $T_s = 0.02$ s. $z(kT_s)$ was computed from the measured $v_b(kT_s)$ and $u(kT_s)$, using the model of the static nonlinear block identified in the previous subsection. The LTI model in (2) was identified from these data, using the function “Transfer Function Models” of the Matlab System Identification toolbox. The obtained value of the pole p in (2) is reported in Table IV.

The model was validated using a PRBS different from that used to generate the identification data. The validation results are illustrated in Fig. 5 in which the continuous time model is shown with the black line.

IV. CONTROL UNIT

The control unit aims to set the digital commands $u_1(i)$ and $u_2(i)$ in order to achieve the control requirements. A gyroscope mounted on the vehicle provides the angular rate $\omega_b(i)$ as input of the control unit. The EMC unit is composed by three sub systems. The reference generator which generates the reference command and the state trajectory to be

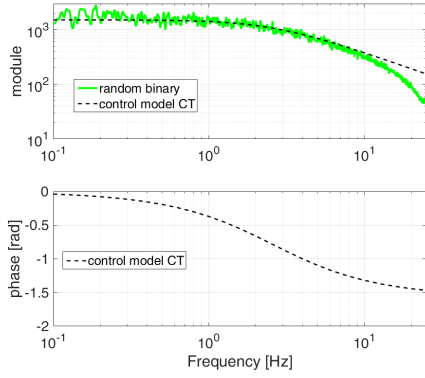


Fig. 5. Bode plots of mid frequency models

followed by the controllable states. This trajectory respects the system dynamics. The internal model or embedded model (EM) plus the noise estimator (NE) provides a prediction of the controllable and disturbance states. Finally, the control law set the commands based on the full state feedback plus the nominal command and the disturbance rejection. The reference generator is not discussed here because of attitude was studied in hovering condition, this implies the angle must be fixed to 0. Therefore, the research is focused on the development of the state predictor (EM+NE) and control law.

A. EMC Control Unit

Starting from the design model presented in the previous section, the EM has been derived. This is a discrete time model that represents the plant inside the control unit. The EM is divided into the dynamics controllable by the command and the dynamics of the disturbance acting on the controllable part.

1) *The Embedded Model (EM)*: the EM is coded into the control unit and has to be executed at each control step. Having a simple model allows to reduce the number of instructions, therefore, the execution time also is reduced. Simplification introduces model errors, that can be treated as disturbances. In the EMC these disturbances are estimated and canceled by means of state predictor and control law respectively.

As already mentioned, the propeller thrust is a nonlinear function of the angular rate and of the wind velocity. The thrust function $f_{pj}(t)$ is here expressed as sum of two components, nominal and error. The nominal component $f_{pn}(t)$ is equal for all actuators and it does not depend on wind velocity $v_{wind}(t)$. On the other hand, the error component $e_{pj}(\omega_{pj}, v_{wind}, t)$ includes neglected dynamics, parametric errors and wind effects. The nonlinear thrust function $f_p(\cdot)$ is expressed in (11).

$$f_{pj}(t) = f_{pn}(\omega_{pj}) + e_{pj}(\omega_{pj}, v_{wind}, t) \quad (11)$$

Combining (1) and (11) the general expression for attitude

dynamics becomes,

$$\dot{\omega}_b(t) = \frac{d}{J} (f_{pn}(\omega_{p1}) - f_{pn}(\omega_{p2})) + e_p(\omega_{p1}, \omega_{p2}, v_{wind}, t). \quad (12)$$

The linear approximation of the nominal thrust function $f_{pn}(\omega_{pj})$ around the hovering is performed in (13) and the consequent model errors are added to $e_p(\cdot)$.

$$f_{pn}(\omega_{pj}(t)) = c_{p1}\omega_{pj}(t) + c_{p0} \quad (13)$$

Substituting (13) in (12) the attitude dynamics is rewritten as

$$\dot{\omega}_b(t) = b(\omega_{p1} - \omega_{p2}) + e_p(\cdot) \quad (14)$$

where $b = dc_{p1}/J$. The values of c_{p0} and c_{p1} are reported in Table IV. In order to achieve an input-output relation is sufficient to make the time derivative of (14) without considering the model error $e_p(\cdot)$ and substitute the actuator dynamics in (2),

$$\dot{\omega}_b(t) = -p\dot{\omega}_b(t) + pb(g(u_1) - g(u_2)). \quad (15)$$

The model error is taken into account as a disturbance in the angular velocity. (15) includes actuator dynamics and represents a new state equation where $\dot{\omega}_b$ is a new state variable. Denoting the new state with $q_b(t) = \dot{\omega}_b(t)$ and defining the new command,

$$u_x(t) = g(u_1(t)) - g(u_2(t)) \quad (16)$$

input-output dynamics of the single axis model is reported in (17),

$$\begin{aligned} \dot{\theta}(t) &= \omega_b(t), \theta(0) = \theta_0 \\ \dot{\omega}_b(t) &= q_b(t), \dot{\omega}_b(0) = \dot{\omega}_{b0} \\ \dot{q}_b(t) &= -p\dot{\omega}_b(t) + pbu_x(t), q_b(t) = q_{b0} \\ y(t) &= \theta(t). \end{aligned} \quad (17)$$

In order to obtain the EM, the Euler forward discretization was applied. Controllable states are represented by the angle $x_{c1}(i) = \theta(i)$, the angular rate $x_{c2}(i) = \omega_b(i)$ and the equivalent actuator dynamics $x_{c3}(i) = q_b(i)$.

Second order disturbance dynamics is represented by a cascade of two digital adders driven by noise. Disturbances states denoted by $\mathbf{x}_d(i)$ are part of the EM. Adders contain state variables denoted by $x_{d1}(i)$ and $x_{d2}(i)$ (see Fig. 6). In addition, a noise is reported at the command level representing some electronic noise ($\bar{w}_0(i)$ in Fig. 6). The EM diagram is reported in Fig. 6 in which the symbol Σ represents digital adders. Noises are collected in the noise vector $\mathbf{w}(i) = [w_1 \ w_2 \ w_3 \ w_4]^T$. In the EM all the variables have the same unit. For this reason the measurement $\omega_b(i)$ is transformed in the dimensionless quantity $y(i) = T_c \omega_b(i)$ and a new command $u^*(i)$ is considered in (18).

$$u^*(i) = T_c^2 b u_x(i) \quad (18)$$

The actual plant input is the torque command $u_p(i)$. This torque is computed from (18) and reported in (19).

$$u_p(i) = J T_c^{-2} u^*(i) \quad (19)$$

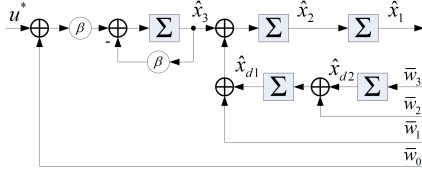


Fig. 6. Attitude Embedded Model

The EM state variables are written in the compact form reported in (20) and the state representation in (21).

$$\begin{aligned} \mathbf{x}(i) &= [\mathbf{x}_c(i) \quad \mathbf{x}_d(i)]^T \\ \mathbf{x}_c(i) &= [x_{c1}(i) \quad x_{c2}(i) \quad x_{c3}(i)]^T \\ \mathbf{x}_d(i) &= [x_{d1}(i) \quad x_{d2}(i)]^T \end{aligned} \quad (20)$$

The controllable system is represented with A_c matrix which defines the controllable dynamics, B_c which determines connection between the command $u^*(i)$ and controllable states.

$$\begin{aligned} \mathbf{x}(i+1) &= \begin{bmatrix} A_c & H_c \\ \mathbf{0} & A_d \end{bmatrix} \begin{bmatrix} \mathbf{x}_c \\ \mathbf{x}_d \end{bmatrix} (i) + \\ &+ \begin{bmatrix} B_c \\ \mathbf{0} \end{bmatrix} u^*(i) + \begin{bmatrix} G_c \\ G_d \end{bmatrix} \mathbf{w}(i) \\ \mathbf{y}_m(i) &= \begin{bmatrix} C_c & C_d \end{bmatrix} \mathbf{x}(i) \\ \mathbf{z}_m(i) &= \begin{bmatrix} F_c & \mathbf{0} \end{bmatrix} \mathbf{x}(i) \end{aligned} \quad (21)$$

where matrices for the controllable part are in (22). $\mathbf{z}_m(i)$ are the outputs on which control requirements are defined, for this case $\mathbf{z}_m(i) = \boldsymbol{\theta}(i)$.

$$\begin{aligned} A_c &= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1-\beta \end{bmatrix}, \quad B_c = \begin{bmatrix} 0 \\ 0 \\ \beta \end{bmatrix} \\ F_c &= [1 \quad 0 \quad 0], \quad C_c = [0 \quad 1 \quad 0] \end{aligned} \quad (22)$$

The second order model of the disturbance is capable to represent the discrepancies (neglected dynamics and parametric errors) between the EM and plant. Disturbance dynamics is represented by the A_d matrix while H_c matrix specifies the connection between the disturbances and the controllable states. The noise is entering according to G_c and G_d matrix. The uncertainty estimator developed in section IV-A includes the EM as core in the control unit.

$$A_d = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad H_c = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \end{bmatrix} \quad (23)$$

2) *The Noise Estimator (NE)*: Estimated variables are denoted with the hat symbol. The noise vector estimation $\hat{\mathbf{w}}(i)$ is performed by the NE starting from the discrepancies between the measurement and the estimation of the angular velocity. This quantity is called model error $e_m(i)$ and is reported in (24)

$$e_m(i) = \hat{x}_{c2}(i) - y(i) \quad (24)$$

The model error represents a key signal because contains the discrepancies between the plant and the EM. The set of EM plus NE is a state predictor, its block diagram is reported in

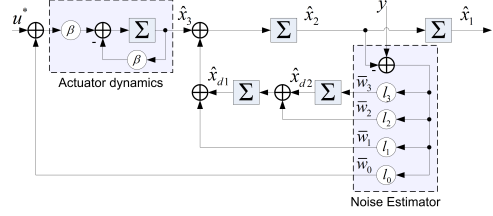


Fig. 7. EM + Noise Estimator

TABLE II
STATE PREDICTOR COMPLEMENTARY EIGENVALUES AND NE GAINS

Symbol	Value	Symbol	Value
γ_0	0.276	l_0	$-5.03e-17$
γ_1	0.120	l_1	$1.95e-1$
γ_2	0.052	l_2	$1.02e-2$
γ_3	0.023	l_3	$1.43e-4$

Fig.7. For this case the NE is composed by four static gains l_0, l_1, l_2 and l_3 .

The closed-loop state equation is reported in (25).

$$\begin{aligned} \begin{bmatrix} \hat{x}_{c2} \\ \hat{x}_{c3} \\ \hat{x}_{d1} \\ \hat{x}_{d2} \end{bmatrix} (i+1) &= \begin{bmatrix} 1-l_1 & 1 & 1 & 0 \\ -\beta l_0 & 1-\beta & 0 & 0 \\ -l_2 & 0 & 1 & 1 \\ -l_3 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{c2} \\ \hat{x}_{c3} \\ \hat{x}_{d1} \\ \hat{x}_{d2} \end{bmatrix} (i) + \\ &+ \begin{bmatrix} 0 & l_1 \\ \beta & \beta l_0 \\ 0 & l_2 \\ 0 & l_3 \end{bmatrix} \begin{bmatrix} u^* \\ y \end{bmatrix} (i) \end{aligned} \quad (25)$$

The characteristic polynomial $p(\gamma)$ for the state predictor system in (25) is equal to

$$\begin{aligned} p(\gamma) &= \gamma^4 + (\beta + l_1)\gamma^3 + (l_2 + \beta l_1 + \beta l_0)\gamma^2 + \\ &+ (l_3 + \beta l_2)\gamma + \beta l_3 \end{aligned} \quad (26)$$

where $\gamma = \lambda - 1$ is the complementary eigenvalue. Computing the NE gains is straightforward after fixing complementary eigenvalues in (26) through pole placement. First complementary eigenvalue γ_0 was located in the same frequency of p , because actuator dynamics has not to be modified, otherwise, command can be saturated. Remaining complementary eigenvalues were fixed according to the algorithm reported in (27),

$$\gamma_k = \gamma_0 \cdot 2^{-k\alpha}, \quad k = 0, 1, 2, 3, \quad \alpha \geq 0 \quad (27)$$

where α is the only parameter to be tuned. Choosing eigenvalues according to (27) allows to avoid overshooting in the sensitivity function of the state predictor. For $\gamma_0 = \beta = 0.276$ and $\alpha = 1.2$ the complementary eigenvalues and the NE gains are in Table II.

3) *Control Law*: the command is divided in three components; first the command reference $u(i)$, second component is the full state feedback denoted as $u_{fb}(i)$, and the third component is due to disturbance rejection $Mx_d(i)$. In (28) the generalized tracking error $e(i)$ is defined and the command expression is shown. The feedback gains collected in K are

TABLE III
STATE FEEDBACK AND PID GAINS

Symbol	Value	Symbol	Value
k_1	0.276	k_p	2.248
k_2	0.120	k_I	0.985
k_3	0.052	k_D	0.574

defined in order to stabilize A_{cc} . The parameter γ_0 is the same for state predictor and control law because is imposed by the actuator, instead α_c allows to have the control law bandwidth wider than state predictor, for these case $\alpha_c = 0$.

$$\begin{aligned} e(i) &= x_c - \hat{x}_c - Qx_d(i) \\ u_{fb}(i) &= Ke(i) \\ A_{cc} &= A_c - B_cK \\ u^*(i) &= \underline{u}(i) + u_{fb}(i) - Mx_d(i) \end{aligned} \quad (28)$$

Disturbance rejection is made through matrices M and Q and they are calculated with Silvester equation as reported in [3]. M and Q matrices are reported in (29).

$$M = \begin{bmatrix} 1 & 1/\beta \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}^T \quad (29)$$

The gain vector $K = [k_1 \ k_2 \ k_3]$ is reported in Table III on the left side.

B. PID Control Unit

A PID control was implemented in order to make a preliminary comparison with EMC controller. The model used to develop the controller was the same for both control units. The PID command is reported in (30). The tracking error is equal to $e(i) = \underline{y}(i) - y(i)$. The reference $\underline{y}(i)$ is zero as the EMC case. The integrative state is represented by $x_I(i)$ and its dynamic equation is $x_I(i+1) = x_I(i) + e(i)$.

$$u(i) = k_p e(i) - \frac{k_D}{T_c} x_2(i) + k_I T_c x_I(i) \quad (30)$$

The PID gains were computed by means of gain and phase margin evaluation. Also pole placement technique was considered. PID gains are reported in Table III on the right side.

V. EXPERIMENTAL RESULTS

Table IV shows the main parameters of the test, these includes the values obtained in the identification process in section III. Tests consist of controlling the angle to zero and apply several torque impulses manually as external disturbance. The angle tracking error is showed in Fig. 8. On the graph impulses are not synchronized because PID and EMC controllers were performed in different tests. PID shows several oscillations in response to the torque impulses.

Disturbance estimation is reported in Fig. 9 where for the PID the integrative state is considered. The impulse disturbance is effectively estimated in the EMC controller and this is reflected on the angle tracking with no oscillations, fast response and a small overshoot. On the other hand PID controller shows oscillations in the angle tracking response and a flat integrative command component.

TABLE IV
DESIGN MODEL PARAMETERS

Parameter	Symbol	Value	Units
Actuator Pole	p	16	rad/s
Discrete Pole	β	0.27	rad
Inertia	J	0.032	Kgm^2
Arm Length	d	0.25	m
Battery Constant	k_V	40	rad/Vs
Control Time Unit	T_c	0.02	s
Max. Battery voltage	v_{bmax}	12.6	V
Propeller coefficient	c_{p1}	0.0197	Ns/rad
Propeller coefficient	c_{p0}	-6.0771	N

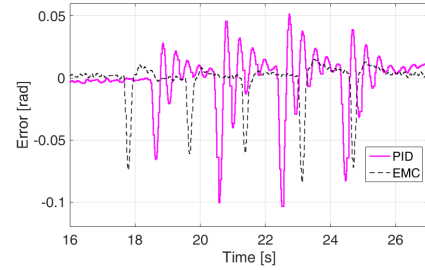


Fig. 8. Tracking error on the angle for PID and EMC

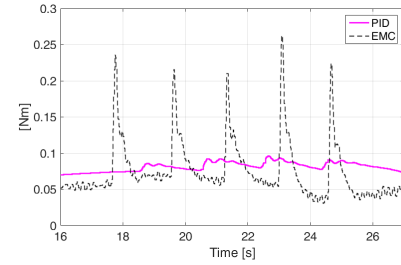


Fig. 9. Disturbance estimation

VI. CONCLUSIONS

The paper presents the development of a control strategy for a test bench based on EMC technique, some experimental results are shown where the performance is evaluated, and preliminary comparison with a PID controller is made.

REFERENCES

- [1] C. Perez-Montenegro, M. Lotufo, and E. Canuto, "Control Architecture and Simulation of the Borea Quadrotor," in *Research, Education and Development of Unmanned Aerial Systems*, vol. 2, pp. 168–173, nov 2013.
- [2] M. A. Lotufo, L. Colangelo, C. Perez-Montenegro, and E. Canuto, "The feedback linearisation method for Embedded Model Control: The Borea project case-study," in *2015 23rd Mediterranean Conference on Control and Automation (MED)*, pp. 501–507, IEEE, jun 2015.
- [3] E. Canuto, "Embedded Model Control: outline of the theory," *ISA transactions*, vol. 46, pp. 363–77, jun 2007.
- [4] E. Canuto, C. P. Montenegro, L. Colangelo, and M. Lotufo, "Embedded Model Control: Design separation under uncertainty," in *Proceedings of the 33rd Chinese Control Conference*, pp. 3637–3643, IEEE, jul 2014.
- [5] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robotics and Automation Magazine*, vol. 19, no. 3, pp. 20–32, 2012.
- [6] P. Pounds, R. Mahony, and P. Corke, "Modelling and Control of a Quad-Rotor Robot," *East*, vol. 4, p. 44, 2006.
- [7] R. Deters and M. Selig, "Static testing of micro propellers," *26th AIAA Applied Aerodynamics Conference*, no. August, 2008.