Model-Free Control applied for position control of Quadrotor using ROS

Z. Chekakta, M. Zerikat
École Nationale Polytechnique d'Oran
Maurice Audin LAAS Laboratory,
Oran, Algeria.
Email: Zakaria.Chekakta@gmail.com
zerikat@yahoo.fr

Y. Bouzid
École militaire polytechnique,
CSCS laboratory,
Algiers, Algeria.
Email: yasseremp@gmail.com

M. Abderrahim
University Carlos III of Madrid,
Department of Systems,
Engineering and Automation
Leganes 28911, Spain,
Email: Mohamed@ing.uc3m.es

***Abstract*—In this paper, an intelligent PD (iPD) controller named Model-Free Controller is proposed. Herein, it is used to control the position of a Vertical Take-Off and Landing (VTOL) Unmanned Aerial Vehicle (UAV) Quadrotor. This strategy increases the control performance as well as its robustness level with respect to the classical PD. This is due to the estimation principle provided by the ultra-local model that estimate the unknown disturbances and uncertainties each iteration. The efficiency of the proposed strategy is shown through various numerical simulations where a thorough analysis is provided. Moreover, a comparison study is elaborated between the iPD and the classical PD.**

***Keywords*—Model-Free Control;Vertical Take-Off and Landing Vehicle; PD controller; robust control; Ultra local model.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have recently turned into an important element in civil and military operational tasks, scientific and public purposes. Depending on various user needs, many types of UAVs are under current development attracting the attention of researchers around the world as for instance [1]. The quadrotor is the most popular configurations of multi-rotors UAVs. In the last decade, the interest of Quadrotors has grown even higher and witnessed a very large number of scientific and technical publications. This is probably due to some of the promising attributes of Quadrotors such as mechanical structure simplicity, hovering capability, high maneuverability, etc.

Several control techniques and algorithms have been formulated for Quadrotors, which are categorized usually into three families [2]:

- Linear control techniques: numerous linear control techniques are successfully implemented by many researchers for attitude stabilization as well as the trajectory tracking issues. The linear-quadratic regulator (LQR) was accomplished in OS4 by Bouabdallah et al [3]. Moreover, the LQR based tuning method is proposed in [4], to improve the capabilities of a PID controller. Gain scheduling technique in incorporation with other control algorithms have also been implemented on Quadrotor [5].
- Nonlinear control methods: hundreds of nonlinear control strategies have been proposed to deal with the nonlinear dynamics of the quadrotor. Benallegue et al. presented a simple feedback linearization controller [6], which is mixed with $GH\infty$ controller in [7]. Bouabdallah & Siegwart introduced a backstepping control scheme considering the hierarchical form of the quadrotor. The same authors proposed a sliding mode based control architecture for the attitude stabilization [8]. Many other techniques are proposed where a complete review is given in [2].
- Intelligent control strategies: learning and intelligent flight control algorithms have been applied successfully to Quadrotor. Robust adaptive-fuzzy controller for stabilization of Quadrotor has been introduced by Coza [9].

We stress that each control algorithm has its advantages and disadvantages (for more details the reader may refer to [2]. However, it is evident that PID controllers are very popular in industrial processes, due to their simplicity of implementation, even though they are sometimes poorly or approximately tuned especially for nonlinear systems or systems with uncertainties. However, most of the non-linear controller techniques need precise knowledge of the system dynamics. For Quadrotor, the exact knowledge of the kinematics and dynamics is hard to achieve due to the high coupling, uncertainties and external disturbance. To handle this problem, recently, a new technique named model-free control have been introduced to the literature [10], and several researchers have successfully implemented this technique for a Quadrotor [11],[12], [13].

Paper contributions: The paper proposes a comparison between a simple PD and iPD. Moreover, a deep study of the robustness level of the iPD is highlighted even in presence of disturbances. As another contribution, an implementation of the controller is achieved using a realistic simulator framework RotorS Gazebo MAV working under the ROS environment.

The paper is organized as follows: In Section 2 the dynamic model of the quadrotor is presented. The proposed control method is described in Section III. Several numerical simulations to demonstrate the efficiency of the proposed strategy is presented in Section IV. Concluding remarks are give in Section V.

II. Dynamic Model of Quadrotor

A. Motion of Quadrotor

The quadrotor has two couples of rotors (one, three) and (two, four) spinning in opposite directions. By varying the rotors speed, one can change the lift force and create a movement. Vertical motion can be performed by synchronous variation in rotors speed. Left and right motion can be accomplished by invert change of rotor speed two and four. In a similar way forward and backward motion can be achieved by invert change of rotor speed one and three. The yaw motion is achieved by varying the speeds of a couple of rotors[14]. This flight principle is summarized in Figure 1.

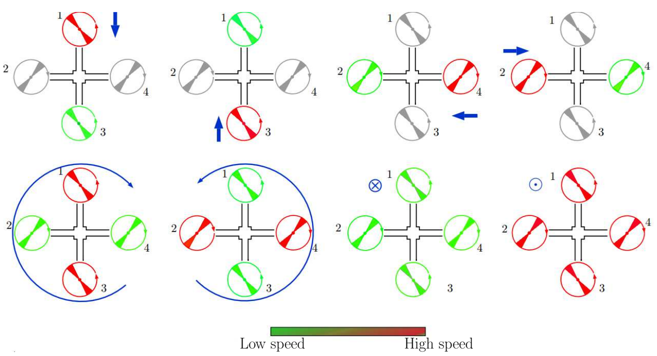

Fig. 1. Quadrotor flight principle

B. Dynamic model

By using Euler-Lagrange formulation, the dynamic model of the quadrotor is derived, where some assumption should be made:

- The body of the vehicle is rigid and symmetrical;
- The center of gravity of the vehicle coincides with the origin of the body frame.

Let $I = \{x_i, y_i, z_i\}$ denotes the inertial frame, and $B = \{x_B, y_B, z_B\}$ denotes the body fixed frame (see Figure 2). The

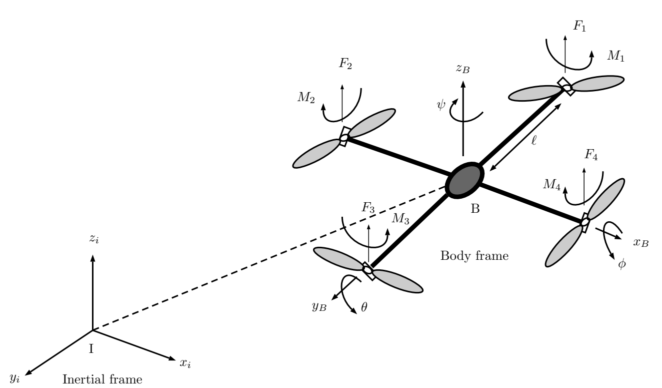

Fig. 2. Forces and Moments on the quadrotor and relative coordinate systems

position of the quadrotor $r = [x \;\; y \;\; z]^T$ can be described using the distance between the inertial frame and center of masse of the vehicle, and the orientation will be obtained by utilizing the rotational matrix R_{ot} from the body frame to the inertial frame. The order of rotation used is yaw (ψ) followed by pitch (θ) followed by roll (ϕ) around the z, y and x axes respectively.

$$Rot = \begin{bmatrix} c\psi c\theta & s\phi s\theta - c\phi s\psi & s\phi s\psi + c\phi c\psi s\theta \\ c\theta s\psi & c\phi c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - c\psi s\phi \\ -s\theta & c\theta s\phi & c\phi c\theta \end{bmatrix} \quad (1)$$

where: $c = \cos()$ and $s = \sin()$

The dynamics model consists of the rotational and translational motions. The translational dynamics are underactuated and written as follows:

$$m\ddot{r} = [\begin{array}{ccc} 0 & 0 & -mg \end{array}]^T + Rot \cdot F_B \quad (2)$$

Where m and $g = 9.81 m/m^2$ are respectively the mass of Quadrotor and gravitational acceleration. F_B non gravitational forces acting on the quadrotor in the body frame and given by:

$$F_B = \begin{bmatrix} 0 \\ 0 \\ \Gamma \end{bmatrix} \quad (3)$$

Where Γ denotes the global thrust.

The rotational dynamics are fully actuated, it can be described by equation (4),

$$J\dot{\omega} + \omega \times J\omega + \omega \times [\begin{array}{ccc} 0 & 0 & J_r\Omega_r \end{array}]^T = M_B \quad (4)$$

where J is the diagonal inertia matrix, w is the angular body rates, J_r is the inertia of the rotors and Ω_r their relative speed. The moments M_B acting on the quadrotor in the body frame are given by:

$$M_B = \begin{bmatrix} M_\phi \\ M_\theta \\ M_\psi \end{bmatrix} \quad (5)$$

Near to the equilibrium state, small angles assumption is made where $\cos\phi \approx 1$, $\cos\theta \approx 1$ and $\sin\phi \approx \sin\theta \approx 0$. The relation between the Euler rates in inertial frame $\dot{\eta} = [\dot{\phi} \;\; \dot{\theta} \;\; \dot{\psi}]$ and the angular body rates $\omega = [p \;\; q \;\; r]$ is:

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} \dot{\phi} - s\theta\dot{\psi} \\ c\phi\dot{\theta} + s\phi c\theta\dot{\psi} \\ -s\phi\dot{\theta} + c\phi c\theta\dot{\psi} \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (6)$$

Therefore, the control input vector is defined as:

$$\mathbf{U} = [\begin{array}{cccc} U_1 & U_2 & U_3 & U_4 \end{array}]^T \quad (7)$$

Where :

$$\begin{cases} U_1 = \Gamma \\ U_2 = M_\phi \\ U_3 = M_\theta \\ U_4 = M_\psi \end{cases} \quad (8)$$

This selection of the input's control U decouples the rotational system, where U_1 will generate the desired altitude, U_2 will produce the desired roll, U_3 will produce the desired pitch, and U_4 will produce the desired yaw.

Rewriting the complete mathematical model of the quadrotor to have the accelerations in terms of the other variables, results in:

$$\begin{cases} \ddot{x} = \frac{U_1}{m}(c\phi s\theta c\psi + s\phi s\psi) \\ \ddot{y} = \frac{U_1}{m}(c\phi s\theta s\psi - s\phi c\psi) \\ \ddot{z} = \frac{U_1}{m}(c\phi c\theta) - g \\ \ddot{\phi} = \frac{J_y - J_z}{J_x}\dot{\theta}\dot{\psi} + \frac{U_2}{J_x} - \frac{J_r}{J_x}\dot{\theta}\Omega_r \\ \ddot{\theta} = \frac{J_z - J_x}{J_y}\dot{\phi}\dot{\psi} + \frac{U_3}{J_y} - \frac{J_r}{J_y}\dot{\phi}\Omega_r \\ \ddot{\psi} = \frac{J_x - J_y}{J_z}\dot{\phi}\dot{\theta} + \frac{U_4}{J_z} \end{cases} \tag{9}$$

The quadrotor's parameters are given in Table I.

TABLE I. QUADROTOR PARAMETERS

Parameter	Symbol	Value	Unit
Inertia on x-axis	J_x	0.007	kg m^2
Inertia on y-axis	J_y	0.007	kg m^2
Inertia on z-axis	J_z	0.012	kg m^2
Mass of Quadrotor	m	0.716	kg
Arm length	ℓ	0.17	m
Thrust coefficient	k_f	8.54858×10^{-6}	N kg s^2
Drag coefficient	k_M	0.016	Nm kg s^2

III. CONTROL DESIGN

In this section, two control techniques are presented for the position controller of the quadrotor. The first one is the classical Proportional-Derivative (PD) controller while the second one is the Model-Free Control MFC (iPD).

A. Classical PD controller

PD controller is very effective and practical for industrial purpose, due to its simplicity in term of tuning parameters and implementation, where the gains are easy to adjust. The PD controller has the following equations form:

$$u(t) = k_p e(t) + k_d \frac{d}{dt} e(t) \tag{10}$$

Where $e(t) = r(t) - y(t)$ denotes the tracking error between the reference trajectory $r(t)$ and the output $y(t)$ and $u(t)$ denotes the control input. k_p and k_d are the proportional and derivative gains respectively.

- Altitude controller: The altitude control input of the quadrotor has the following equations:

$$\Gamma = (g + u_1)\frac{m}{c\phi c\theta} \tag{11}$$

  Where the derived control law is as follows

$$u_1 = k_{pz}(z_{des} - z) + k_{dz}(\dot{z}_{des} - \dot{z}) \tag{12}$$

- Translational x and y controller: The outputs of PD controller are defined by:

$$\begin{cases} u_x = k_{px}(x_{des} - x) + k_{dx}(\dot{x}_{des} - \dot{x}) \\ u_y = k_{py}(y_{des} - y) + k_{dy}(\dot{y}_{des} - \dot{y}) \end{cases} \tag{13}$$

Herein, the desired roll ϕ_d and pitch θ_d angles can be generated from the equation:

$$\begin{cases} \phi_d = \frac{1}{g} \times (u_x \cos(\psi) - u_y \sin(\psi)) \\ \theta_d = \frac{1}{g} \times (u_x \sin(\psi) + u_y \cos(\psi)) \end{cases} \tag{14}$$

The global control architecture of the quadrotor multi-rotor system is presented in Figure 3

B. Model-Free Controller

In general a nonlinear system v^{th} order single input single output variable can be represented in the following form:

$$x^{(v)} = f(x, \dot{x}, \ldots, x^{(v)}) + au \tag{15}$$

Where, $f(\cdot)$ is the model dynamics of the system, u is the input's system, and a is input factor wich is unkonwn.

Generally, the idea of model-free control is to locally approximate the above system by using a simple local model

$$x^{(v)} = F + \lambda u \tag{16}$$

The ultra-local model F also compensate the unknown and neglected dynamics of the model and uncertainties at time t, and it can be estimated by the following equation:

$$\widehat{F} = x^{(v)} - \lambda u \tag{17}$$

where:

x the last measured output;

$x^{(v)}$ is the derivative of order $v \geq 1$ of x. The integer v is selected by the operator. The most examples show that $v \in \mathbb{N}$ should always be chosen lower at possible, in example v equal to 1, or, rarely equal to 2. See [10] for more details;

$\lambda \in \mathbb{R}$ is a constant to estimate of the unknown factor a. It is chosen by the operator to achieve certain control performance;

u is last applied control input;

The estimation $\widehat{F}$ of the ultra-local model should be updated online during the control process for each iteration of the closed-loop system. Otherwise, the model-free control use one of the feedback controllers, herein the classical Proportional-Derivative (PD) controller has been chosen due to their simplicity and effectiveness in the control algorithm, and it has the following form:

$$u_c = k_p e + k_d \dot{e} \tag{18}$$

Where $e = x - x_d$ denotes the tracking error between the reference x_d and the output x.

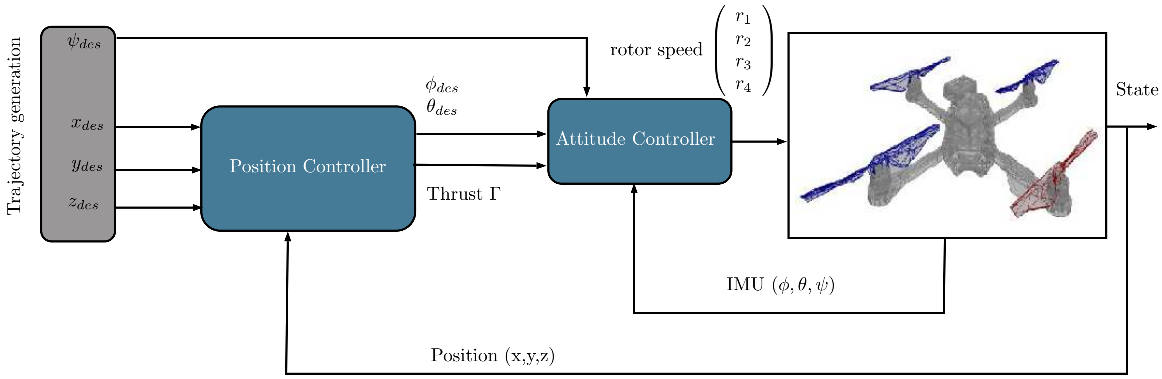

Fig. 3. Quadrotor control architecture

Lastly, the global control input u for model-free controller shown in Figure 4 is given by:

$$u = -\frac{\widehat{F} - x_d^{(v)} + u_c}{\lambda} \tag{19}$$

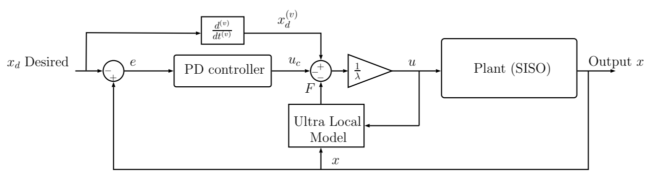

Fig. 4. Model-Free Control scheme

The value of v, chosen according to the system stability and the type of feedback controller used in the system. For instant, iPD would use a second-order system with $v = 2$ [10]. Which is yield to:

$$F = \ddot{x} + \lambda u \tag{20}$$

Finally, the control input for iPD is:

$$u = -\frac{\widehat{F} - \ddot{x}_d + k_p e + k_d \dot{e}}{\lambda} \tag{21}$$

Stability of the model-free control: Combining equations, (20) and (21), we obtain the following relation:

$$\ddot{x} = \ddot{x}_d + F - \widehat{F} + k_p e + k_d \dot{e} \tag{22}$$

We define e_{est} the error of estimation:

$$\ddot{e} + k_p e + k_d \dot{e} + e_{est} = 0 \tag{23}$$

Where $e_{est} \in \mathbb{R}$ is time-varying. In each short time interval h:

$$e_{est} = |\ddot{x} - \widehat{\ddot{x}} - \lambda(u(t) - u(t - \epsilon))| \leqslant |\frac{d\widehat{y}^{(v)}}{dt} h| + |\frac{d\widehat{u}}{dt} h| \tag{24}$$

In a physical system, $\frac{d\widehat{y}^{(v)}}{dt}$ and $\frac{d\widehat{u}}{dt}$ are usually bounded and $\neq \infty$, $\frac{d\widehat{y}^{(v)}}{dt} \leqslant \epsilon_{x^{(v)}}$, $\frac{d\widehat{u}}{dt} \leqslant \epsilon_u$ where the measure $\widehat{x}^{(v)}$ is supposed noise free or filtred, if a rather small time interval h whith respect to system dynamic selected,e_{est} will be bounded in small limit$|e_{est}| \leq \epsilon$, $\epsilon \geq 0$. Therefore, by choosing convenient gain k_p and k_d, the error in equation (24) will converge towrd this small limit ϵ. Thus, the system is practically stable.

The model-free control for position x, y and z is given by the following set of equations:

$$\begin{aligned} \Gamma &= -\tfrac{1}{\lambda_z}(\widehat{F}_z - \ddot{z}_d + k_{pz} e_z + k_{dz} \dot{e}_z) \\ u_x &= -\tfrac{1}{\lambda_x}(\widehat{F}_x - \ddot{x}_d + k_{px} e_x + k_{dx} \dot{e}_x) \\ u_y &= -\tfrac{1}{\lambda_y}(\widehat{F}_y - \ddot{y}_d + k_{py} e_y + k_{dy} \dot{e}_y) \end{aligned} \tag{25}$$

Where :

$$\begin{cases} \widehat{F}_x = \widehat{\ddot{x}} - \lambda_x u_x \\ \widehat{F}_y = \widehat{\ddot{y}} - \lambda_y u_y \\ \widehat{F}_z = \widehat{\ddot{z}} - \lambda_z u_1 \end{cases} \tag{26}$$

IV. Results and Discussion

In this section, we present the results of the techniques presented in Section III. Further, for ROS integration, we use the framework RotorS Gazebo MAV Simulator [15], which allows the use of the same controller including their parameters in the simulation world as with the real system. The MFC was implemented using C++ to interface the controller to ROS, the PID attitude controller node should receive the roll and pitch desired angle in addition to the thrust, and calculate the speed of each rotor, herein the attitude PID controller node used in this work was previously implemented by Kamal [16]. The control gains are given in Table II

TABLE II. Controller gains (P: parameter)

P	PD	iPD	P	PD	iPD	P	iPD
k_{px}	1.5	6.7	k_{dx}	4	11.5	λ_x	10
k_{py}	1.5	6.7	k_{dy}	4	11.5	λ_x	10
k_{pz}	3	8.3	k_{dz}	2.4	4.6	λ_x	5

A. Hovering test and disturbance rejection

In the first simulation, the quadrotor is commanded to stabilize itself at altitude $z_{des} = 1m$. After the quadrotor reaches the reference, we apply an external force $F_{ext} = 100N$ along z-axis as can be seen in Figure 5(b), The results of the two techniques are shown in Figure 5(a). We can observe that the model-free control has the capability of disturbance rejection better than classical PD. (see https://youtu.be/cUQBeEglCYU)

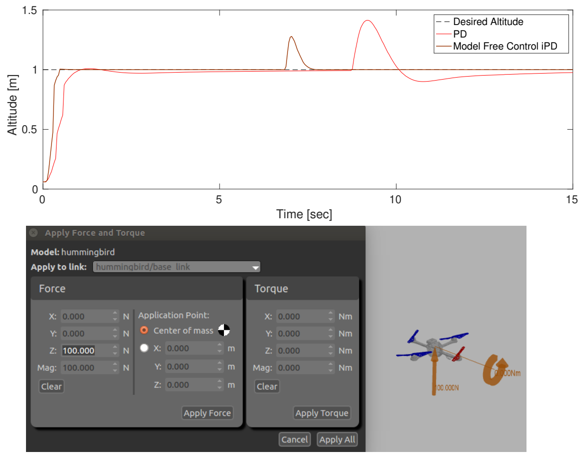

Fig. 5. (a) Altitude response, (b) Screenshot taken from Gazebo

B. Set point in presence of an extra payload

In the second simulation, we gave as a reference a coordinate points ($x_{des} = -1, y_{des} = 1, z_{des} = 2$). The curves shown in Figure 6, demonstrate the MFC exhibit satisfactory results with respect to classical PD.

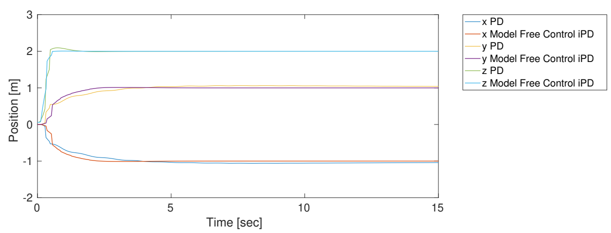

Fig. 6. Set point position response

Figure 7 shows the response of two control techniques in presence of an extra payload $+0.2\ kg$, It can be seen that the performance of the MFC is significantly better when compared with the PD.

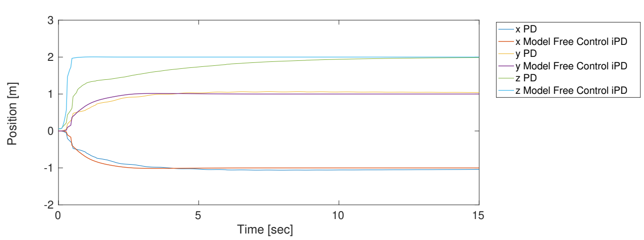

Fig. 7. Set point position response in presence of an extra payload

C. Trajectory tracking

The third simulation was performed considering a square trajectory ($2\ m \times 2\ m$) showed in Figure 8.

Furthermore, we plot the 2D trajectory response in Figure 9, where it is clear that the Model-Free Control iPD ensures the good following of the desired reference.

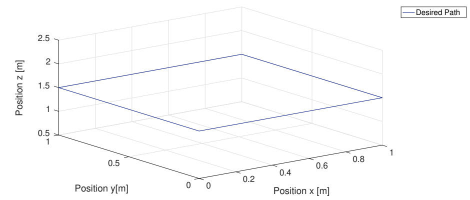

Fig. 8. Desired translation path

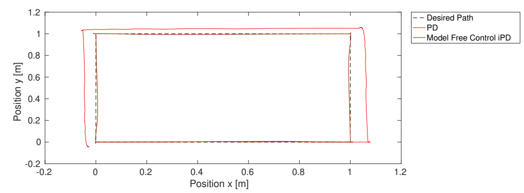

Fig. 9. 2D trajectory response

Figure 10 demonstrate that the MFC produce a more stable desired roll and pitch angle to the attitude controller.

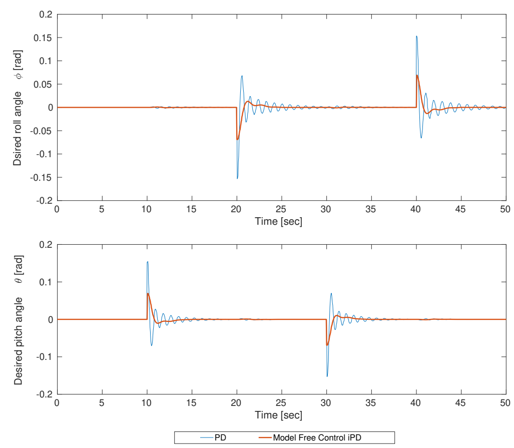

Fig. 10. Attitude desired

Figure 11 represent the estimation error e_{est}, where the curves illustrate the efficiency of online estimation of the ultra-local model F, and it is obvious that the estimation error converges toward a small (ϵ). Consequently, the system is practically stable.

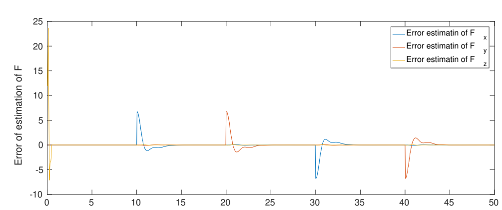

Fig. 11. Estimation error of F e_{est}

Moreover, we present a more thorough comparison in Figure 12, which indicate clearly that the proposed MFC control can ensure the best track of the desired trajectory, with small error compared with classical PD.

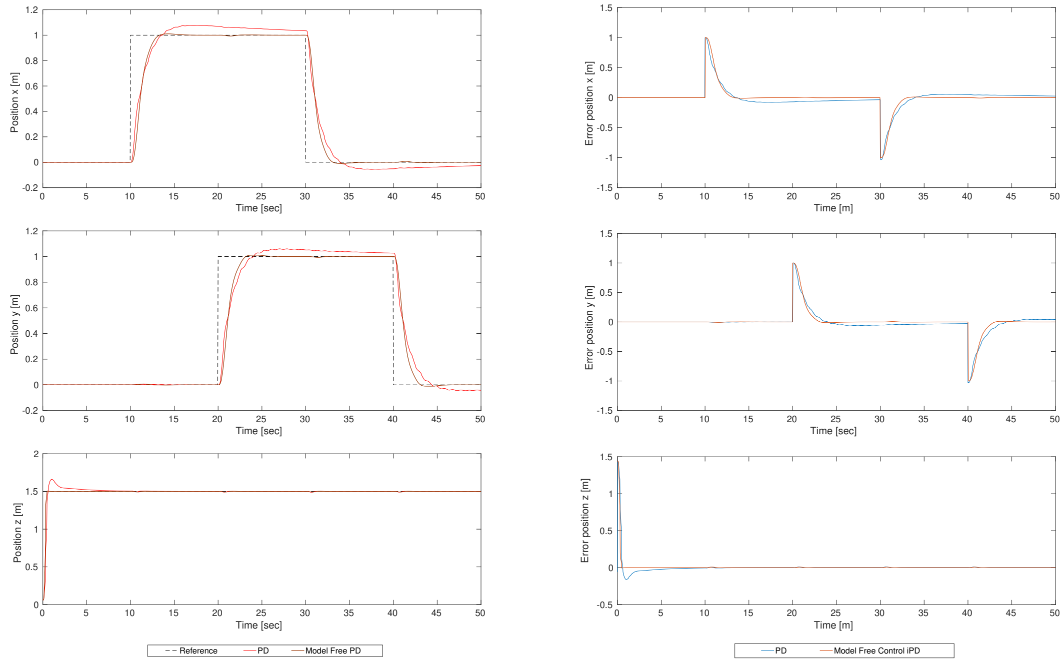

Fig. 12. Deep comparison for trajectory tracking response

V. CONCLUSION

In this paper, an intelligent PD (iPD) controller (MFC) is presented. In the first step, we use a simple PD to control the position of Quadrotor, where the PD gains were adjusted manually. As a second step, the Model-Free Control was introduced. Different simulations were performed using the robotic operating system ROS environment under the framework RotorS Gazebo MAV Simulator, in presence of an external disturbance and extra payload in order to test and compare these two control techniques. The performances of Model-Free controller were better than the classical PD controller as you can see in the result section. In our future studies, we will implement and test our proposed strategy on a real Quadrotor. We aim also to develop other control methods and by comparing them, we will select the best one in terms of performance and power consumption, robustness, among other aspects.

REFERENCES

[1] K. A. Demir, H. Cicibas, and N. Arica, "Unmanned Aerial Vehicle Domain : Areas of Research," vol. 65, no. 4, pp. 319–329, 2015.

[2] R. Amin, L. Aijun, and S. Shamshirband, "A review of quadrotor UAV: Control methodologies and performance evaluation," *International Journal of Automation and Control*, vol. 10, no. 2, pp. 87–103, 2016.

[3] S. Bouabdallah, A. Noth, and R. Siegwart, "PID vs LQ control techniques applied to an indoor micro quadrotor," *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, pp. 2451–2456, 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1389776/

[4] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," *2013 International Conference on Informatics, Electronics and Vision, ICIEV 2013*, 2013.

[5] F. Niel, Y. Ameho, and J.-M. Biannic, "A Novel Parameter Varying Controller Synthesis Method for Quadrotor Control," *AIAA Guidance, Navigation and Control Conference*, 2013.

[6] a. Benallegue, A. Mokhtari, and L. Fridman, "Feedback linearization and high order sliding mode observer for a quadrotor UAV," *International Workshop on Variable Structure Systems, (VSS'06)*, pp. 365–372, 2006.

[7] A. Mokhtaril, A. Benallegue, and B. Daachi, "Robust feedback linearization and GH controller for a quadrotor unmanned aerial vehicle," *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, vol. 57, no. 1, pp. 1009–1014, 2005.

[8] S. Bouabdallah and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro Quadrotor," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. April, pp. 2247–2252, 2005.

[9] C. Coza and C. J. B. Macnab, "A New Robust Adaptive-Fuzzy Control Method Applied to Quadrotor Helicopter Stabilization guide X4a2Q)/ bi rF (x) Tc1 Ei (X)," pp. 475–479.

[10] M. Fliess and C. Join, "Model-free control," *International Journal of Control*, vol. 86, no. 12, pp. 2228–2252, 2013.

[11] Y. A. Younes, A. Drak, H. Noura, A. Rabhi, and A. E. Hajjaji, "Robust Model-Free Control Applied to a Quadrotor UAV," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 84, no. 1-4, pp. 37–52, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10846-016-0351-2

[12] H. Wang, X. Ye, Y. Tian, G. Zheng, and N. Christov, "Model-free-based terminal SMC of quadrotor attitude and position," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2519–2528, 2016.

[13] M. Bekcheva, C. Join, and H. Mounier, "Cascaded Model-Free Control for trajectory tracking of quadrotors," *2018 International Conference on Unmanned Aircraft Systems, ICUAS 2018*, pp. 1359–1368, 2018.

[14] L. D. Minh and C. Ha, "Modeling and control of quadrotor MAV using vision-based measurement," *2010 International Forum on Strategic Technology, IFOST 2010*, pp. 70–75, 2010.

[15] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23

[16] M. Kamel, M. Burri, and R. Siegwart, "Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles," *ArXiv e-prints*, Nov. 2016.