World Academy of Science, Engineering and Technology
International Journal of Electrical and Computer Engineering
Vol:18, No:10, 2024

Model Predictive Control and Proportional-Integral-Derivative Control of Quadcopters: A Comparative Analysis

Anel Hasić, Naser Prljača

***Abstract*—**In the domain of autonomous or piloted flights, the accurate control of quadrotor trajectories is of paramount significance for large numbers of tasks. These adaptable aerial platforms find applications that span from high-precision aerial photography and surveillance to demanding search and rescue missions. Among the fundamental challenges confronting quadrotor operation is the demand for accurate following of desired flight paths. To address this control challenge, among others, two celebrated well-established control strategies have emerged as noteworthy contenders: Model Predictive Control (MPC) and Proportional-Integral-Derivative (PID) control. In this work, we focus on the extensive examination of MPC and PID control techniques by using comprehensive simulation studies in MATLAB/Simulink. Intensive simulation results demonstrate the performance of the studied control algorithms.

***Keywords*—**MATLAB, MPC, Model Predictive Control, PID, Proportional-Integral-Derivative, quadcopter, Simulink.

I. Introduction

A quadcopter, also known as a quadrotor, is a type of helicopter that features four rotors. These rotors are oriented with their blades pointing upward and positioned in a square formation, ensuring they are equidistant from the center of mass of the quadcopter. The quadcopter's movement and flight control [1] are achieved by precisely adjusting the angular speed of these rotors, which are driven by electric motors. This design allows for agile and stable flight by altering the rotational speed of individual rotors, providing control over pitch, roll, yaw, and vertical motion.

Controlling a quadcopter presents a particularly complex and intriguing challenge, as highlighted in [2] and [3]. This complexity arises from the fact that quadcopters possess six degrees of freedom, consisting of three translational and three rotational axes, while being equipped with only four independent inputs or actuators, namely the rotors. This fundamental configuration renders quadcopters highly non-linear, multivariable, and inherently unstable systems.

Furthermore, one of the distinctive challenges faced when working with quadcopters is their minimal friction with the environment, unlike land vehicles. Because of this, they must generate their own damping forces to decelerate, stop, and maintain stability. In combination, these factors create a fascinating and intricate control and management problem specific to quadcopters, as discussed in [4] and [5]. Addressing these challenges requires advanced control strategies and algorithms to ensure safe and stable flight and precise maneuvering.

In order to improve quadrotor (quadcopter) tracking performance while taking environmental factors into account, research studies have introduced controllers such as Linear Quadratic Regulator (LQR), PID, Feedback Linearization Control (FLC), Backstepping, MPC, Sliding Mode Control (SMC), Linear Quadratic Gaussian (LQR), neural network, fuzzy logic, robust, adaptive, etc. These controllers are divided into three categories: linear, nonlinear and learning-based controllers [6]-[9].

MPC is a comprehensive category of automatic control system design methods. It operates by utilizing predictions of the process response in the future, often referred to as the "prediction horizon" [10], to determine the appropriate control signal for the process. This control signal is chosen to drive the system's response in such a way, that it tracks the desired process output. The fundamental concept behind MPC is straightforward and intuitive: formulate a mathematically defined problem to adjust the system's behavior in the future, ensuring it matches the desired or intended behavior. By considering predictions of how the system will evolve, MPC enables the optimization of control inputs to meet specific performance objectives, making it a powerful and versatile control strategy used in various fields, including engineering, robotics, and process control.

The prediction horizon of MPC is not infinite but extends only over a finite time period into the future. This finite prediction horizon introduces numerous theoretical challenges, particularly in the analysis of the stability of the resulting control system. These challenges have continued to be a source of ongoing theoretical research to this day.

Okasha et al. [11] demonstrate that all three controllers (PID, LQR and MPC) demonstrate similar tracking performance in simulations and experiments. Saraf et al. [12] indicate that the PID controller, requiring six feedback loops, is computationally more complex than the single control loop in the case of an LQR controller. The LQR controller is better suited for a quadrotor (quadcopter) control mechanism than the classical PID controller in terms of output, complexity and computation time.

Ma.El. Anel Hasić is with the Department of Automation, Robotics and Industrial informatics, Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina (corresponding author, phone: +38762910542; e-mail: anel.hasic@ fet.ba).

Prof. Dr. Naser Prljača is with the Department of Automation, Robotics and Industrial informatics, Faculty of Electrical Engineering, University of Tuzla, Bosnia and Herzegovina (e-mail: naser.prljaca@ fet.ba).

Khatoon et al. [13] have explored how PID controllers give better stability by pushing the closed-loop poles to the negative side of the s-plane as compared to LQR controllers. Yavuz et al. [14] tackle the quadcopter control's problem by proposing an adaptive controller that is a hybrid of both PID and LQR controllers. Nair et al. [15] present results on a PID controller coupled with a Kalman filter being applied to UAVs. Argentim et al. [16] work on using LQR control to tune a PID controller, combining the control techniques to manage the quadcopter. Prljača and Bjelić [5] present a decentralized robust adaptive controller for UAVs, capable of overcoming propeller faults.

In [17], PD, LQR and MPC controllers have been considered on a quaternion orientation-based quadcopter platform. This study compares the performance of controllers based on trajectory tracking and control efforts through simulations, and infers suitable environments for different controllers. Based on the results, MPC is a suitable controller for quadcopter platforms in outdoor applications due to its disturbance rejection capacity. In contrast, PD and LQR can be considered for indoor applications where disturbance is very negligible.

One practical consideration in implementing MPC is the relatively large computational requirements, especially when performing real-time computations. This limitation can restrict the use of MPC to slower processes or systems that operate with sufficiently large sampling intervals. For instance, MPC was initially applied in fields such as chemical processes, which often exhibit slower dynamics and can accommodate the computational demands of MPC. However, in recent years, efforts have been made to adapt and optimize MPC for faster systems and real-time applications, expanding its scope of use to a wider range of processes and industries. In [23], MPC implementation on microcontroller unit (MCU) is discussed.

This paper presents the development of a quadcopter position control system employing MPC. Following the introduction, the first section includes the formulation of a comprehensive nonlinear model to describe the quadcopter's dynamics within a three-dimensional spatial framework. This nonlinear model is then subject to linearization, centered around an equilibrium point, commonly the hovering state. Subsequently, the continuous-time linear model is discretized to accommodate discrete time intervals.

The heart of this approach lies in the formulation of an iterative optimization problem based on the discretized system model. This problem aims to identify optimal control signals within a predefined prediction horizon. The objective is to ensure that the quadcopter's future behavior closely adheres to desired reference trajectories and overall objectives.

A pivotal aspect of this study is the comparative analysis conducted to assess the effectiveness of the newly developed closed-loop position control system. This analysis entails a thorough evaluation of the system's performance, stability, and robustness. It is conducted in direct comparison with a control system implementing a PID controller.

Multiple distinct scenarios are scrutinized in depth to comprehensively appraise the respective merits and limitations of each control method.

In summary, this paper offers a comprehensive and rigorously technical framework for quadcopter position control. It emphasizes the unique advantages presented by MPC, particularly in situations characterized by abrupt changes in reference positions. The methodical approach exhibited in this study underscores the practicality and efficacy of implementing MPC to effectively address the intricate control challenges intrinsic to quadcopter systems.

II. Mathematical Model of Quadcopter

A. Non-Linear Model

By taking into account the position and translational velocity of the quadcopter, along with its orientation and angular velocity, it is possible to derive a nonlinear model in the state space [18]:

$$\dot{\xi}=\mathbf{f}(\xi, u) \tag{1}$$

where the vector state ξ is defined as: $\xi = [\dot{x}, \dot{y}, \dot{z}, q_0, q_1, q_2, q_3, \omega_x, \omega_y, \omega_z, x, y, z]^T$ and the input vector u is a non-linear function of the rotor angular velocities $u = [u_1, u_2, u_3, u_4]^T$, where $u = h(\omega)$.

Let x, y, and z represent the center-of-mass position in an inertial coordinate system. The time derivatives of x, y, and z, denoted as $\dot{x}, \dot{y}, \dot{z}$ respectively, represent the quadcopter's velocity. Q_0, q_1, q_2, and q_3 denote the quaternions representing the quadcopter's orientation. The angular velocity elements of the quadcopter, expressed in the body frame, are represented by $\boldsymbol{\omega}_x$, $\boldsymbol{\omega}_y$, and $\boldsymbol{\omega}_z$. The mass of the quadcopter is denoted by m, and its moments of inertia about the body axes are represented by J_x, J_y, and J_z.

Quadcopter rotor angular speeds are denoted as $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, $\boldsymbol{\omega}_3$, and $\boldsymbol{\omega}_4$. It is worth noting that the quadcopter configuration is as shown in Fig. 1, with adjacent rotors rotating in opposite directions.

To simplify the analysis, we define the following set of artificial control input variables:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} \begin{bmatrix} b & b & b & b \\ 0 & b & 0 & -b \\ b & 0 & -b & 0 \\ d & -d & d & -d \end{bmatrix} \tag{2}$$

where b represents the aerodynamic lift or thrust factor, and d represents the aerodynamic moment of drag factor applied to the rotors.

These factors are crucial in understanding and modeling the quadcopter's aerodynamic behavior and control. In (2), u_1 is the sum of all rotor trust forces and thus matches the resulting lift. The variables u_2 and u_3 correspond to those forces resulting from a speed difference of two opposite rotating motors, leading to roll and pitch movements. The last virtual input u_4 can be interpreted as the yaw moment.

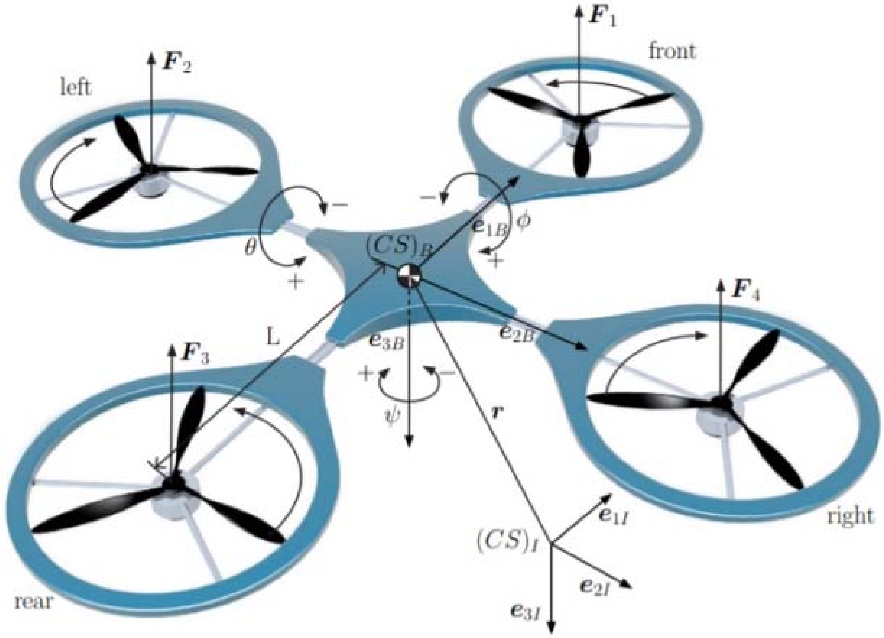

Fig. 1 Mapping nonlinear data to a higher dimensional feature space

According to the formulations presented in [18] and [19], the translation acceleration is obtainable as:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\begin{bmatrix}-2(q_1q_3+q_0q_2)\frac{u_1}{m}\\ -2(q_2q_3+q_0q_1)\frac{u_1}{m}\\ \omega_1-\omega_2+\omega_3-\omega_4-(q_0^2-q_1^2-q_2^2+q_3^2)\frac{u_1}{m}\end{bmatrix} \tag{3}$$

Angular acceleration $\ddot{\omega}_i$ can be stated as:

$$\begin{bmatrix}\ddot{\omega}_x\\ \ddot{\omega}_y\\ \ddot{\omega}_z\end{bmatrix}=\begin{bmatrix}\omega_y\omega_z\frac{J_y-J_z}{J_x}-\frac{J_r}{J_x}\omega_y(\omega_1-\omega_2+\omega_3-\omega_4)+\frac{L}{J_x}u_2\\ \omega_x\omega_z\frac{J_z-J_x}{J_y}-\frac{J_r}{J_y}\omega_x(\omega_1-\omega_2+\omega_3-\omega_4)+\frac{L}{J_y}u_3\\ \omega_x\omega_y\frac{J_x-J_y}{J_z}+\frac{1}{J_z}u_4\end{bmatrix} \tag{4}$$

where J_r is the inertia factor of the rotors (which are assumed as equal between all rotors), and L stands for the lengths of the lever between the quadcopter's center-of-mass and the four motors (which are also assumed as equal).

The attitude kinematics can be expressed via quaternion derivates as follows:

$$\begin{bmatrix}\dot{q}_0\\ \dot{q}_1\\ \dot{q}_2\\ \dot{q}_3\end{bmatrix}=\frac{1}{2}\begin{bmatrix}q_0 & -q_1 & -q_2 & -q_3\\ q_1 & q_0 & -q_3 & q_2\\ q_2 & q_3 & q_0 & -q_1\\ q_3 & -q_2 & q_1 & q_0\end{bmatrix}\begin{bmatrix}0\\ \omega_x\\ \omega_y\\ \omega_z\end{bmatrix} \tag{5}$$

Therefore, we define the total 13-dimensional state vector of the six degrees-of freedom quadcopter dynamics as:

$$\xi=\left[\dot{x},\dot{y},\dot{z},q_0,q_1,q_2,q_3,\omega_x,\omega_y,\omega_z,x,y,z\right]^T \tag{6}$$

The following system summarizes (3)-(5) merely in terms of ξ.

$$\dot{\boldsymbol{\xi}}=\begin{bmatrix}-2(\xi_5\xi_7+\xi_4\xi_6)\frac{u_1}{m}\\ -2(\xi_6\xi_7-\xi_4\xi_5)\frac{u_1}{m}\\ g(u)-(\xi_4^2-\xi_5^2-\xi_6^2+\xi_7^2)\frac{u_1}{m}\\ \frac{1}{2}(-\xi_5\xi_8-\xi_6\xi_9-\xi_7\xi_{10})\\ \frac{1}{2}(\xi_4\xi_8-\xi_7\xi_9+\xi_6\xi_{10})\\ \frac{1}{2}(\xi_7\xi_8+\xi_4\xi_9-\xi_5\xi_{10})\\ \frac{1}{2}(-\xi_6\xi_8+\xi_5\xi_9+\xi_4\xi_{10})\\ \xi_9\xi_{10}\frac{J_y-J_z}{J_x}-\frac{J_r}{J_x}\xi_9 g(u)+\frac{L}{J_x}u_2\\ \xi_8\xi_{10}\frac{J_z-J_x}{J_y}-\frac{J_r}{J_y}\xi_8 g(u)+\frac{L}{J_y}u_3\\ \xi_8\xi_9\frac{J_x-J_y}{J_z}+\frac{1}{J_z}u_4\\ \xi_1\\ \xi_2\\ \xi_3\end{bmatrix} \tag{7}$$

where $g(u)$ represents a linear combination of the angular velocities of a quadcopter's rotors and it is defined as:

$$g(u)=\omega_1-\omega_2+\omega_3-\omega_4 \tag{8}$$

B. Linearized Model

For the development of the control system, the nonlinear model can be linearized by approximating selected operating points.

General, for a non-linear system $\dot{\xi}=f(\xi,u)$ and the selected operating point (ξr,ur) the linearized system can be written in a linear time-invariant form:

$$\Delta\dot{\xi}=A\Delta\xi+B\Delta u \tag{9}$$

where $\Delta\xi = \xi - \xi_r$ and $\Delta u = u - u_r$, and matrices A and B are calculated as:

$$A = \left.\frac{\partial f}{\partial \xi}\right|_{(\xi_r, u_r)} \quad B = \left.\frac{\partial f}{\partial u}\right|_{(\xi_r, u_r)} \tag{10}$$

In the case of a quadcopter, a common point for linearization is during hovering, where the condition $u_1 = mg$ holds. This condition allows us to derive an expression for the rotor speeds as follows:

$$\omega_i = \sqrt{mg/(4b)} \tag{11}$$

III. Software Solution

A. MPC Controller for Trajectory Tracking Control of Quadcopter

The initial step in implementing MPC is to compile the quadcopter parameters, as outlined in Table I. This step involves gathering and organizing the necessary data to develop and set up the MPC control system for the quadcopter.

Following the compilation of quadcopter parameters as outlined in Table I, the next crucial step is the initialization of Euler angles and their conversion into quaternions using the MATLAB function 'euler2quaternion'. It is essential to establish the hovering state at 0 meters above the ground (with the z-axis pointing down) and then transform the hover input accordingly. Subsequently, one can proceed to linearize the system around the hovering state to obtain the state-space matrices. Once these initial settings and linearization are completed, the controller can be simulated considering the specified parameters and constraints [20].

TABLE I
INITIAL VALUES OF QUADCOPTER PARAMETERS

Parameter	Variable	Value	Unit
Mass	QUAD.MASS	0.58	kg
Lever length	QUAD.LEVER_LENGTH	0.25	m
Roll inertia	QUAD.ROLL_INTERIA	0.01	kgm^2
Pitch inertia	QUAD.PITCH_INTERIA	0.01	kgm^2
Yaw inertia	QUAD.YAW_INTERIA	0.02	kgm^2
Rotor inertia	QUAD.ROTOR_INTERIA	3.8e-5	kgm^2
Drag factor	QUAD.DRAG_FACTOR	2.82e-7	kgm^2
Thrust factor	QUAD.THRUST_FACTOR	1.55e-5	kgm^2
Gravity	QUAD.GRAVITY	9.81	ms^2
Mass	QUAD.MASS	0.58	kg
Roll inertia	QUAD.ROLL_INTERIA	0.01	kgm^2
Lever length	QUAD.LEVER_LENGTH	0.25	m

This MPC controller will play a pivotal role in the quadcopter's control and stabilization during various flight scenarios. The Simulink model for the MPC controller is shown in Fig. 2.

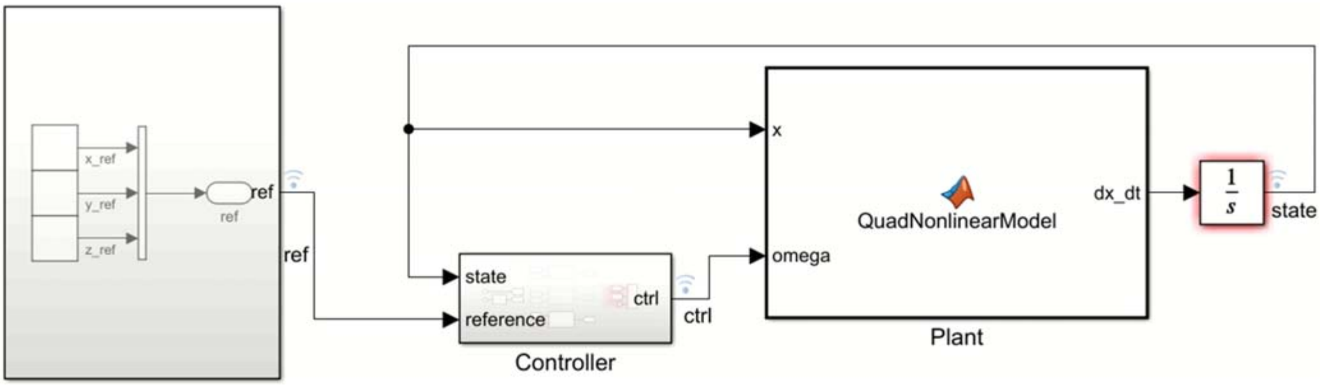

Fig. 2 Simulink MPC quadcopter model for real system

The input to the system is the MATLAB/Simulink ‘ref’ block, which consists of quadcopter positions (x,y,z), as shown in Fig. 3.

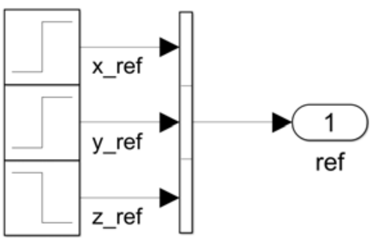

Fig. 3 System input

The optimization problem was formulated using YALMIP [21], and the MATLAB function ´quadprog´ [22] was employed to solve it. This combination of tools allowed efficient formulation and solution of the optimization problem, which is a fundamental aspect of implementing the MPC controller for the quadcopter.

The parameters for the regulator are provided in Table II. These parameters play a significant role in determining the behavior and performance of the quadcopter's control system.

TABLE II
MPC PARAMETERS

Parameter	Value	Unit	Description
N	10	-	Prediction horizon
Q_x, Q_y, Q_z	1	$1/m^2$	Position tracking error weight
R_1, R_2, R_3	0.1	$1/N^2$	The difficulty of using control forces
R_4	0.1	$1/(Nm)^2$	The difficulty of using the steering torque
T_S	1	s	Sampling time

TABLE III
MPC CONSTRAINTS

u_{min}	Variable	u_{max}
1N	$\leq u_1 \leq$	15N
-2N	$\leq u_2, u_3 \leq$	2N
-0.05 Nm	$\leq u_4 \leq$	0.05 Nm

The matrices Q and R are diagonal, with the elements specified for each. This structure is commonly used in control design to weigh the importance of state variables and control inputs. Additionally, it is important to note that control inputs are constrained within the values presented in Table III. These constraints are essential for ensuring safe and effective control of the quadcopter while taking into account practical limitations.

B. PID Controller for Trajectory Tracking Control of Quadcopter

For the purpose of comparing the control system with the PID controller, the simulation model as described in [5] was employed. For the PID controller, the same parameters as those used for the MPC controller (following the same initial procedure) were utilized. The Simulink model for the PID controller is illustrated in Fig. 4. This setup enables a direct comparison between the MPC and PID control strategies for the quadcopter.

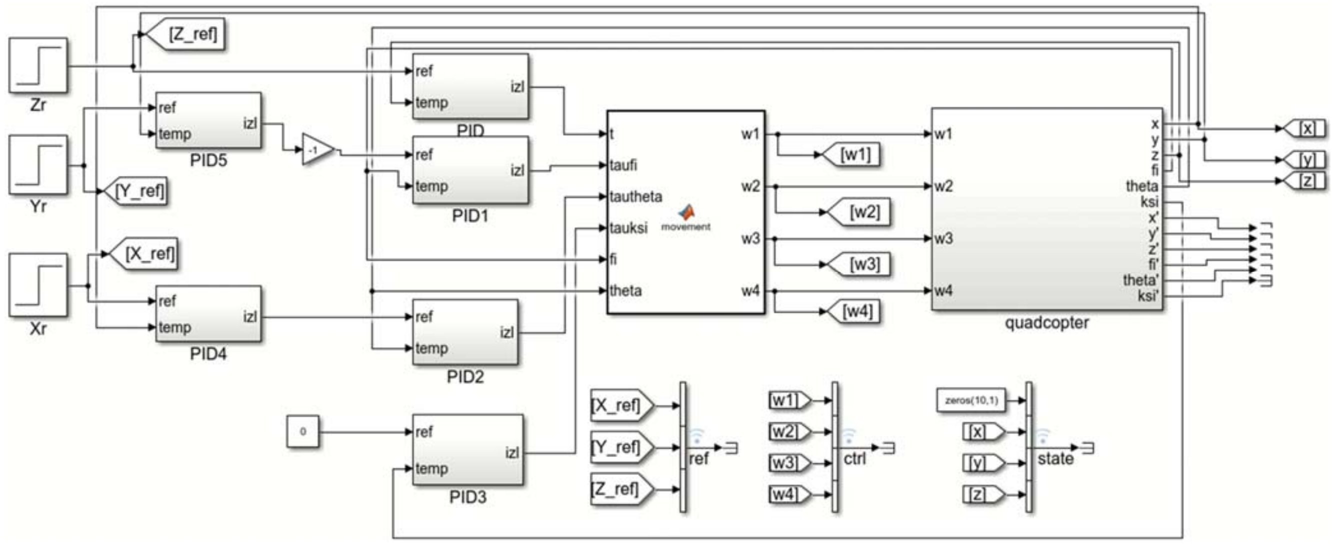

Fig. 4 Simulink PID quadcopter model for real system

IV. RESULTS AND DISCUSSION

The control of the quadcopter is governed by its position (x, y, z) and constraints on the attainable control signals. The study of both MPC and PID controllers involved two distinct cases:

- Step Response: This case focuses on analyzing the behavior and performance of the controllers when subjected to a step input, typically used to evaluate the system's response to sudden changes in references or disturbances.
- Response with Uncertain System Parameters: In this case, the controllers are tested under conditions where the model parameters may not accurately represent the physical system. This scenario assesses the controllers' robustness and their ability to handle deviations or uncertainties in the model.

These cases provide a comprehensive evaluation of both the MPC and PID controllers in various operating conditions, offering insights into their performance, stability, and robustness.

A. Step Response

Figs. 5 and 6 display the results achieved using the MPC controller.

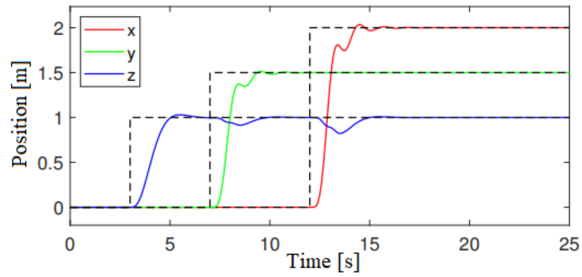

Fig. 5 Step response to change of reference - MPC

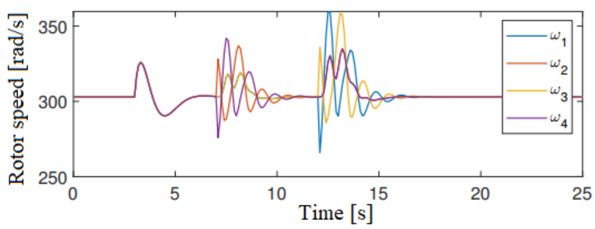

Fig. 6 Step response to change of reference – MPC

In Figs. 5 and 6, it is evident that the reference tracking is highly satisfactory, with minimal oscillations. Furthermore, it is noteworthy that the rotor speed does not exhibit significant increases during the simulation, with the maximum change being approximately 20%. This performance demonstrates the effectiveness of the MPC controller in maintaining stable and accurate control over the quadcopter.

Figs. 7 and 8 illustrate the results obtained with the PID controllers, which were used for comparison.

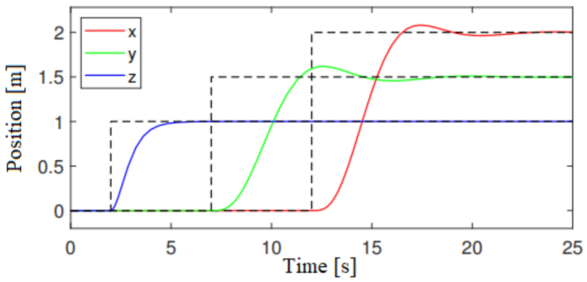

Fig. 7 Step response to change of reference – PID

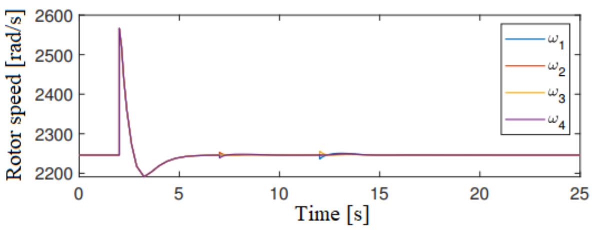

Fig. 8 Control signals with a step reference change – PID

In this scenario, the response is observed to be slower but more stable, characterized by a smoother trajectory with less oscillation or overshoot. The response also appears to exhibit a bit more overhang, indicating a controlled and gradual approach to the reference setpoints. These characteristics demonstrate the differences in performance and behavior between the MPC and PID controllers, offering insights into the trade-offs between speed and stability in control strategies.

Analyzing the results presented in Figs. 5-8, it is apparent that the tracking and monitoring performance for all movement scenarios discussed in the paper is excellent. The controllers exhibit good response times, accurately following reference trajectories, and maintaining system stability.

As mentioned before, the next section will delve into a deeper level of analysis by introducing changes to the system parameters. This will provide valuable insights into the controllers' robustness and their ability to adapt to variations or uncertainties in the model, which is a crucial aspect of real-world control system design and deployment.

B. Response with Wrong Model Parameters

In this chapter we will explore the impact of changing system parameters, specifically by increasing the mass of the quadcopter by 30% and decreasing the thrust factor by 20%.

As we can see in Figs. 9 and 10, as expected, when using MPC with parameters that do not accurately represent the physical system, it is evident that the tracking of the vertical position reference is weaker. This is primarily because the model's deviation from the actual system parameters results in a greater influence of gravity, affecting the quadcopter's vertical position control.

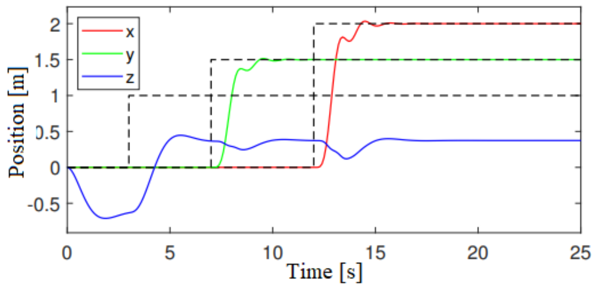

Fig. 9 Response with wrong parameters – MPC

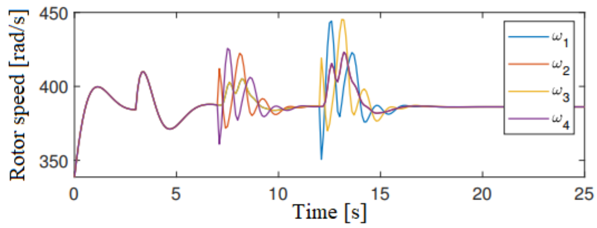

Fig. 10 Control signals with wrong parameters – MPC

These findings highlight the importance of model accuracy and the trade-offs between control strategies. While MPC is a powerful control method when the model matches the system, it can be more sensitive to parameter deviations. This emphasizes the need for robust control strategies in real-world applications where model uncertainty or parameter variations are common.

Figs. 11 and 12 display the responses obtained with PID controllers when subjected to changes in system parameters.

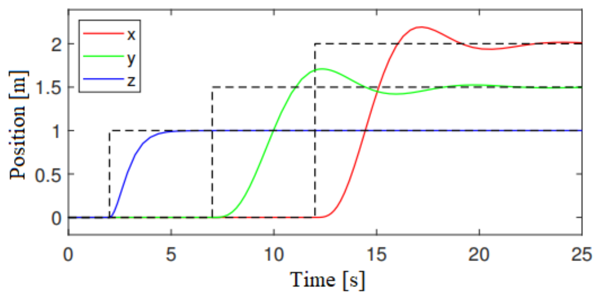

Fig. 11 Response with wrong parameters – PID

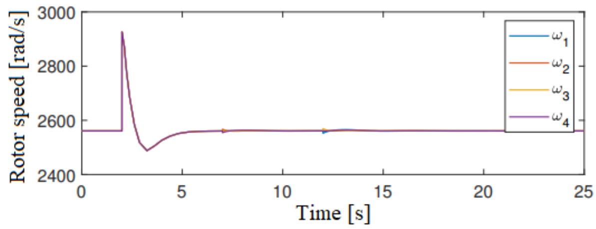

Fig. 12 Control signals with wrong parameters – PID

Notably, the responses with the PID controllers did not change significantly, which indicates that the tracking remains proper even in the presence of altered parameters. This suggests that PID controllers may exhibit greater robustness to parameter variations compared to the MPC controller, maintaining consistent performance under these conditions.

These results emphasize the resilience of PID controllers in scenarios with uncertain or varying parameters, making them a valuable choice in applications where system parameters may not be precisely known.

V. Conclusion

In this paper, we present systematic design of MPC for position control of quadcopters. Through extensive simulations, the closed-loop performance of the control system was validated. When compared to a control system using a PID controller, it is observed that MPC yields superior results, particularly when faced with sudden changes in the reference signals.

However, it is worth noting that the performance of MPC degrades when inaccurate or wrong parameters (model) are used. An advantage of MPC over PID control lies in easier parameter tuning. Additionally, the imposition of constraints on control inputs and outputs ensures the prevention of undesirable quadcopter behavior and ensures its safety.

As a future work, the study could progress towards non-linear or adaptive MPC to mitigate the influence of model errors and improve performance. Moreover, conducting real-world tests on an actual quadcopter would demonstrate the practical applicability and effectiveness of the proposed algorithm. This research paves the way for more robust and adaptable control strategies in the field of quadcopter control and aerial robotics.

References

[1] Y. Fan, "Flight Control System Simulation for Quadcopter Unmanned Aerial Vehicle (UAV) based on Matlab Simulink," J. Phys.: Conf. Ser., vol. 2283, p. 012011, 2022.
[2] G. Caprari, A. Breitenmoser, and A. Fischer, "Highly compact robots for inspection of power plants," J. Field Robot., vol. 29, no. 1, pp. 47–68, 2012.
[3] Committee on Autonomous Vehicles in Support of Naval Operations, National Research Council, "Autonomous vehicles in support of naval operations," Naval Studies Board, Washington DC, USA, 2005.
[4] K. Alexis, G. Nikolakopoulos, and G. Tzes, "Autonomous quadcopter position and attitude PID/PIDD controls in GPS-denied environments," Int. Rev. Autom. Control, vol. 3, pp. 421-430, 2011.
[5] N. Prljača and A. Bjelić, "Robust Decentralized Adaptive Control of a Quadrotor," in Proc. 23rd Mediterranean Conf. Control and Automation, Spain, 2015.
[6] A. Zulu and S. John, "A Review of Control Algorithms for Autonomous Quadrotors," arXiv:1602.02622, 2016.
[7] J. C. V. Junior, J. C. Paula, J. C. De Paula, G. V. Leandro, and M. C. Bonfim, "Stability Control of a Quad-rotor Using a PID Controller," Braz. J. Instrum. Control, vol. 1, pp. 15–20, 2013.
[8] H. ElKholy, "Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches," Master's thesis, The American University in Cairo, New Cairo, Egypt, 2014.
[9] F. Kendoul, "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems," J. Field Robot., vol. 29, pp. 315–378, 2012.
[10] G. Ganga and M. M. Dharmana, "MPC Controller for Trajectory Tracking Control of Quadcopter," 2017.
[11] M. Okasha, J. Kralev, and M. Islam, "Design and Experimental Comparison of PID, LQR and Stabilizing Controllers for Parrot Mambo Mini-D," 2022.
[12] P. Saraf, M. Gupta, and A. M. Parimi, "A Comparative Study Between a Classical and Optimal Controller for a Quadcopter," 2020.
[13] S. Khatoon, D. Gupta, and L. K. Das, "PID & LQR control for a quadrotor: Modeling and simulation," in Proc. 2014 Int. Conf. Advances in Computing, Communications and Informatics (ICACCI), New Delhi, 2014, pp. 796-802, doi: 10.1109/ICACCI.2014.6968232.
[14] H. Yavuz and S. İkizoğlu, "Hyperbolic Tangent Adaptive LQR+PID Control of a Quadrotor," in Proc. 6th Int. Conf. Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 2018, pp. 1-6, doi: 10.1109/CEIT.2018.8751878.
[15] M. P. Nair and R. Harikumar, "Longitudinal dynamics control of UAV," in Proc. Int. Conf. Control Communication & Computing India (ICCC), Trivandrum, 2015, pp. 30-35, doi: 10.1109/ICCC.2015.7432865.
[16] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, "PID, LQR and LQR-PID on a quadcopter platform," in Proc. Int. Conf. Informatics, Electronics and Vision (ICIEV), Dhaka, 2013, pp. 1-6, doi: 10.1109/ICIEV.2013.6572698.
[17] M. Islam and M. Okasha, "A Comparative Study of PD, LQR and MPC on Quadrotor Using Quaternion Approach," 2019.
[18] M. Schreier, "Modeling and adaptive control of a quadcopter," in Proc. IEEE Int. Conf. Mechatronics and Automation, pp. 383-390, 2012.
[19] M. N. Shauqee, P. Rajendran, and N. M. Suhadis, "An effective proportional-double derivative-linear quadratic regulator for quadcopter attitude and altitude control," 2021.
[20] M. D. Shuster, "A survey of attitude representations," Navigation, vol. 8, no. 9, pp. 439–517, 1993.
[21] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in Proc. CACSD Conf., Taipei, Taiwan, 2004.
[22] MATLAB, "Quadprog package – functions to solve quadratic programming problems."
[23] N. Hrustić and N. Prljača, "An implementation and evaluation of fast embedded model predictive control," in Proc. 16th Int. Symp. INFOTEH-Jahorina, 2020.