Non-linear Model-Based Predictive Control for Trajectory Tracking and Control Effort Minimization in a Smartphone-Based Quadrotor

Submitted: 15th June 2021; accepted: 6th September 2022

Luis García, Esteban Rosero

DOI: 10.14313/JAMRIS/4-2022/28

Abstract:
In this paper, the design and implementation of a non-linear model-based predictive controller (NMPC) for predefined trajectory tracking and to minimize the control effort of a smartphone-based quadrotor are developed. The optimal control actions are calculated in each iteration by means of an optimal control algorithm based on the non-linear model of the quadrotor, considering some aerodynamic effects. Control algorithm implementation and simulation tests are executed on a smartphone using the CasADi framework. In addition, a technique for estimating the energy consumed based on control signals is presented. NMPC controller performance was compared with other works developed towards the control of quadrotors, based on an H_∞ controller and an LQI controller, and using three predefined trajectories, where the NMPC average tracking error was around 50% lower, and average estimated power and energy consumption slightly higher, with respect to the H_∞ and LQI controllers.

Keywords: *Quadrotor, Model-based Predictive Control, Smartphone, Trajectory Tracking, Energy Consumption.*

1. Introduction

Interest in unmanned aerial vehicles (UAV) control has grown significantly in recent years in research areas focused on both military and civil applications (e.g., intelligence, reconnaissance, surveillance, exploration of dangerous environments). One type of UAV that has caught the attention of the scientific community is the quadrotor which, because of its size, mechanical simplicity, low cost, maneuverability, flight autonomy, and wide range of applications, is increasingly deployed and able to replace humans in difficult and risky tasks.

To develop a quadrotor's control system, it is necessary to measure the position and orientation of the vehicle, which requires a flight control card and various sensors. These features can be found in systems embedded in devices such as smartphones, whose use has grown rapidly in recent decades. Hardware features such as processing power, memory, sensors, and communication technologies, as well as its programmable architecture, combine to provide a lot of potential benefits in mobile application development, so that the smartphone can execute complex computational tasks and make use of its peripherals [1].

Due to their features, quadrotors are increasingly used in smartphone-based robots and unmanned vehicles as part of closed-loop control systems. In [2], a mobile application is developed to measure and control the angular position of a test-bed, where a smartphone is located, feeding back its position using its embedded sensors and calculating the control signal using a PD controller. In [3], an autonomous robot controlled via an Android-based smartphone is developed using odometry derived from the robot's wheels and image recognition using the smartphone's camera to obtain a stereo image to describe three-dimensional objects. This allows the robot to reach a certain position in a room and avoid obstacles placed in its way. Taking advantage of the inherent benefits of using an embedded system such as a smartphone, in the work developed in [4] the jOptimizer framework (based on java) and the CasADi framework (based on C) were tested to solve an MPC control optimization problem for the control of a smartphone-based quadrotor for trajectory tracking. It can be seen that CasADi enables the solving of non-linear programming (NLP) problems, achieving shorter processing times and less tracking error.

Aerodynamic effects have significant impact on energy optimization and trajectory tracking, helping researchers to get closer to real-life quadrotor performance, so that more appropriate control signals can be calculated. In [5], the quadrotor's energy consumption is studied in three types of trajectories (minimum acceleration, jerk, and snap), contributing to the trajectory planning based on aerodynamic effects to improve energy efficiency in vehicles using an optimization algorithm. In [6], position error reduction by incorporating aerodynamic drag into the dynamic model of a quadrotor and performing a drag and thrust compensation is achieved in single direction displacement tests and control actions computing via an ODROID-C2 card. In [7], the mathematical model of a DC motor (including aerodynamic drag losses) and the dynamic model of a quadrotor are considered, and two problems are formulated to optimize trajectories with minimum energy consumption, subject to system constraints, with minimal computing time and fixed energy consumption, to identify an energy efficiency function that quantifies the energy saved in a mission.

Developments have been made around optimal control for trajectory tracking and energy consumption optimization in quadrotors. One of the most widely used optimal controllers is the Model-based Predictive Control (MPC), which aims to find the optimal control signals by minimizing a cost function, subject to the system's dynamics, inputs, and state variables, among other constraints. For this purpose, an optimal control problem (OCP) is solved at each iteration along a prediction horizon. In [8], a linear MPC controller is used to move a quadrotor in one direction (round trip) using information provided by a disturbance estimation model to suppress its effects. In [9], a particle filter MPC (PF-MPC) control is presented, which has advantages of conventional MPC and adds measured noise and unmeasured disturbances effects to follow a 2D trajectory and minimize disturbances in a quadrotor. In [10], a method to generate trajectories through a 3D terrain for a quadrotor flight using an MPC with acceleration, position, and jerk linear constraints is presented, as well as the terrain map cost, solving a convex optimal control problem using the CVX package. In [11], a hierarchical MPC control is applied to a fleet of quadrotors, which consists of a linear and time-varying MPC (LTV-MPC) at the top level to generate trajectories and avoid obstacles, and a linear and invariant in time MPC (LTI-MPC) at the lower level to stabilize each quadrotor.

In reviewed studies, no research and development around MPC controller implementation using a quaternion-based non-linear quadrotor model were found. In addition, no online non-linear MPC (NMPC) has been implemented using a smartphone to reduce trajectory tracking error or energy consumption reduction. For these reasons, this paper presents a novel design and implementation of NMPC controllers based on the dynamic model of a quadrotor, considering aerodynamic effects, for predefined trajectories tracking and energy consumption monimization. The optimal control problem is solved using the CasADi framework for the control algorithm.

This paper is structured as follows: a dynamic model of the quadrotor in X configuration, including the aerodynamic effects and the estimation of energy consumption, is presented in Section 2; an MPC optimal control problem is defined in Section 3; in Section 4, the implementation of the designed MPC controllers and trajectory tracking tests are performed; and conclusions are presented in Section 5.

2. Quadrotor Model

A quaternion-based quadrotor dynamic model is used [12–15], which has advantages for eliminating effects such as gimbal-lock and the discontinuities of the Euler angles-based model [13]. This model includes the aerodynamic effects due to translational drag, rotational drag, and gyroscopic torque. The system dynamics can be expressed in relation to an inertial reference frame to measure the quadrotor position, and to a body-fixed frame to measure the quadrotor rotation, as shown in Figure 1.

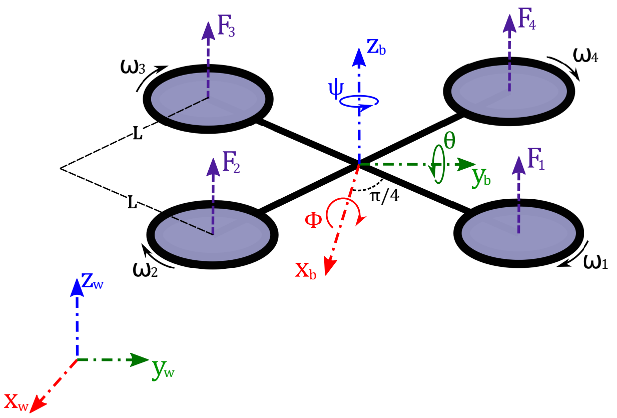

Fig. 1. Quadrotor scheme in "X" configuration.

The quadrotor model is based on the state vector which is defined as:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{\xi}^T & \dot{\boldsymbol{\xi}}^T & \boldsymbol{q}^T & \boldsymbol{\eta}^T \end{bmatrix}^T, \tag{1}$$

where $\boldsymbol{\xi} = \begin{bmatrix} x & y & z \end{bmatrix}^T$ and $\dot{\boldsymbol{\xi}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T$ indicate the position and velocity, respectively, in the inertial reference frame, $\boldsymbol{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T$ is the orientation quaternion, and $\boldsymbol{\eta} = \begin{bmatrix} \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ represents the angular velocity of the quadrotor, both in the body-fixed frame. Furthermore, the input vector $\boldsymbol{U} = \begin{bmatrix} F_th & \boldsymbol{\tau} \end{bmatrix}^T = \begin{bmatrix} F_th & \tau_x & \tau_y & \tau_z \end{bmatrix}^T$ is defined as:

$$\boldsymbol{U} = \begin{bmatrix} F_{th} \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ -k_M & k_M & -k_M & k_M \\ -L_x & -L_x & L_x & L_x \\ -L_x & L_x & L_x & -L_x \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix}, \tag{2}$$

where F_{th}, τ_x, τ_y, and τ_z indicate the thrust on the z axis and the torques on the x, y, and z axis in the body-fixed frame, and $F_i = k_{Ti}{}^2 (i = 1, 2, 3, 4)$ is the thrust applied by the *i-th* motor. The quaternion-based quadrotor model will be defined as:

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{k}) = \begin{bmatrix} \dot{\boldsymbol{\xi}} \\ \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{F_{th}}/m \end{bmatrix} \otimes \boldsymbol{q}^* + \boldsymbol{g} - \frac{1}{m} \boldsymbol{D_v} \\ \frac{1}{2} \boldsymbol{q} \otimes \begin{bmatrix} 0 \\ \boldsymbol{\eta} \end{bmatrix} \\ \boldsymbol{J_q}^{-1} (\boldsymbol{\tau} - \boldsymbol{\eta} \times \boldsymbol{J_q} \boldsymbol{\eta} - \boldsymbol{\tau_{gyro}} - \boldsymbol{D_\eta}) \end{bmatrix}, \tag{3}$$

where m is the quadrotor mass, g is the gravity vector, and $\boldsymbol{J_q} = diag\,(J_{xx}, J_{yy}, J_{zz})$ is the vehicle inertia matrix. $\boldsymbol{D_v}$, $\boldsymbol{D_\eta}$, and $\boldsymbol{\tau_{gyro}}$ correspond to the aerodynamic losses due to translational drag, rotational drag, and gyroscopic torque, respectively, which are defined as follows:

$$\boldsymbol{D_v} = \boldsymbol{K_v} \mathrm{diag}(\boldsymbol{R}_q \dot{\boldsymbol{\xi}}) \boldsymbol{R}_q \dot{\boldsymbol{\xi}}, \tag{4}$$

$$\boldsymbol{\tau_{gyro}} = J_r \sum_{i=1}^{4} \boldsymbol{\eta} \times \begin{bmatrix} 0 \\ 0 \\ (-1)^i \omega_i \end{bmatrix}, \tag{5}$$

$$\boldsymbol{D_\eta} = \boldsymbol{K_\eta} \mathrm{diag}(\boldsymbol{\eta}) \boldsymbol{\eta}, \tag{6}$$

where $\boldsymbol{K_v}$ and $\boldsymbol{K_\eta}$ are the diagonal matrices that define the translational and rotational drag coefficients, respectively, J_r is the rotor moment of inertia, and $\boldsymbol{R_q}$ is the quaternion rotation matrix defined as:

$$\boldsymbol{R}_q = \begin{bmatrix} 2\left(q_0^2+q_1^2\right)-1 & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_3+q_0q_2\right) \\ 2\left(q_0q_3+q_1q_2\right) & 2\left(q_0^2+q_2^2\right)-1 & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_1q_3-q_0q_2\right) & 2\left(q_0q_1+q_2q_3\right) & 2\left(q_0^2+q_3^2\right)-1 \end{bmatrix}.$$

2.1. Rotor Speeds Estimation

Since there are no sensors to measure motor speeds, an estimation is obtained using the relationship between the control inputs and the thrust forces of the motors. Furthermore, it is known that the thrust force of the motor is proportional to the square of its speed, such that:

$$F_i = k_T\omega_i^2 \quad [N] \quad , \quad i = 1,2,3,4. \tag{7}$$

By replacing equation 7 in equation 2, a relationship between control signals and motor speeds is presented. In this way, solving the equation for the speeds, it is determined that:

$$\omega_1 = \sqrt{\frac{1}{4k_T}\left(F_{th} - \frac{\tau_x}{L} - \frac{\tau_y}{L} - \frac{\tau_z}{k_M}\right)},$$

$$\omega_2 = \sqrt{\frac{1}{4k_T}\left(F_{th} + \frac{\tau_x}{L} - \frac{\tau_y}{L} + \frac{\tau_z}{k_M}\right)},$$

$$\omega_3 = \sqrt{\frac{1}{4k_T}\left(F_{th} + \frac{\tau_x}{L} + \frac{\tau_y}{L} - \frac{\tau_z}{k_M}\right)},$$

$$\omega_4 = \sqrt{\frac{1}{4k_T}\left(F_{th} - \frac{\tau_x}{L} + \frac{\tau_y}{L} + \frac{\tau_z}{k_M}\right)}.$$

Speeds are used to calculate the gyroscopic torque given in equation 5. To reduce the computational load caused by the square roots that contain the estimated speeds of the motors, a Taylor series linear approximation of these equations is obtained as follows, based on the inputs $\bar{\boldsymbol{U}} = \begin{bmatrix} mg & 0 & 0 & 0 \end{bmatrix}^T$ (hovering):

$$\bar{\omega}_1 \approx \frac{1}{4}\sqrt{\frac{mg}{k_T}} + \frac{1}{4\sqrt{k_T mg}}\left(F_{th} - \frac{\tau_x}{L} - \frac{\tau_y}{L} - \frac{\tau_z}{k_M}\right),$$

$$\bar{\omega}_2 \approx \frac{1}{4}\sqrt{\frac{mg}{k_T}} + \frac{1}{4\sqrt{k_T mg}}\left(F_{th} + \frac{\tau_x}{L} - \frac{\tau_y}{L} + \frac{\tau_z}{k_M}\right),$$

$$\bar{\omega}_3 \approx \frac{1}{4}\sqrt{\frac{mg}{k_T}} + \frac{1}{4\sqrt{k_T mg}}\left(F_{th} + \frac{\tau_x}{L} + \frac{\tau_y}{L} - \frac{\tau_z}{k_M}\right),$$

$$\bar{\omega}_4 \approx \frac{1}{4}\sqrt{\frac{mg}{k_T}} + \frac{1}{4\sqrt{k_T mg}}\left(F_{th} - \frac{\tau_x}{L} + \frac{\tau_y}{L} + \frac{\tau_z}{k_M}\right).$$

Solving equation 5:

$$\boldsymbol{\tau_{gyro}} = J_r \begin{bmatrix} \omega_y \\ -\omega_x \\ 0 \end{bmatrix} \left(-\omega_1 + \omega_2 - \omega_3 + \omega_4\right).$$

2.2. Energy Consumption Estimation

A key focus of this study is to reduce the power consumption of the quadrotor. For this reason, the consumption from the quadrotor's rotors is considered since it is much higher than that of the other electronic elements. The power consumed by each motor is calculated by:

$$P_i = F_i\omega_i = k_T\omega_i^3 \quad , \quad i = 1,2,3,4.$$

Therefore, the net power consumed by the rotors is:

$$P = \sum_{j=1}^{4} P_i = k_T \sum_{j=1}^{4} \omega_i^3. \tag{8}$$

Furthermore, the energy consumed over time T is related to the power through:

$$E = \int_T P.dt.$$

3. NMPC Controller

NMPC controller design is based on the mathematical non-linear modelling of the system. An optimization problem is defined to minimize tracking error and the control signal. For this, optimal control inputs are calculated by solving an optimization problem in each instant of time. The optimal control problem is represented as [16, 17]:

$$\begin{aligned} \underset{\boldsymbol{u}}{\text{minimize}} \quad & \sum_{k=1}^{N_p} \left[\left(\boldsymbol{\xi}_k - \boldsymbol{r}_k\right)^T \boldsymbol{H}\left(\boldsymbol{\xi}_k - \boldsymbol{r}_k\right) + \boldsymbol{u}_k^T \boldsymbol{R} \boldsymbol{u}_k\right] \\ \text{subject to} \quad & \boldsymbol{x}_{k+1} = \mathcal{RK}_4\left(f\left(\boldsymbol{x}_k, \boldsymbol{u}_k\right), T_s\right), \; k = 1,\ldots,N_p-1, \\ & \boldsymbol{x}^{min} \le \boldsymbol{x}_k \le \boldsymbol{x}^{max}, \qquad k = 1,\ldots,N_p-1, \\ & \boldsymbol{u}^{min} \le \boldsymbol{u}_k \le \boldsymbol{u}^{max}, \qquad k = 1,\ldots,N_p-1, \\ & \|\boldsymbol{q}_k\| = 1, \qquad k = 1,\ldots,N_p, \\ & \boldsymbol{x}_0 = \boldsymbol{x}_{est}, \end{aligned} \tag{9}$$

where $\mathcal{RK}_4$ is the fourth order Runge-Kutta integrator applied to the model given by equation 3 and is evaluated at (x_k, u_k). $\boldsymbol{H}$ and $\boldsymbol{R}$ are diagonal matrices which contain the reference tracking and control weights for the cost function. The estimated states vector $\boldsymbol{x}_{est}$ is fed back at each iteration to compute the next optimal control signal. This algorithm was implemented in Android using C code generated through the CasADi framework.

4. Implementation Results

Performance tests were developed using two C functions created in Matlab via the CasADi framework: one function calculates the optimal control signal $\boldsymbol{u}_k$ and the other feeds back the estimated states $\boldsymbol{x}_{k+1}$ by evaluating the quadrotor dynamic function using a fourth order Runge-Kutta integrator. Estimated states vector $\boldsymbol{x}_{est}$ corresponds to the $\boldsymbol{x}_{k+1}$ state calculated in the previous iteration, which is used as the initial state for the optimal control problem. To implement the flight control algorithm, an *LG Nexus 5X* smartphone with a Hexacore Qualcomm Snapdragon 808 CPU (with a maximum clock speed of 1.8 GHz), 2 *GB* of RAM, and Android 8.0 operating system was used. In addition, three test trajectories were defined: the

Tab. 1. Comparison of MPC controllers implemented on Android.

Model	Trajectory	t_{ev} $[ms]$		RMSE	P_{mean}	E_{mean}
		t_{mean}	σ_t	$[m]$	$[kW]$	$[kJ]$
Non-linear	Square	42.29	10.94	0.09	9.73	779.47
	Helical	59.47	31.82	0.49	9.74	584.91
	Crop	59.47	23.97	0.38	9.74	730.95
Non-linear + Aero	Square.	84.69	39.55	0.07	9.73	779.47
	Helical	86.06	45.48	0.49	9.74	584.91
	Crop	73.15	19.40	0.38	9.74	730.94

square trajectory is defined as a square with side lengths of $2\ m$, whose vertices are located at $(0, 0, 2)$, $(2, 0, 2)$, $(2, 2, 2)$, and $(0, 2, 2)$; the helical trajectory is defined as an ascending spiral with a radius of $2\ m$ and origin at $(0, 0, 3)$ that reaches a height of $8\ m$; and the crop trajectory is represented by a series of lines that simulate movement through a crop, so that the quadrotor starts its trip at position $(0, 5, 2)$ and ends at position $(7.5, 0, 2)$. A prediction horizon $N_p = 20$ is used in the simulations.

In preliminary tests, a comparison was made between two types of MPC controllers: an MPC controller based on the non-linear model without aerodynamics and an MPC controller based on the non-linear model including the mentioned aerodynamic effects. This comparison was carried out to analyze the evaluation times, tracking errors, and estimated power and energy consumption. The results of the tests implemented on the smartphone using the CasADi framework are shown in Table 1, where t_{ev} is the evaluation time with mean evaluation time t_{mean} and standard deviation σ_t, RMSE is the root mean square trajectory tracking error, and P_{mean} and E_{mean} are the mean power and energy consumption. The mean evaluation time for the non-linear MPC control without aerodynamics was $53.74\ ms$, while the average evaluation time for the non-linear MPC control with aerodynamics was $81.30\ ms$. The average trajectory tracking error for the non-linear MPC control without aerodynamics was $0.32\ m$, while for the non-linear MPC control with aerodynamics it was $0.31\ m$. The estimated average power consumption was $9.74\ kW$ for the non-linear MPC control both with and without aerodynamics. The estimated average energy consumption for the non-linear MPC control without aerodynamics was $698.44\ kJ$, while for the non-linear MPC control with aerodynamics it was $698.45\ kJ$.

Subsequently, trajectory tracking tests were performed, where the performance of the non-linear MPC controller based on the model that includes the aerodynamic effects was compared with H_∞ and LQI controllers, which were designed based on the linearized model of the quadrotor, as shown in [18]. Tests were performed using Matlab and Simulink, and results are shown in Table 2. Figure 2 shows square trajectory tracking, where the MPC control shows an RMS error of $0.04\ m$, which is much lower than the tracking error of the H_∞ and LQI controllers. However, estimated average power and energy consumption was slightly higher than the LQI control, but lower than the H_∞ control. Figure 3 shows helical trajectory tracking, where the MPC control shows an RMS error of $0.22\ m$, which is much lower than the tracking error reached by H_∞ and LQI controllers, but the estimated average power and energy consumption was slightly higher than both the LQI and H_∞ controllers. Figure 4 shows crop trajectory tracking, where the MPC controller shows an RMS error of $0.19\ m$, which is almost half that of the tracking error for the H_∞ and LQI controllers, while the estimated average power and energy consumption was slightly higher than for the LQI and H_∞ controllers.

Because flight conditions outside are not ideal for a quadrotor, a translation test is defined where pulses are applied to the position of the system to analyze MPC control disturbance rejection. These disturbances emulate wind flow that can affect the movement of the aircraft along the trajectory. As can be seen in Figure 5, the MPC controller stabilizes the quadrotor to track reference trajectory, reaching a tracking error of $0.07\ m$.

Tab. 2. Comparison of MPC, H_∞, and LQI controllers for trajectory tracking.

Model	Control	RMSE $[m]$	P_{mean} $[kW]$	P_{mean} $[kW]$
Square	MPC	0.04	9.91	647.52
	H_∞	0.07	9.76	634.73
	LQI	0.08	9.75	634.31
Helical	MPC	0.22	9.75	490.78
	H_∞	0.95	9.75	487.83
	LQI	0.84	9.75	487.64
Crop	MPC	0.19	9.74	636.73
	H_∞	0.36	9.73	632.97
	LQI	0.40	9.73	633.01

5. Conclusion

In this work, non-linear model-based predictive control (NMPC) algorithms were implemented to control a quadrotor for trajectory tracking, where an optimization problem must be solved at each sampling instant. Controllers were designed via Matlab

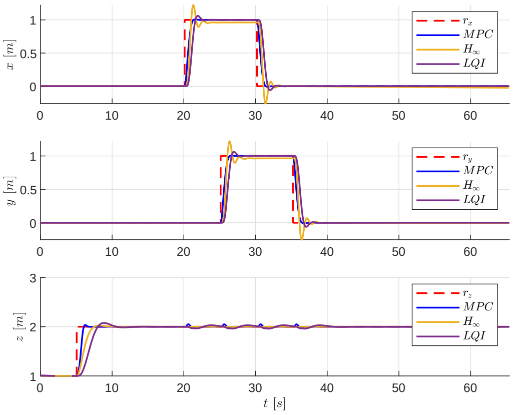

Fig. 2. Comparison of controllers on square trajectory tracking.

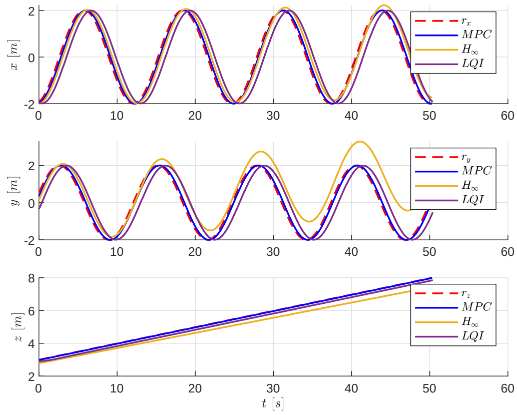

Fig. 3. Comparison of controllers on helical trajectory tracking.

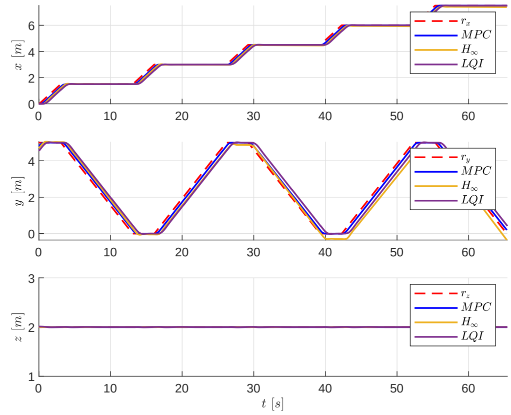

Fig. 4. Comparison of controllers on crop trajectory tracking.

using the CasADi framework and exported as C language files, which were used as libraries to implement an Android-based smartphone application. The main challenge of this development was to execute MPC algorithms and emulate a quadrotor's behavior on a smartphone, taking advantage of its processing power, thus demonstrating the advantages of using smartphones in dynamic system control loops and their ability to handle an MPC controller with computationally-intensive calculations. Also, these type of devices provide various sensors that can be used for state estimation in real implementation. It is recommended to use a smartphone with high processing power, so that calculations can be executed without affecting the sampling time of the system.

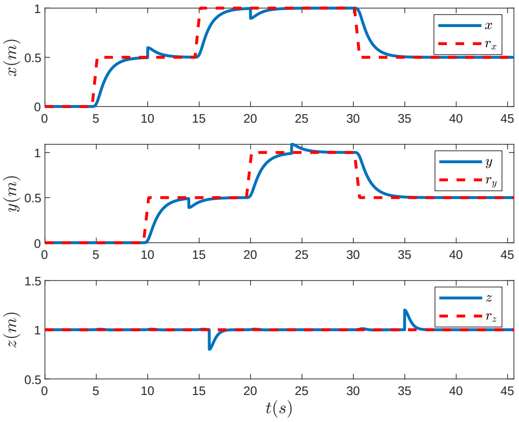

Fig. 5. NMPC controller disturbance rejection.

Establishing the same optimal control problem for both simulation cases (nonlinear MPC with and without aerodynamic effects), a significant reduction in tracking error and slight reduction of estimated energy consumption was obtained using the NMPC controller which considers aerodynamic effects. However, a very similar performance can be achieved with the NMPC controller without aerodynamics, which requires less evaluation time to solve the optimal control problem. However, an evaluation of the aerodynamic effects on the system must be considered according to the structure of the quadrotor used in tests.

A nonlinear MPC (NMPC) controller was compared with H_∞ and LQI controllers [18] in path tracking tests, where NMPC control reached 50% lower tracking error in square trajectory, 80% less for helical trajectory and $40-50\%$ less for the crop trajectory. Estimated energy consumption was about 2% higher than H_∞ and LQI controllers in the square trajectory and about 1% higher in the helical and crop trajectory, due to the control effort. Also, depending on the test trajectory, H_∞ and LQI controllers had to be adjusted to correct the tracking error, while the MPC control used only one design for all tests. It should be noted that NMPC controller was based on the non-linear model of the quadrotor, while the H_∞ and LQI controllers were based on the linearized model, so input thrust compensation may be required to hold the quadrotor on a desired height. It could be noted that the energy consumption was lower in the helical trajectory, which has a smooth shape compared to the square and crop trajectories. This is because abrupt changes in trajectory due to ”corners” require strong movements for the quadrotor and, therefore, a greater control effort is required to follow the reference path.

Although a quadrotor is an inherently unstable system which has complex dynamics, a smooth transient response was achieved by implementing the NMPC controller using a smartphone. It was observed that the settling time obtained by using the NMPC controller was lower than in the H_∞ and LQI controllers (about 80% less), which contributed to the reduction of trajectory tracking error. This shows that the NMPC control algorithm developed in this work can be implemented for real-life applications where the aircraft can be tested on demanding paths.

AUTHORS

Luis García* – School of Electrical and Electronic Engineering, Universidad del Valle, Ciudad Universitaria Melendez, Calle 13 # 100-00, Cali, Colombia, e-mail: luis.linares@correounivalle.edu.co, www: https://gici.univalle.edu.co/.

Esteban Rosero – School of Electrical and Electronic Engineering, Universidad del Valle, Ciudad Universitaria Melendez, Calle 13 # 100-00, Cali, Colombia, e-mail: esteban.rosero@correounivalle.edu.co, www: https://gici.univalle.edu.co/.

*Corresponding author

REFERENCES

[1] A. Banerjee and A. Roychoudhury, “Future of mobile software for smartphones and drones: Energy and performance,” pp. 1–12, 2017.

[2] J. A. Frank, A. Brill, J. Bae, and V. Kapila, “Exploring the role of a smartphone as a motion sensing and control device in the wireless networked control of a motor test-bed,” in *2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2015, pp. 328–335.

[3] C. Bodenstein, M. Tremer, J. Overhoff, and R. P. Würtz, “A smartphone-controlled autonomous robot,” in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2015, pp. 2314–2321.

[4] L. Garcia, A. Astudillo, and E. Rosero, “Fast model predictive control on a smartphone-based flight controller,” in *2019 IEEE 4th Colombian Conference on Automatic Control (CCAC)*, 2019, pp. 1–6.

[5] N. Kreciglowa, K. Karydis, and V. Kumar, “Energy efficiency of trajectory generation methods for stop-and-go aerial robot navigation,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 656–662.

[6] J. Svacha, K. Mohta, and V. Kumar, “Improving quadrotor trajectory tracking by compensating for aerodynamic effects,” in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 860–866.

[7] F. Yacef, N. Rizoug, L. Degaa, O. Bouhali, and M. Hamerlain, “Trajectory optimisation for a quadrotor helicopter considering energy consumption,” in *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2017, pp. 1030–1035.

[8] Z. Wang, K. Akiyama, K. Nonaka, and K. Sekiguchi, “Experimental verification of the model predictive control with disturbance rejection for quadrotors,” in *2015 54th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2015, pp. 778–783.

[9] K. Shimada and T. Nishida, “Particle filter-model predictive control of quadcopters,” in *Proceedings of the 2014 International Conference on Advanced Mechatronic Systems*, 2014, pp. 421–424.

[10] R. Singhal and P. B. Sujit, “3d trajectory tracking for a quadcopter using mpc on a 3d terrain,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 1385–1390.

[11] A. Bemporad and C. Rocchi, “Decentralized linear time-varying model predictive control of a formation of unmanned aerial vehicles,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, 2011, pp. 7488–7493.

[12] M. E. Guerrero-Sanchez, H. Abaunza, P. Castillo, R. Lozano, and C. D. García-Beltrán, “Quadrotor energy-based control laws: A unit-quaternion approach,” *Journal of Intelligent & Robotic Systems*, no. 2, pp. 347–377, 2017.

[13] J. Cariño, H. Abaunza, and P. Castillo, “Quadrotor quaternion control,” in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 825–831.

[14] W. Dong, G.-Y. Gu, X. Zhu, and H. Ding, “Modeling and control of a quadrotor uav with aerodynamic concepts,” *International Journal of Aerospace and Mechanical Engineering*, no. 5, pp. 901–906, 2013.

[15] A. Chovancová, T. Fico, L. Chovanec, and P. Hubinsk, “Mathematical modelling and parameter identification of quadrotor (a survey),” *Procedia Engineering*, pp. 172–181, 2014.

[16] T. T. Ribeiro, A. G. Conceiçao, I. Sa, and P. Corke, “Nonlinear model predictive formation control for quadcopters,” *IFAC-PapersOnLine*, no. 19, pp. 39–44, 2015.

[17] M. Neunert, C. De Crousaz, F. Furrer, M. Kamel, F. Farshidian, R. Siegwart, and J. Buchli, “Fast nonlinear model predictive control for unified trajectory optimization and tracking,” in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1398–1404.

[18] A. Astudillo, B. Bacca, and E. Rosero, “Optimal and robust controllers design for a smartphone-based quadrotor,” in *2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC)*, 2017, pp. 1–6.