

# Data-Driven Control Strategies for Rotary Wing Aerial Vehicles

Simão Caeiro<sup>1</sup>, Bruno J. Guerreiro<sup>2,1</sup>, and Rita Cunha<sup>1</sup>

<sup>1</sup> Institute for Systems and Robotics (ISR/LARSYS), Instituto Superior Técnico,  
University of Lisbon, 1049-001 Lisbon, Portugal

<sup>2</sup> NOVA School of Science and Technology, UNINOVA-CTS, LASI, NOVA University  
Lisbon, 2829-516 Caparica, Portugal

E-mails: `simao.caeiro@tecnico.ulisboa.pt`,  
`bj.guerreiro@fct.unl.pt`, `rita@isr.tecnico.ulisboa.pt`

**Abstract.** To evaluate the performance of data-driven control methods applied to Unmanned Aerial Vehicles (UAVs), this work addresses the implementation of these strategies, particularly the Data-enabled Predictive Control (DeePC) algorithm. This strategy computes optimal controls for unknown systems through real-time output feedback using a receding horizon implementation. Moreover, this research investigates the influence of different hyperparameters on the DeePC's performance and conducts a realistic comparison between this method and two model-based control approaches: Linear Quadratic Regulator (LQR) and Model Predictive Control (MPC). The simulation results validate the applicability of DeePC algorithm and highlight its superior robustness to system degradation and yaw calibration errors. However, it is less suitable for complex nonlinear systems subject to aggressive trajectories.

**Keywords:** Data-driven Control, Data-enabled Predictive Control, Unmanned Aerial Vehicles, Performance Evaluation

## 1 Introduction

Unmanned Aerial Vehicles (UAVs) have grown significantly in popularity and notoriety in recent years, largely attributable to their enhanced stability and endurance in several military and civil operations [7]. The control of these systems is conventionally tackled through model-based approaches, such as the well-known Proportional-Integrative-Derivative (PID), Linear Quadratic Regulator (LQR), and Model Predictive Control (MPC) techniques [1, 2, 4].

Nevertheless, with the increasing complexity of systems and the widespread availability of data, there has been a recent trend in the literature where classical model-based techniques have been superseded by data-driven methodologies [6]. These strategies, referred to as data-driven control methods, are particularly advantageous in complex systems where system identification is excessively time-consuming and complex. In [3], the authors introduced the Data-enabled Predictive Control (DeePC) method, which is based on the *Fundamental Lemma* [10]

and does not require function learning or system identification. This algorithm uses a finite collected dataset to learn the unknown system’s behaviour and then applies real-time output feedback to compute optimal controls that guide the system towards a desired path. Unlike other learning-based control methods that use machine learning techniques, this approach does not need time-consuming offline learning and can handle nonlinear systems in real time.

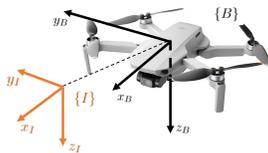
The main objective of this work is to evaluate the performance and key elements of data-driven control techniques applied to quadrotors. Thus, to achieve this goal, the DeePC algorithm is implemented, adapted to the problem at hand, and analysed in detail. Subsequently, the main contributions of this paper encompass a performance evaluation of the DeePC method using realistic simulations and a detailed comparison between this algorithm and two conventional model-based control methods: LQR and MPC. This comparative analysis includes an assessment of the algorithms’ response to a simple trajectory and their adaptability to performance degradation of the system’s inner loop and yaw calibration errors in the system measurements.

The remainder of this document is structured as follows. In Section 2, a description of both the linear and nonlinear models of the quadrotor’s dynamics is presented, followed by a brief review of the conventional model-based methods relevant to this work. Section 3 formulates the DeePC optimisation problem. Section 4 details the simulation setup and results, including the findings obtained from the comparison of DeePC with the conventional control methods. Finally, Section 5 provides some concluding remarks and suggestions for future work.

## 2 Theoretical Background

### 2.1 Quadrotor Model

The nonlinear dynamics are described in both the body-fixed reference frame  $\{B\}$  and the inertial reference frame  $\{I\}$ , as illustrated in Fig. 1. The unit vectors along the axis of the body-fixed frame  $\{B\}$  are labelled as  $\{\mathbf{x}_B, \mathbf{y}_B, \mathbf{z}_B\}$ , while the unit vectors along the inertial frame  $\{I\}$  axis are denoted by  $\{\mathbf{x}_I, \mathbf{y}_I, \mathbf{z}_I\}$ . It is assumed that the origin of the body-fixed frame  $\{B\}$  coincides with the centre of mass of the quadrotor. The  $x_B$  and  $y_B$  axes are situated in the plane defined by the four rotors, while the  $z_B$  axis is perpendicular to this plane.



**Fig. 1.** Illustration of the reference frames used.

Let  $\mathbf{p} = [x, y, z]^T$  denote the position vector of the centre of mass of the UAV in  $\{I\}$ . Let  $\mathbf{R}$  represent the rotation matrix from  $\{B\}$  to  $\{I\}$ , which can be parameterised by  $\mathbf{R}(\boldsymbol{\lambda})$  with  $\boldsymbol{\lambda} = [\phi, \theta, \psi]^T$ , where the Euler angles  $\phi$ ,  $\theta$ , and  $\psi$  correspond to the roll, pitch and yaw angles, respectively. Consider  $\boldsymbol{\omega} = [p, q, r]^T$  as the angular velocity of  $\{B\}$  relative to  $\{I\}$ , expressed in  $\{B\}$ . Let the control input to the system  $\mathbf{u}$  comprise the total thrust  $T \in \mathbb{R}^+$  and the body torques  $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^T \in \mathbb{R}^3$ , both defined in  $\{B\}$ . Hence, by considering the state vector  $\mathbf{x} = [\mathbf{p}^T, \mathbf{v}^T, \boldsymbol{\lambda}^T, \boldsymbol{\omega}^T]^T$  and the input vector  $\mathbf{u} = [T, \boldsymbol{\tau}^T]^T$ , it is possible to express the nonlinear dynamics of the quadrotor in the compact form  $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$  as follows:

$$f(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \mathbf{v} \\ gz_I - \frac{1}{m_b} \mathbf{R}T \mathbf{z}_I \\ \mathbf{Q}(\boldsymbol{\lambda})\boldsymbol{\omega} \\ -\mathbf{J}^{-1}S(\boldsymbol{\omega})\mathbf{J}\boldsymbol{\omega} + \mathbf{J}^{-1}\boldsymbol{\tau} \end{bmatrix}, \quad (1)$$

where  $m_b \in \mathbb{R}^+$  represents the mass of the quadrotor,  $\mathbf{J} \in \mathbb{R}^{3 \times 3}$  denotes the inertia tensor described in  $\{B\}$ , the constant  $g$  corresponds to the Earth's gravity,  $S(\cdot)$  denotes the skew-symmetric operator such that  $S(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$ , and

$$\mathbf{Q}(\boldsymbol{\lambda}) = \begin{bmatrix} 1 \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix}.$$

To linearise the obtained model, it is assumed that the equilibrium point corresponds to the quadrotor's hover condition, at a particular position  $\mathbf{p}_0$  and for  $\psi = 0$ . From the Taylor series expansion, the linearisation of (1) around this equilibrium point can be expressed in the form  $\dot{\mathbf{x}} = \mathbf{A}_c \mathbf{x} + \mathbf{B}_c \mathbf{u}$ . Thus, the matrices  $\mathbf{A}_c$  and  $\mathbf{B}_c$  are given by

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{G} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\ -\frac{1}{m_b} \mathbf{z}_I & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{J}^{-1} \end{bmatrix}, \quad \text{where } \mathbf{G} = \begin{bmatrix} 0 & -g & 0 \\ g & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

In addition to the linearised model deduced above, this work also utilises another quadrotor model, whose state and input vectors are, respectively, given by  $\mathbf{x} = [\mathbf{p}^T, \mathbf{v}^T, \boldsymbol{\lambda}^T]$  and  $\mathbf{u} = [T, \boldsymbol{\omega}^T]$ . By employing an identical linearisation process and considering the relevant formulations in (1), the ensuing matrices can be obtained:

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{G} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad \mathbf{B}_c = \begin{bmatrix} \mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\ -\frac{1}{m_b} \mathbf{z}_I & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 1} & \mathbf{I}_{3 \times 3} \end{bmatrix}.$$

Finally, the discretisation of this linear system is necessary to allow the implementation of the subsequent proposed methods. Thus, using the zero-order hold method, the linear system can be described in the following equivalent discrete

state space representation

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \\ \mathbf{y}_k &= \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k,\end{aligned}\tag{2}$$

where  $\mathbf{A} = e^{\mathbf{A}_c T_s}$ ,  $\mathbf{B} = \left(\int_0^{T_s} e^{\mathbf{A}_c \tau} d\tau\right) \mathbf{B}_c$ ,  $\mathbf{C} = \mathbf{C}_c$ ,  $\mathbf{D} = \mathbf{D}_c$ , and  $T_s$  is the sampling period.

## 2.2 Linear Quadratic Regulator

In terms of model-based control approaches, the LQR is the first one to be presented. This method is a modern control strategy that aims to determine a feedback control law so that the system to be controlled can meet physical constraints while also minimising a quadratic cost function [8].

Consider the discrete system described by (2), where  $\mathbf{x} \in \mathbb{R}^n$  is the vector of state-space variables,  $\mathbf{y} \in \mathbb{R}^p$  is the system's output, and  $\mathbf{u} \in \mathbb{R}^m$  is the system's input. The optimal regulator problem computes the gain matrix  $\mathbf{K}$  of the optimal control vector  $\mathbf{u}_k = -\mathbf{K}\mathbf{x}_k$ , in order to minimise the cost function

$$J = \sum_{k=0}^{\infty} \mathbf{x}_k^T \mathbf{Q} \mathbf{x}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k,\tag{3}$$

where  $\mathbf{Q} \in \mathbb{R}^{n \times n}$ , denominated as state weighting matrix, is a positive semi-definitive matrix and  $\mathbf{R} \in \mathbb{R}^{m \times m}$ , designated as control weighting matrix, is a positive definitive matrix. It is also important to note that since this is an infinite time control problem, the control solution turns into a steady-state solution, leading to a constant optimal gain matrix  $\mathbf{K}$  that minimises (3), and can be computed using the discrete-time *Riccati* equation.

## 2.3 Model Predictive Control

The fundamental principle of model predictive control (MPC) is to predict the future behaviour of the controlled system over a specified time horizon and compute an optimal control input that minimises a chosen cost function while ensuring the satisfaction of the system constraints.

The classical MPC algorithm uses a receding horizon approach to solve the following reference tracking optimal control problem:

$$\begin{aligned}\min_{\mathbf{u}, \mathbf{x}, \mathbf{y}} \quad & \sum_{k=0}^{T_f-1} (\mathbf{y}_k - \bar{\mathbf{y}}_k)^T \mathbf{Q} (\mathbf{y}_k - \bar{\mathbf{y}}_k) + (\mathbf{u}_k - \bar{\mathbf{u}}_k)^T \mathbf{R} (\mathbf{u}_k - \bar{\mathbf{u}}_k) \\ \text{s.t.} \quad & \mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k, \forall k \in \{0, \dots, T_f - 1\}, \\ & \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{D}\mathbf{u}_k, \forall k \in \{0, \dots, T_f - 1\}, \\ & \mathbf{x}_0 = \hat{\mathbf{x}}_t, \\ & \mathbf{u}_k \in \mathcal{U}, \forall k \in \{0, \dots, T_f - 1\}, \\ & \mathbf{y}_k \in \mathcal{Y}, \forall k \in \{0, \dots, T_f - 1\},\end{aligned}$$

where  $t$  is the current time,  $k$  is the temporal instance of the predictive horizon window,  $T_f \in \mathbb{N}$  is the time horizon,  $\mathbf{u}, \mathbf{x}, \mathbf{y}$  are the decision variables,  $\mathcal{U} \subseteq \mathbb{R}^m$  is an input constraint set,  $\mathcal{Y} \subseteq \mathbb{R}^p$  is an output constraint set,  $\bar{\mathbf{u}}(k) \in \mathbb{R}^m$  and  $\bar{\mathbf{y}}(k) \in \mathbb{R}^p$  are, respectively, the desired input and output reference calculated for each iteration of the algorithm,  $\mathbf{Q} \in \mathbb{R}^{p \times p}$  is the output cost matrix,  $\mathbf{R} \in \mathbb{R}^{m \times m}$  is the control cost matrix, and  $\hat{\mathbf{x}}_t$  is the estimated state at time  $t$ .

### 3 Data-enabled Predictive Control

#### 3.1 Algorithm

Consider an unknown deterministic discrete-time LTI system represented by (2). Assuming that (2) is given in its minimal realisation, it is possible to ensure controllability and observability properties of the represented system. The lag of the system (2) is defined by the smallest integer  $\ell \in \mathbb{Z}_{\geq 0}$  for which the observability matrix  $\mathcal{O}_\ell(\mathbf{A}, \mathbf{C}) := [\mathbf{C}, \mathbf{C}\mathbf{A}, \dots, \mathbf{C}\mathbf{A}^{\ell-1}]^T$  has rank  $n$ .

**Definition 1 (Persistency of excitation [10]).** *Let  $L, T \in \mathbb{Z}_{\geq 0}$  such that  $T \geq L$ . The sequence of signals  $\mathbf{u} = \{\mathbf{u}_k\}_{k=0}^T \in \mathbb{R}^{mT}$  is persistently exciting of order  $L$  if the Hankel matrix*

$$\mathcal{H}_L(\mathbf{u}) := \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_{T-L+2} \\ \mathbf{u}_2 & \mathbf{u}_3 & \dots & \mathbf{u}_{T-L+2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{u}_L & \mathbf{u}_{L+1} & \dots & \mathbf{u}_T \end{bmatrix}$$

has full row rank.

The term persistently exciting refers to an input signal that is sufficiently rich and long to excite the system and produce an output sequence that is representative of its behaviour. Note that the DeePC algorithm is based on the following fundamental result.

**Theorem 1.** [10] *Consider (2) and let  $T_d, L \in \mathbb{Z}_{\geq 0}$ . Let the sequences of collected input/output data  $(\mathbf{u}_d, \mathbf{y}_d) = \{\mathbf{u}_{d,k}, \mathbf{y}_{d,k}\}_{k=0}^{T_d}$  be a trajectory of (2) of length  $T_d$ , assuming that  $\mathbf{u}_d$  is persistently exciting of order  $L$ . Hence,  $(\mathbf{u}, \mathbf{y}) = \{\mathbf{u}_k, \mathbf{y}_k\}_{k=0}^L$  is a trajectory of (2) if and only if there exists  $\mathbf{g} \in \mathbb{R}^{T_d-L+1}$  such that*

$$\begin{bmatrix} \mathcal{H}_L(\mathbf{u}_d) \\ \mathcal{H}_L(\mathbf{y}_d) \end{bmatrix} \mathbf{g} = \begin{bmatrix} \mathbf{u} \\ \mathbf{y} \end{bmatrix}.$$

Note that, in this particular case, considering that  $\mathbf{u}_d$  is persistently exciting of order  $L$ , it implies that the condition  $T_d \geq (m+1)L - 1$  must be satisfied.

Therefore, the Hankel matrices composed by input/output collected data, are divided into two parts,

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{U}_f \end{bmatrix} := \mathcal{H}_{T_{ini}+T_f}(\mathbf{u}_d), \quad \begin{bmatrix} \mathbf{Y}_p \\ \mathbf{Y}_f \end{bmatrix} := \mathcal{H}_{T_{ini}+T_f}(\mathbf{y}_d),$$

where  $\mathbf{U}_p$  consists of the first  $T_{ini}$  block rows of  $\mathcal{H}_{T_{ini}+T_f}(\mathbf{u}_d)$  and  $\mathbf{U}_f$  consists of the last  $T_f$  block rows of  $\mathcal{H}_{T_{ini}+T_f}(\mathbf{u}_d)$  (similarly for  $\mathbf{Y}_p$  and  $\mathbf{Y}_f$ ). Hence, the data in  $\mathbf{U}_p$  and  $\mathbf{Y}_p$  are used to estimate the initial conditions, whereas the data in  $\mathbf{U}_f$  and  $\mathbf{Y}_f$  are used to predict future trajectories.

From Theorem 1,  $(\mathbf{u}, \mathbf{y}) = \{\mathbf{u}_k, \mathbf{y}_k\}_{k=0}^{T_f-1}$  is a possible future trajectory of (2) if and only if there exists  $\mathbf{g} \in \mathbb{R}^{T_d - T_{ini} - T_f + 1}$  such that

$$\begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \\ \mathbf{Y}_f \end{bmatrix} \mathbf{g} = \begin{bmatrix} \mathbf{u}_{ini} \\ \mathbf{y}_{ini} \\ \mathbf{u} \\ \mathbf{y} \end{bmatrix}. \quad (4)$$

In order to uniquely fix the initial condition from which the future trajectory departs, it is necessary to ensure that  $T_{ini} \geq \ell$ . Therefore, this condition also implies that the predicted trajectory calculated by  $\mathbf{y} = \mathbf{Y}_f \mathbf{g}$  is unique.

Given a reference input  $\mathbf{u}_r \in \mathbb{R}^m$ , a reference trajectory  $\mathbf{y}_r \in \mathbb{R}^p$ , past input/output data  $[\mathbf{u}_{ini}, \mathbf{y}_{ini}]^T$ , an input constraint set  $\mathcal{U} \subseteq \mathbb{R}^m$ , an output constraint set  $\mathcal{Y} \subseteq \mathbb{R}^p$ , an output cost matrix  $\mathbf{Q} \in \mathbb{R}^{p \times p}$ , and a control cost matrix  $\mathbf{R} \in \mathbb{R}^{m \times m}$ , it is possible to formulate the following data-driven optimisation problem:

$$\begin{aligned} \min_{\mathbf{g}, \mathbf{u}, \mathbf{y}} \quad & \sum_{k=0}^{T_f-1} (\mathbf{y}_k - \mathbf{y}_r)^T \mathbf{Q} (\mathbf{y}_k - \mathbf{y}_r) + (\mathbf{u}_k - \mathbf{u}_r)^T \mathbf{R} (\mathbf{u}_k - \mathbf{u}_r) \\ \text{s.t.} \quad & (4) \\ & \mathbf{u}_k \in \mathcal{U}, \forall k \in \{0, \dots, T_f - 1\}, \\ & \mathbf{y}_k \in \mathcal{Y}, \forall k \in \{0, \dots, T_f - 1\}. \end{aligned} \quad (5)$$

### 3.2 Regularised Algorithm

To implement the DeePC algorithm on a nonlinear system corrupted by process noise, such as a real-world quadrotor, it is necessary to include some regularisations in the optimal control problem (5), resulting in the following regularised optimisation problem:

$$\begin{aligned} \min_{\mathbf{g}, \mathbf{u}, \mathbf{y}, \boldsymbol{\sigma}_y} \quad & \sum_{k=0}^{T_f-1} c(\mathbf{u}_k, \mathbf{y}_k) + \lambda_g \|\mathbf{g}\|_2 + \lambda_y \|\boldsymbol{\sigma}_y\|_2 \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{U}_p \\ \mathbf{Y}_p \\ \mathbf{U}_f \\ \mathbf{Y}_f \end{bmatrix} \mathbf{g} = \begin{bmatrix} \mathbf{u}_{ini} \\ \mathbf{y}_{ini} \\ \mathbf{u} \\ \mathbf{y} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{\sigma}_y \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ & \mathbf{u}_k \in \mathcal{U}, \forall k \in \{0, \dots, T_f - 1\}, \\ & \mathbf{y}_k \in \mathcal{Y}, \forall k \in \{0, \dots, T_f - 1\}, \end{aligned} \quad (6)$$

where  $c(\mathbf{u}_k, \mathbf{y}_k) = (\mathbf{y}_k - \mathbf{y}_r)^T \mathbf{Q} (\mathbf{y}_k - \mathbf{y}_r) + (\mathbf{u}_k - \mathbf{u}_r)^T \mathbf{R} (\mathbf{u}_k - \mathbf{u}_r)$ ,  $\boldsymbol{\sigma}_y \in \mathbb{R}^{p T_{ini}}$  is an auxiliary slack variable, and  $\lambda_g, \lambda_y \in \mathbb{R}_{>0}$  are regularisation parameters. The

inclusion of a two-norm penalty of the slack variable  $\sigma_y$  is primarily intended to always ensure the feasibility of the constraint equation. Moreover, it is intended to choose  $\lambda_y$  sufficiently large, in such a way that the condition  $\sigma_y \neq 0$  is only verified in cases where the constraint is infeasible. Furthermore, the inclusion of the two-norm regularisation on  $\mathbf{g}$  is a common technique in a distributionally robust problem formulation.

Finally, the DeePC method, in which (6) is implemented in a receding horizon approach, is summarised in Algorithm 1.

---

**Algorithm 1** Regularised DeePC [5]

---

**Input:**  $T_d, T_f, T_{ini}$ ,  $\mathcal{H} = [\mathbf{U}_p^T, \mathbf{Y}_p^T, \mathbf{U}_f^T, \mathbf{Y}_f^T]^T$ , input and output references  $(\mathbf{u}_r, \mathbf{y}_r)$ , constraint sets  $\mathcal{U}$  and  $\mathcal{Y}$ , performance matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , regularisation parameters  $(\lambda_g, \lambda_y)$ , and past input/output data  $(\mathbf{u}_{ini}, \mathbf{y}_{ini})$ .

1. Solve (6) for  $\mathbf{g}^*$ .
  2. Compute the optimal input sequence  $\mathbf{u}^* = \mathbf{U}_f \mathbf{g}^*$ .
  3. Apply inputs  $(\mathbf{u}_t^T, \dots, \mathbf{u}_{t+s}^T)^T = (\mathbf{u}_0^{*T}, \dots, \mathbf{u}_s^{*T})^T$  for some  $s \leq T_f - 1$ .
  4. Set  $t$  to  $t + s$  and update  $u_{ini}$  and  $y_{ini}$  to the  $T_{ini}$  most recent past input/output measurements.
  5. Return to 1.
- 

### 3.3 Data Collection

As mentioned above, the input signal used to fill the Hankel matrices must be persistently exciting of sufficient order. Therefore, it is imperative to carefully collect the data for the successful implementation of this data-driven control algorithm. To ensure the repeatability of results, this data was collected by applying a Pseudorandom Binary Sequence (PRBS) added to an existing simple hover controller.

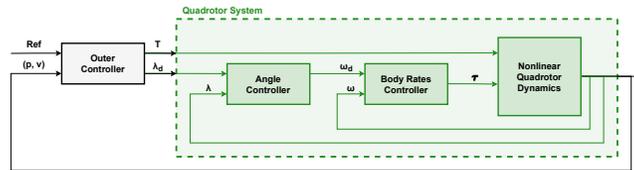
In this work, the PRBS excitation signal was generated using MATLAB with the following steps:

1. A PRBS is generated as the basis for creating the excitation signals used for each system input.
2. Select only the first  $T_d$  elements of the generated PRBS.
3. Determine the desired amplitude for the excitation signal of the first input of the system.
4. To generate the excitation signal linked with the subsequent system input, execute a circular shift of the positions of the original PRBS elements before performing the amplitude multiplication.
5. Repeat the previous step for the remaining system inputs.

## 4 Implementation and Results

This section presents the setup, evaluation, and discussion of the simulation results, which are obtained using a realistic model implemented in MATLAB/Simulink, employing a 3DR<sup>®</sup> Iris+ quadrotor, based on [5]. The results were obtained using a computer with an Intel Core i7-1165G7 CPU 2.80GHz, and 16GB of RAM, whereas the optimisation problems of the DeePC algorithm were solved using the OSQP solver [9].

This work implements a different control architecture relative to the one presented in [5], which addresses some problems associated with the use of body-rate commands for system actuation. Fig. 2 depicts the block diagram of the implemented cascade control architecture for the nonlinear case.



**Fig. 2.** Schematic of control architecture for nonlinear quadrotor dynamics (1).

Furthermore, to successfully implement the regularised DeePC method, it is crucial to tune the following hyperparameters:  $T_d$ , the total number of data points used to build the Hankel matrices;  $T_{ini}$ , the time horizon used for initial condition estimation;  $T_f$ , the prediction time horizon;  $\lambda_y$ , the weight on the regularisation of the initial condition constraint;  $\lambda_g$ , the weight on the regularisation of  $\mathbf{g}$ ;  $\mathbf{Q}$ , the tracking error cost matrix; and  $\mathbf{R}$ , the control effort cost matrix. In addition to these hyperparameters, the excitation amplitudes employed in the PRBS generation are fundamental parameters in the data collection phase, thus having a significant influence on the performance of the DeePC algorithm. The subsequent results are obtained using the following hyperparameters:  $T_{ini}=5$ ;  $T_d=900$ ;  $T_f = 50$ ;  $\lambda_g = 500$ ;  $\lambda_y = 7.5 \times 10^8$ ;  $\mathbf{Q} = \text{diag}(40, 40, 500, 0, 0, 0, 0, 40)$ ; and  $\mathbf{R} = \text{diag}(0.5, 20, 20, 2)$ . For these hyperparameters, the DeePC algorithm's average computing time was 12.4 ms, which compares with the sampling time of 40 ms for the outer loop.

Finally, regarding constraint sets, the control input constraint set  $\mathcal{U}$  is given by  $T \in [7.906, 30.094]\text{N}$ ,  $\phi_d, \theta_d \in [-\frac{\pi}{4}, \frac{\pi}{4}]$  rad, whereas the the output constraint set  $\mathcal{Y}$  is denoted by  $x, y, z \in [-5, 5]$  m.

### 4.1 Influence of Hyperparameters

To study the impact of each hyperparameter on the performance of the DeePC controller, series of experiments were conducted, changing only the value of a

single hyperparameter at each time. Subsequently, the performance of the DeePC controller in each scenario is evaluated by analysing the average overshoot  $S$ , maximum settling time  $t_s$ , average algorithm computation time  $t_c$ , and average static error  $e$ .

**$T_{ini}$**  : An increase of this hyperparameter tends to directly increase both  $S$  and  $t_c$  parameters. However, it should be noted that the selected value was not the minimum possible option, taking into account that the settling time does not evolve as the other parameters. Corroborating this observation, it is important to emphasise that this particular hyperparameter determines the time horizon used for initial condition estimation and consequently, choosing an excessively small value would fail to expose certain nonlinearities inherent in the system's response to the controller.

**$T_d$**  : For an excessively low value of this hyperparameter, the system stabilises in a position with an average static error of 0.40m. In addition, an increase in  $T_d$  leads to an unequivocal increase in  $t_c$ . This finding was already expected since  $T_d$  is directly associated with the dimension of the Hankel matrices used by the algorithm in the optimisation problem, defined in (6). Thus, a higher value of  $T_d$  implies a larger size of the Hankel matrices, which consequently leads to a higher computation time of the DeePC algorithm.

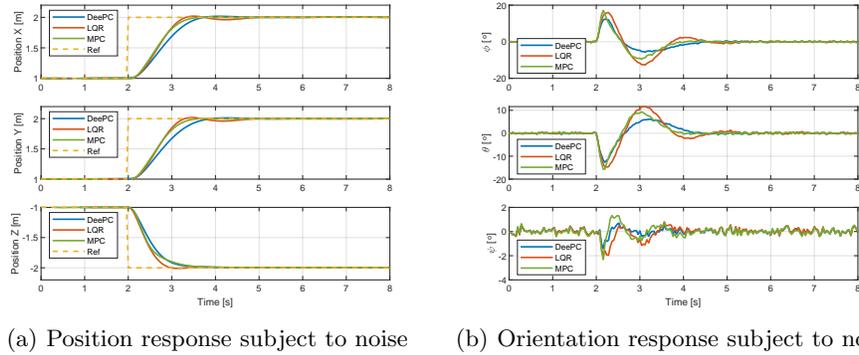
**$T_f$**  : For low values of this parameter, the obtained responses are characterised by a high value of  $S$  and  $t_s$ . However, there is a threshold beyond which augmenting  $T_f$  does not improve the performance of the DeePC controller. Instead, only the significant increase of  $t_c$  is verified.

**$\lambda_g$**  : Assuming a null value for this regularisation parameter results in a static error of 1.03m. Moreover, an increase in  $\lambda_g$  produces a softer but significantly slower response. On the other hand, low values of this parameter lead to an oscillatory response.

**$\lambda_y$**  : The non-utilisation of this regularisation parameter leads to an infeasible optimisation problem and consequently, the DeePC controller fails to stabilise the system. Additionally, there is a threshold from which the results obtained are similar. This observation reinforces the notion that the crucial factor is to select a value of  $\lambda_y$  sufficiently high to render the optimisation problem feasible, regardless of its precise tuning.

## 4.2 Comparison of the Performance of Control Methods

After a detailed study of the DeePC algorithm, this subsection aims to compare the performance of the DeePC controller with two conventional model-based controllers: the LQR and MPC controllers. In this paper, these traditional controllers were implemented according to the information presented in Sections 2.2 and 2.3. Fig. 3 shows the system responses to a unit step employing the DeePC, LQR, and MPC controllers. It is important to highlight that the results presented were obtained using the same inner-loop controllers of the nonlinear control architecture B. The initial analysis shows a satisfactory performance of all controllers. It can also be inferred that, in general, the DeePC responses are characterised by a longer settling time when compared to the others. Neverthe-



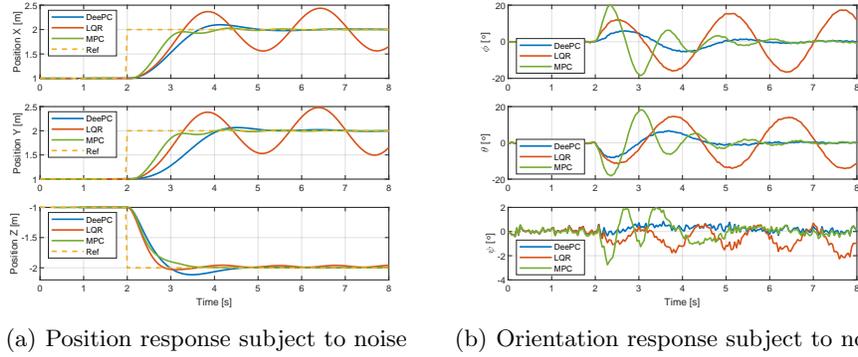
**Fig. 3.** Comparison of the performance of DeePC, LQR, and MPC controllers.

less, these results allow the conclusion that the DeePC algorithm exhibits similar performance to the other two conventional model-based control approaches, when the quadrotor system is subjected to simple trajectories.

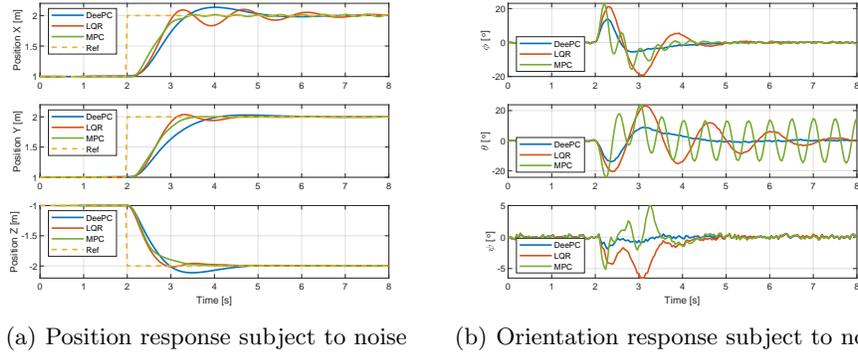
The next step is to study the robustness of the implemented methods to performance degradation of the inner-loop controllers. Hence, the gains of these controllers were modified to slow down their response. Starting with the inner angle controller, its gains were obtained through the multiplication of the original values by  $\frac{1}{3}$ . Regarding the inner body rates PID controller, only the proportional gains were modified in the same proportion as for the inner angle controller.

Figs. 4 and 5 depict, respectively, the effect of the degradation of the inner angle controller and the inner body rates controller on the performance of the DeePC, LQR and MPC controllers. In summary, it can be concluded that the DeePC controller is the least influenced method by the performance degradation of the inner loop controllers. This robustness demonstrates the adaptive properties of this data-driven control method, which is one of the advantages of DeePC over conventional model-based methods.

Finally, the effect of the presence of a bias in the system measurements was evaluated. Considering that it is not always possible to obtain accurate measurements of the yaw angle in a real-world experiment, it was decided to simulate a yaw miscalibration, by introducing a  $25^\circ$  offset between the true and measured yaw angles. It should be noted that this value is excessively high, but it was selected to highlight the differences between the results of the different controllers. Moreover, it can be concluded that the presence of this yaw error in system measurements yields a deterioration in the LQR and MPC performance. Concerning the DeePC method, it is possible to note that this algorithm provides robustness to this bias error. Thus, it can be concluded once again that the DeePC algorithm has the ability to adapt to the unknown operating conditions of the system, in contrast to the other model-based control methods.



**Fig. 4.** Comparison of the controllers' performance facing inner angle controller degradation.



**Fig. 5.** Comparison of the controllers' performance facing inner body rates controller degradation.

## 5 Conclusions

The main goal of this paper was to develop a data-driven control technique suitable for the control of a quadrotor system. The implementation of the DeePC algorithm was first proposed and a realistic simulation was conducted to evaluate the method on models of different complexity. Finally, a detailed comparison between the DeePC and other model-based methods was performed, to situate this framework in relation to conventional control methods.

The implementation of the DeePC controller proves that it does not require access to full-state measurements, unlike traditional control methods. Furthermore, the simulation results show that the DeePC algorithm performs similarly to MPC and LQR approaches when the quadrotor system is subjected to simple trajectories. This data-driven method also demonstrates greater robustness to

the performance degradation of inner-loop controllers and the presence of a yaw calibration error than the conventional approaches. Nevertheless, it is important to note that the implementation of the DeePC method becomes impractical when tracking aggressive trajectories, whereas the data collection step may fail to capture the essential dynamics of nonlinear systems, and the dimension of the Hankel matrices may significantly increase in the computational time.

For future work, the presented algorithms could be validated through Software-in-the-Loop (SITL) simulations and experimental results, which would provide more guarantees of the performance of the proposed algorithm. Additionally, to increase the applicability of the DeePC method, the data collected for the Hankel matrices could be updated online, resulting in an algorithm that is more adaptable to various unexpected scenarios.

## Acknowledgements

This work was partially funded by the FCT projects CAPTURE (<https://doi.org/10.54499/PTDC/EEI-AUT/1732/2020>), CTS (UIDB/EEA/00066/2020), and LARSYS (UIDB/50009/2020).

## References

1. Alexis, K., Nikolakopoulos, G., Tzes, A.: Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications* **6**(12), 1812–1827 (2012). DOI 10.1049/iet-cta.2011.0348
2. Astrom, K., Hagglund, T.: *Pid controller: Theory, design, and tuning* (1995)
3. Coulson, J., Lygeros, J., Dörfler, F.: Data-enabled predictive control: In the shallows of the deepc. In: 2019 18th European Control Conference (ECC), pp. 307–312. IEEE (2019). DOI 10.23919/ECC.2019.8795639
4. Cowling, I.D., Yakimenko, O.A., Whidborne, J.F., Cooke, A.K.: A prototype of an autonomous controller for a quadrotor uav. In: 2007 European Control Conference (ECC), pp. 4001–4008. IEEE (2007). DOI 10.23919/ECC.2007.7068316
5. Elokda, E., Coulson, J., Beuchat, P.N., Lygeros, J., Dörfler, F.: Data-enabled predictive control for quadcopters. *International Journal of Robust and Nonlinear Control* **31**(18), 8916–8936 (2021). DOI 10.1002/rnc.5686
6. Hou, Z.S., Wang, Z.: From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences* **235**, 3–35 (2013). DOI 10.1016/j.ins.2012.07.014
7. Mohsan, S.A.H., Khan, M.A., Noor, F., Ullah, I., Alsharif, M.H.: Towards the unmanned aerial vehicles (uavs): A comprehensive review. *Drones* **6**(6), 147 (2022). DOI 10.3390/drones6060147
8. Ogata, K.: *Modern control engineering*. Prentice hall (2010)
9. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: Osqp: An operator splitting solver for quadratic programs. *Mathematical Programming Computation* **12**(4), 637–672 (2020). DOI 10.1007/s12532-020-00179-2
10. Willems, J.C., Rapisarda, P., Markovsky, I., De Moor, B.L.: A note on persistency of excitation. *Systems & Control Letters* **54**(4), 325–329 (2005). DOI 10.1016/j.sysconle.2004.09.003