

# DQ-NMPC: Dual-Quaternion NMPC for Quadrotor Flight

Luis F. Recalde<sup>1</sup>, Dhruv Agrawal<sup>1</sup>, Jon Arrizabalaga<sup>2</sup>, and Guanrui Li<sup>1</sup>

**Abstract**—MAVs have great potential to assist humans in complex tasks, with applications ranging from logistics to emergency response. Their agility makes them ideal for operations in complex and dynamic environments. However, achieving precise control in agile flights remains a significant challenge, particularly due to the underactuated nature of quadrotors and the strong coupling between their translational and rotational dynamics. In this work, we propose a novel NMPC framework based on dual-quaternions (DQ-NMPC) for quadrotor flight. By representing both quadrotor dynamics and the pose error directly on the dual-quaternion manifold, our approach enables a compact and globally non-singular formulation that captures the quadrotor coupled dynamics. We validate our approach through simulations and real-world experiments, demonstrating better numerical conditioning and significantly improved tracking performance, with reductions in position and orientation errors of up to 56.11% and 56.77%, compared to a conventional baseline NMPC method. Furthermore, our controller successfully handles aggressive trajectories, reaching maximum speeds up to 13.66 m/s and accelerations reaching 4.2 g within confined space conditions of dimensions 11m × 4.5m × 3.65m under which the baseline controller fails.

## SUPPLEMENTARY MATERIAL

Page: <https://acp-lab.github.io/dq-nmpc.github.io/>

## I. INTRODUCTION

Micro Aerial Vehicles (MAVs) have great potential in assisting humans in complex tasks such as environmental monitoring, logistics, and emergency response. Their agility and versatility make them ideal for operations in complex and dynamic environments. Nonlinear Model Predictive Control (NMPC) has been widely demonstrated to be effective in enabling agile maneuvers for MAVs [1], [2]. However, these approaches are often computationally intensive, especially compared to classical methods, such as geometric control [3]. Moreover, most existing NMPC frameworks treat translational and rotational dynamics separately, neglecting the inherent coupling between these states and potentially limiting performance in agile flight scenarios [4], [5]. These methods typically rely on a time-scale separation between rotational and translational dynamics. However, insufficient convergence in the rotational dynamics can severely impact the overall stability and performance of the system [6].

Researchers have also investigated coupled representations of rigid body dynamics in the Special Euclidean group  $SE(3)$  for MPC. MPC formulations based on error-state linearization in  $SE(3)$  have been proposed in [7], [8]. Although these

<sup>1</sup>The authors are with the Department of Robotics Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA. email: {lfrcalde, dagrawal, gli7}@wpi.edu.

<sup>2</sup>The author is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA. email: jarrizab@andrew.cmu.edu.

The authors acknowledge Amazon Robotics for their support on this paper.



Fig. 1. Trajectory tracking experiment comparing Dual-Quaternion NMPC and Baseline NMPC. Green lines indicate the desired trajectory, blue lines show the system's evolution under the DQ-NMPC controller, and orange lines represent the results obtained with the baseline NMPC.

approaches capture coupled dynamics, their reliance on local error approximations limits their validity to small deviations from a reference trajectory. In contrast, other methods operate directly on  $SE(3)$  without relying on linearization, often computing rotations using over-parameterized representations from the special orthogonal group  $SO(3)$ . Although this avoids approximation, it introduces a non-minimal state representation increasing computational complexity [9], [10].

In contrast, unit dual-quaternions offer a more compact and globally non-singular representation of  $SE(3)$  for rigid body dynamics, using only 8 parameters compared to the 12 required by  $SE(3)$  [11], [12]. Moreover, dual-quaternion algebra provides a unified framework to represent twists, wrenches, and various geometric primitives, which makes it suitable for robotic modeling and control tasks [13], [14]. Despite these advantages, relatively few studies have addressed control problems using unit dual-quaternions [14], highlighting the need for further exploration.

We propose an NMPC framework for quadrotor flight that employs dual quaternions (see Fig. 1) to achieve a unified and compact representation of the system dynamics. In addition, the proposed approach leverages a quadratic cost function in the Lie algebra of dual-quaternions that simultaneously encapsulates both rotation and translation errors.

In summary, our contributions are the following:

- We introduce a novel NMPC framework formulated directly on the dual-quaternion manifold, offering a unified and compact representation of quadrotor dynamics. The framework employs a quadratic cost function defined in the Lie algebra, leveraging a unified pose error that couples translation and rotation, enabling consistently scaled cost with respect to pose errors.
- We validate the proposed control method through various sets of simulation experiments, showing that the proposed method improves convergence of the optimization prob-

lem and enables faster convergence to the desired state compared to the baseline NMPC method.

- We test our method in real-world experiments, demonstrating that DQ-NMPC reduces position and orientation tracking errors by up to 56.11% and 56.77%, compared to the baseline for velocities of up to 7.5 m/s. In addition, our controller can track aggressive trajectories with maximum velocities up to 13.66 m/s and accelerations up to 4.2 g in confined space conditions  $11\text{m} \times 4.5\text{m} \times 3.65\text{m}$ , where the baseline fails.

This work is organized as follows: Section II introduces the dual-quaternion preliminaries and system dynamics modeling using dual-quaternions. Section III presents the proposed DQ-NMPC. Section IV reports the results of the simulation and real-world experiments, and Section V presents the conclusions and future research directions.

## II. SYSTEM DYNAMICS

In this section, we introduce the compact representation of the system dynamics using dual-quaternions. We first present brief mathematical preliminaries on dual quaternions, followed by a formulation of quadrotor dynamics using dual quaternions, which enables a unified pose representation.

### A. Mathematical Preliminaries

This section introduces dual-quaternion algebra, briefly revisiting the fundamental concepts of quaternions and dual numbers as prerequisites. Additional details on these topics can be found in [15], [14], [16].

1) *Quaternions*: Quaternions are an extension of complex numbers with a compact set defined as:

$$\mathbb{H} \triangleq \{q_1 + \hat{i}q_2 + \hat{j}q_3 + \hat{k}q_4 : q_1, q_2, q_3, q_4 \in \mathbb{R}\},$$

where the imaginary units  $\hat{i}$ ,  $\hat{j}$ , and  $\hat{k}$  have the following properties  $\hat{i}^2 = \hat{j}^2 = \hat{k}^2 = \hat{i}\hat{j}\hat{k} = -1$ . A quaternion can be expressed as  $\mathbf{q} = \text{Re}(\mathbf{q}) + \text{Im}(\mathbf{q})$ , where  $\text{Re}(\mathbf{q})$  represents the real part, and  $\text{Im}(\mathbf{q})$  is the imaginary part that contains  $\{\hat{i}, \hat{j}, \hat{k}\}$ . The norm of quaternions can be formulated as  $\|\mathbf{q}\| = \sqrt{\mathbf{q} \otimes \mathbf{q}^*}$ , where  $\mathbf{q}^* = \text{Re}(\mathbf{q}) - \text{Im}(\mathbf{q})$  is the conjugate of  $\mathbf{q}$  and  $\otimes$  is the quaternion multiplication operator.

We can represent three-dimensional vectors as a subset of quaternions containing only imaginary components, known as *pure quaternions*. This subset is defined as

$$\mathbb{H}_p \triangleq \{\mathbf{q} \in \mathbb{H} : \text{Re}(\mathbf{q}) = 0\}.$$

We can also represent a rigid body's orientation by *unit quaternions*, whose norm is 1, defined as

$$\mathbb{S}^3 \triangleq \{\mathbf{q} \in \mathbb{H} : \|\mathbf{q}\| = 1\}.$$

2) *Dual Numbers*: To define dual-quaternions, it is essential to first introduce a new algebraic unit known as the dual unit  $\varepsilon$ . The algebra generated by  $\varepsilon$  was first introduced by Clifford [17] and is characterized by the properties  $\varepsilon \neq 0$  and  $\varepsilon^2 = 0$ . A dual number can be expressed as  $\hat{\mathbf{a}} = \mathbf{a} + \varepsilon\mathbf{b}$ , where  $\mathbf{a}$  is the primary part and  $\mathbf{b}$  is the dual part. These components can be extracted using the operators  $\mathcal{P}(\hat{\mathbf{a}})$  and  $\mathcal{D}(\hat{\mathbf{a}})$ , respectively, such that  $\hat{\mathbf{a}} = \mathcal{P}(\hat{\mathbf{a}}) + \varepsilon\mathcal{D}(\hat{\mathbf{a}})$ .

3) *Dual quaternions*: Dual-quaternions are dual numbers in which both the primary and dual parts are quaternions. We can formulate the set of dual-quaternions as follows

$$\mathcal{H} \triangleq \{\mathbf{q}_p + \varepsilon\mathbf{q}_d : (\mathbf{q}_p, \mathbf{q}_d) \in \mathbb{H}, \varepsilon^2 = 0, \varepsilon \neq 0\},$$

thus, given the dual-quaternion  $\hat{\mathbf{q}} = a + \hat{i}b + \hat{j}c + \hat{k}d + \varepsilon(e + \hat{i}f + \hat{j}g + \hat{k}h)$ , we can establish the following operations

$$\mathcal{P}(\hat{\mathbf{q}}) = a + \hat{i}b + \hat{j}c + \hat{k}d \quad \mathcal{D}(\hat{\mathbf{q}}) = e + \hat{i}f + \hat{j}g + \hat{k}h.$$

$$\text{Re}(\hat{\mathbf{q}}) = a + \varepsilon(e) \quad \text{Im}(\hat{\mathbf{q}}) = \hat{i}b + \hat{j}c + \hat{k}d + \varepsilon(\hat{i}f + \hat{j}g + \hat{k}h)$$

We can define the full conjugate of dual-quaternions as  $\hat{\mathbf{q}}^* = \text{Re}(\hat{\mathbf{q}}) - \text{Im}(\hat{\mathbf{q}})$  and the norm as  $\|\hat{\mathbf{q}}\| = \sqrt{\hat{\mathbf{q}} \boxtimes \hat{\mathbf{q}}^*}$ , where  $\boxtimes$  is the dual-quaternion multiplication operator. With the definition of the norm of dual quaternion, we can further define the unit dual quaternions as follows,

$$\mathcal{S} \triangleq \{\hat{\mathbf{q}} \in \mathcal{H} : \|\hat{\mathbf{q}}\| = 1\}.$$

which we will leverage later in Section II-C to represent the pose of the quadrotor in a compact fashion. In addition, we further introduce *pure dual-quaternions*, defined as

$$\mathcal{H}_p \triangleq \{\hat{\mathbf{q}} \in \mathcal{H} : \text{Re}(\hat{\mathbf{q}}) = 0\},$$

which we will use to represent the twist and wrench of a rigid body in Section II-C.

### B. Classical Quadrotor Dynamics

We define the inertial frame as  $\mathcal{W} = \{\mathbf{e}_x^{\mathcal{W}}, \mathbf{e}_y^{\mathcal{W}}, \mathbf{e}_z^{\mathcal{W}}\}$ , and the body frame as  $\mathcal{B} = \{\mathbf{e}_x^{\mathcal{B}}, \mathbf{e}_y^{\mathcal{B}}, \mathbf{e}_z^{\mathcal{B}}\}$ , which is attached to the quadrotor's center of mass (CoM). The unit quaternion  $\mathbf{r} \in \mathbb{S}^3$  represents the orientation of the body frame  $\mathcal{B}$  with respect to the inertial frame  $\mathcal{W}$ , as shown in Fig. 2.

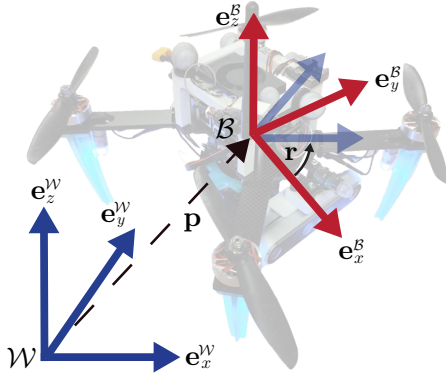
The classical quadrotor dynamics is decoupled into translation and rotation motion, which is widely used in research literature [1], as shown in the following:

$$\dot{\mathbf{p}} = \mathbf{v}, \dot{\mathbf{r}} = \frac{1}{2}\mathbf{r} \otimes \boldsymbol{\omega} \quad (1)$$

$$\dot{\mathbf{v}} = \frac{f}{m}\mathbf{r} \otimes \mathbf{e}_z^{\mathcal{W}} \otimes \mathbf{r}^* - \mathbf{g}, \quad (2)$$

$$\dot{\boldsymbol{\omega}} = \mathbf{J}^{-1}(\boldsymbol{\tau} - \boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})), \quad (3)$$

where  $(\mathbf{p}, \mathbf{v}) \in \mathbb{R}^3$  denote the position and velocity of the quadrotor with respect to the inertial frame  $\mathcal{W}$ ,  $\boldsymbol{\omega} \in \mathbb{R}^3$  is the quadrotor's angular velocity in the body frame  $\mathcal{B}$ .  $\mathbf{g} = g\mathbf{e}_z^{\mathcal{W}}$  is the gravity vector, where  $g = 9.81 \text{ m/s}^2$  and  $\mathbf{e}_z^{\mathcal{W}} = [0 \ 0 \ 1]^T$ . The inertia matrix of the aerial vehicle is given by  $\mathbf{J} = \text{diag}(J_{xx}, J_{yy}, J_{zz}) \in \mathbb{R}^{3 \times 3}$ , and its mass is denoted by  $m \in \mathbb{R}_{>0}$ . Finally, the control inputs consist of the thrust force  $f \in \mathbb{R}_{>0}$  and the torque vector  $\boldsymbol{\tau} \in \mathbb{R}^3$  expressed in the body frame  $\mathcal{B}$ .


 Fig. 2. Aerial system representation with inertial  $\mathcal{W}$  and body frames  $\mathcal{B}$ 

### C. Quadrotor Dynamics Based on Dual-Quaternions

In this section, we present the quadrotor dynamics modeling using dual-quaternions. First of all, a unit dual-quaternion  $\hat{\mathbf{q}} \in \mathcal{S}$  can represent a rigid-body transformation consisting of a translation  $\mathbf{p} \in \mathbb{H}_p$  followed by a rotation  $\mathbf{r} \in \mathbb{S}^3$ , shown as follows:

$$\hat{\mathbf{q}} = \mathbf{r} + \frac{1}{2}\varepsilon(\mathbf{p} \otimes \mathbf{r}), \quad (4)$$

where the position  $\mathbf{p}$  can be explicitly recovered through  $\mathbf{p} = 2\mathcal{D}(\hat{\mathbf{q}}) \otimes \mathcal{P}(\hat{\mathbf{q}})^*$ . Taking the time derivative of Eq. (4) yields the kinematics of the pose:

$$\dot{\hat{\mathbf{q}}} = \frac{1}{2}\hat{\mathbf{q}} \boxtimes \hat{\boldsymbol{\omega}}, \quad (5)$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{\omega} + \varepsilon(\mathbf{r}^* \otimes \mathbf{v} \otimes \mathbf{r}), \quad (6)$$

where  $\hat{\boldsymbol{\omega}} \in \mathcal{H}_p$  denotes the dual twist expressed in the body frame  $\mathcal{B}$ ,  $(\boldsymbol{\omega}, \mathbf{v}) \in \mathbb{H}_p$  represents the angular and linear velocities as pure quaternions. To express the quadrotor's dynamics using dual-quaternions, we take the time derivative of Equation (6), yielding the following expression:

$$\dot{\hat{\boldsymbol{\omega}}} = \hat{\boldsymbol{\omega}} + \varepsilon((\text{ad}_{(\mathbf{r}^*)}\dot{\mathbf{p}}) \times \boldsymbol{\omega} + \mathbf{r}^* \otimes \dot{\mathbf{v}} \otimes \mathbf{r}), \quad (7)$$

where  $\text{ad}_{(\mathbf{r}^*)}\dot{\mathbf{p}} = \mathbf{r}^* \otimes \dot{\mathbf{p}} \otimes \mathbf{r}$  is the adjoint transformation and the cross product operation  $\times$ : between pure quaternions is expressed as follows:

$$\text{ad}_{(\mathbf{r}^*)}\dot{\mathbf{p}} \times \boldsymbol{\omega} = \frac{\text{ad}_{(\mathbf{r}^*)}\dot{\mathbf{p}} \otimes \boldsymbol{\omega} - \boldsymbol{\omega} \otimes \text{ad}_{(\mathbf{r}^*)}\dot{\mathbf{p}}}{2}.$$

By substituting (2) and (3) into (7), we have

$$\dot{\hat{\boldsymbol{\omega}}} = \hat{\mathbf{f}} + \hat{\mathbf{u}}, \quad (8)$$

where

$$\hat{\mathbf{f}} = \mathbf{a} + \varepsilon((\text{ad}_{(\mathbf{r}^*)}\dot{\mathbf{p}}) \times \boldsymbol{\omega} - (\mathbf{r}^* \otimes \mathbf{g} \otimes \mathbf{r})),$$

$$\hat{\mathbf{u}} = \mathbf{J}^{-1}\boldsymbol{\tau} + \varepsilon\left(\frac{f}{m}\mathbf{e}_z^{\mathcal{W}}\right), \quad \mathbf{a} = -\mathbf{J}^{-1}\boldsymbol{\omega} \times \mathbf{J}\boldsymbol{\omega}.$$

Hence, based on eqs. (5) and (8), the quadrotor's dynamics can be compactly written using dual-quaternions as follows:

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}_d(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \begin{bmatrix} \frac{1}{2}\hat{\mathbf{q}} \boxtimes \hat{\boldsymbol{\omega}} \\ \hat{\mathbf{f}} + \hat{\mathbf{u}} \end{bmatrix}, \quad (10)$$

where  $\hat{\mathbf{x}} = [\hat{\mathbf{q}} \ \hat{\boldsymbol{\omega}}]^T \in \mathcal{S} \times \mathcal{H}_p$  denotes the internal state, comprising the unit dual-quaternion  $\hat{\mathbf{q}}$  and the dual twist  $\hat{\boldsymbol{\omega}}$ ; the control input is represented by  $\hat{\mathbf{u}} \in \mathcal{H}_p$ . As shown in eq. (10), we can represent the system's pose and twist in a unified and compact form by leveraging dual-quaternions.

### III. DQ-NMPC

This section introduces **Dual-Quaternion Nonlinear Model Predictive Control (DQ-NMPC)** method, an MPC framework developed for quadrotor flight based on dual-quaternions, as illustrated in Fig. 3. The approach defines the evolution of quadrotor dynamics on a dual-quaternion manifold, which unifies translational and rotational dynamics by leveraging unit dual-quaternions and dual twists. The cost function of the proposed **DQ-NMPC** is designed to jointly penalize translational and rotational errors in a unified manner, using the left-invariant pose error and its projection onto the Lie algebra of dual-quaternions.

#### A. Dual-Quaternion NMPC

NMPC formulations aim to compute a sequence of system states and control actions that optimize an objective function over a fixed time horizon, while explicitly incorporating the system dynamics and constraints into the optimal control problem. A general formulation of an NMPC can be written as follows:

$$\begin{aligned} \arg \min_{\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_N, \hat{\mathbf{u}}_1, \dots, \hat{\mathbf{u}}_{N-1}} \quad & l_N(\hat{\mathbf{x}}_N) + \sum_{k=1}^{N-1} l(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k) \\ \text{subject to:} \quad & \hat{\mathbf{x}}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k), \forall k = 1, \dots, N-1 \\ & \mathbf{g}(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k) \leq 0 \end{aligned} \quad (11)$$

where  $l_N(\hat{\mathbf{x}}_N)$  and  $l(\hat{\mathbf{x}}, \hat{\mathbf{u}})$  represent the terminal and running cost functions, respectively. The dynamics of the discrete nonlinear system is given by  $\hat{\mathbf{x}}_{k+1} = \mathbf{f}(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k)$ , and the state and input constraints are denoted by  $\mathbf{g}(\hat{\mathbf{x}}, \hat{\mathbf{u}})$ .

In the following, we present the detailed design of the proposed **DQ-NMPC** method for quadrotor flight. It includes 1) a cost function that simultaneously accounts for both translational and rotational errors by using a dual quaternion-based *left-invariant error* metric; 2) dynamics equations represented using unit dual-quaternions and dual twists, enabling a unified treatment of translation and orientation within the control scheme.

1) *Cost Function*: To enable a quadrotor to track the desired trajectory, we define the error between the current pose  $\hat{\mathbf{q}}$  and the desired pose  $\hat{\mathbf{q}}_d$  using the *left-invariant error* as following:

$$\hat{\mathbf{q}}_e = \hat{\mathbf{q}}_d^* \boxtimes \hat{\mathbf{q}}, \quad (12)$$

Additionally, the dual twist error can be expressed as:

$$\hat{\boldsymbol{\omega}}_e = \hat{\boldsymbol{\omega}}_d - \hat{\boldsymbol{\omega}}, \quad (13)$$

where  $\hat{\boldsymbol{\omega}}_d \in \mathcal{H}_p$  represents the desired dual twist. Using the errors above, we can further design the DQ-NMPC running

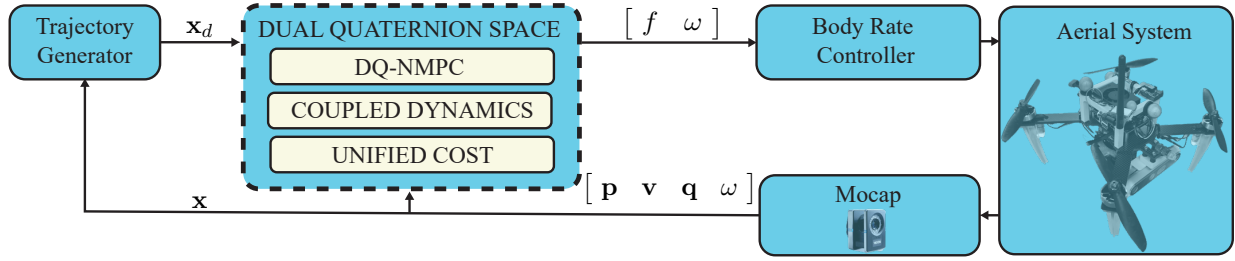


Fig. 3. Control block diagram of the proposed DQ-NMPC, illustrating the flow of information through the DQ-NMPC formulation.  $\mathbf{x}$  and  $\mathbf{x}_d$  represent the current and desired pose of the quadrotor based on decoupled dynamics. These values are transformed into the dual-quaternion space using eqs. (4) and (6) and subsequently used in the DQ-NMPC method.

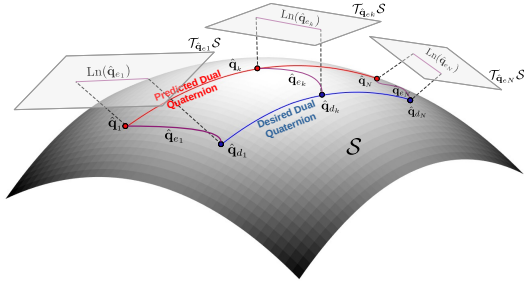


Fig. 4. Interpretation of the DQ-NMPC formulation: Both the predicted and desired poses of the quadrotor are represented as unit dual-quaternions, and their difference is captured using a left-invariant error. This error lies on the dual-quaternion manifold and is projected onto the Lie algebra and included into the NMPC cost function.

and terminal cost functions that jointly penalize both pose and twist errors over a finite horizon  $N$ :

$$l(\hat{\mathbf{q}}_{eN}, \hat{\boldsymbol{\omega}}_{eN}) + \sum_{k=1}^{N-1} l(\hat{\mathbf{q}}_{ek}, \hat{\boldsymbol{\omega}}_{ek}) + l_u(\hat{\mathbf{u}}_{ek}), \quad (14)$$

with

$$l(\hat{\mathbf{q}}_e) = \|\text{Ln}(\hat{\mathbf{q}}_e)\|_{\mathbf{Q}_p}^2 + \|\hat{\boldsymbol{\omega}}_e\|_{\mathbf{Q}_v}^2, \quad l_u(\hat{\mathbf{u}}_e) = \|\hat{\mathbf{u}}_e\|_{\mathbf{R}}^2. \quad (15)$$

The logarithmic map  $\text{Ln}(\hat{\mathbf{q}}) = \frac{1}{2}(\boldsymbol{\phi} + \boldsymbol{\varepsilon} \mathbf{t})$  extracts the minimal representation of pose error in the Lie algebra of dual-quaternions, where  $\boldsymbol{\phi}$  and  $\mathbf{t}$  represent the rotational and translational components.  $\mathbf{Q}_p$  and  $\mathbf{Q}_v$  are positive definite weighting matrices. Finally,  $l_u$  penalizes the error from desired control actions  $\hat{\mathbf{u}}_e = \hat{\mathbf{u}}_d - \hat{\mathbf{u}}$ . We choose this cost function inspired by [18], which showed that a proper left-invariant metric defined in the Lie algebra can ensure globally exponential convergence.

To provide further insights into the left-invariant error and its projection onto the Lie algebra used in our DQ-NMPC framework, we present a comparison of the orientation errors used by the DQ-NMPC and the baseline NMPC [1]. As shown in Fig. 5, the consistent scaling of the error metric in DQ-NMPC can result in improved trajectory tracking accuracy, particularly during aggressive maneuvers that induce significant rotational errors. On the other hand, the baseline cost function remains nearly flat in the presence of large orientation errors, indicating a lack of sensitivity. This results in insufficient corrective control actions, which negatively affect the stability and tracking performance of the system.

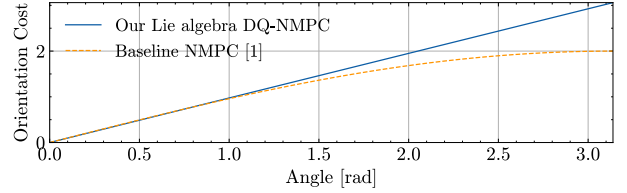


Fig. 5. Comparison of different orientation error cost functions. The baseline cost function lacks sensitivity to large orientation errors, which limits its performance and results in slower convergence

2) *System Dynamics*: We integrate both translational and rotational dynamics employing unit dual-quaternions and dual twists. Hence, we apply a discrete form of eq. (10), constructed with the fourth-order Runge-Kutta method [19], which enables efficient and singularity-free numerical integration of quaternions. This can be expressed as

$$\hat{\mathbf{x}}_{k+1} = \mathbf{f}_{RK4d}(\mathbf{f}_d(\hat{\mathbf{x}}_k, \hat{\mathbf{u}}_k), dt), \quad (16)$$

where  $dt$  denotes the sampling interval.

Fig. 4 provides a visual representation of the role of dual-quaternions within the NMPC framework.

3) *Constraints*: To address potential issues related to the primary part of unit dual-quaternions, we introduce a non-linear constraint to enforce its unit norm. In addition, control input constraints are imposed to ensure feasibility within pre-defined bounds. These conditions can be compactly expressed as:

$$\mathbf{g}_d(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \leq 0, \quad \mathbf{g}_d(\hat{\mathbf{x}}, \hat{\mathbf{u}}) := \begin{bmatrix} \|\mathcal{P}(\hat{\mathbf{q}})\| - 1 \\ 1 - \|\mathcal{P}(\hat{\mathbf{q}})\| \\ \hat{\mathbf{u}} - \hat{\mathbf{u}}_{max} \\ -\hat{\mathbf{u}} + \hat{\mathbf{u}}_{min} \end{bmatrix} \quad (17)$$

where  $\hat{\mathbf{u}}_{max}$  and  $\hat{\mathbf{u}}_{min}$  represent the maximum and minimum control actions.

#### IV. EXPERIMENTAL RESULTS

In this section, we present the experimental results of our proposed method in both simulation and real-world environments. We compared the proposed method with the recent state-of-the-art NMPC method in [1], which decoupled position and orientation in dynamics and cost functions. Both methods generate thrust and angular rate commands to the low-level angular rate controller.

In the following, we first present the simulation results, which demonstrate improved optimality, enhanced pose tracking performance, and robustness compared to the baseline

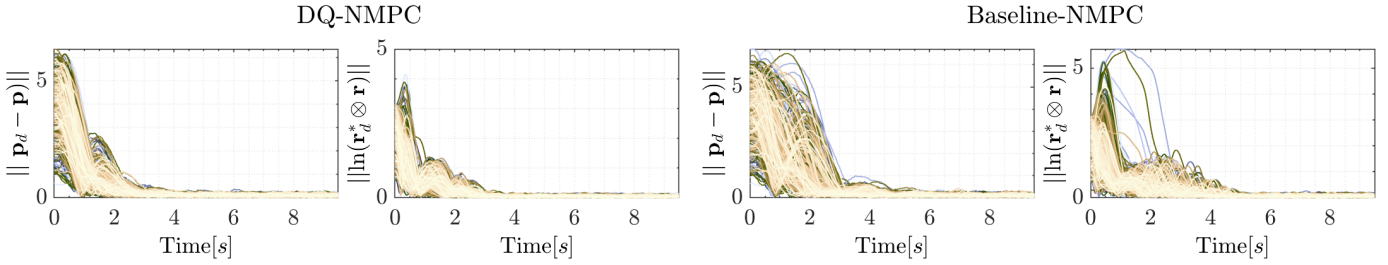


Fig. 6. The position and orientation error metrics of both NMPC methods are evaluated in simulations using 600 randomly sampled initial poses for pose regulation. The initial translation vectors are sampled within the range  $[-4 \ -4 \ 0]^T$  to  $[4 \ 4 \ 4]^T$ , and the initial orientations are randomly sampled such that  $\ln(\mathbf{r})$  is uniform in the interval  $[0, \pi]$ .

NMPC. Next, we present the results of the real world experiments, demonstrating that the proposed DQ-NMPC outperforms the baseline NMPC considering trajectory tracking tasks at velocities of up to 13.66 m/s and accelerations up to 4.2 g.

### A. Simulation Results

We conducted simulation experiments in two different scenarios, which led to a systematic analysis of the effectiveness of our proposed method.

- *Pose Regulation*: This scenario focuses on a regulation problem using a well-identified model. We analyze the controllers’ behavior in terms of pose error, optimality conditions, and convergence achieved by each control method.
- *Robustness Study*: To assess robustness, we introduce model mismatches and external disturbances during the simulation experiments. The performance of both controllers is evaluated across a range of trajectories under modeled mismatches and disturbances.

Our results show that the DQ-NMPC consistently achieves better convergence to the desired pose under different initial conditions. This formulation also yields better-conditioned solutions and requires fewer iterations within the optimization framework. Finally, it demonstrates improved performance under model mismatches compared to baseline.

1) *Pose Regulation*: This section presents results related to the pose regulation task. We report the outcomes for both controllers by examining position and orientation metrics and evaluating how closely the optimization solutions approach optimality. The desired attitude, represented as zero rotation quaternion and the desired position, represented as the origin are identical for both formulations. During these experiments, both NMPC methods were solved using Sequential Quadratic Programming (SQP), with HPIPM [20] handling the internal QP subproblems, and the Gauss-Newton approximation, all using tools available in the ACADOS framework [21]. A fixed prediction horizon of 1.5 s was used and the gain matrices were meticulously designed to ensure similar stabilization times.

**Control Performance** To quantitatively evaluate the behavior of both controllers, we use two primary metrics: the position error, defined as  $\|\mathbf{p}_d - \mathbf{p}\|$ , and the orientation error, given by  $\|\ln(\mathbf{r}_d^* \otimes \mathbf{r})\|$ . Performance metrics for both control

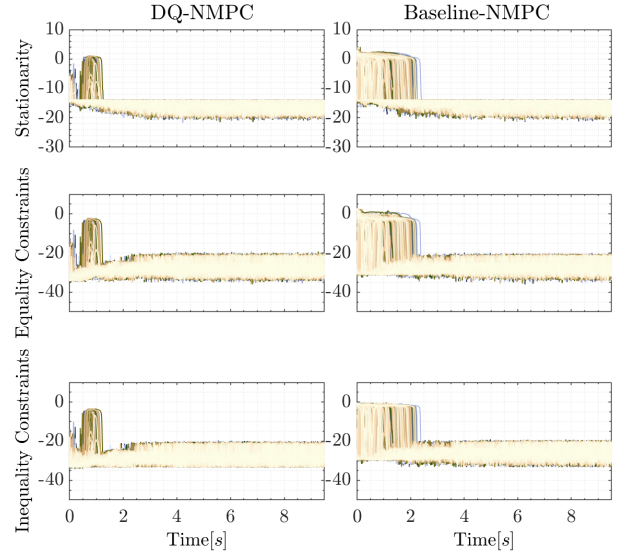


Fig. 7. KKT condition residuals obtained from 600 simulations for pose regulation. The left column shows the results for DQ-NMPC, while the right column presents those for the baseline.

formulations are presented in Fig. 6, based on an evaluation of 600 initial positions and orientations randomly sampled from a uniform distribution. We observe that the position and orientation errors are influenced by large initial pose error, they affect the convergence of the baseline NMPC. On the other hand, the DQ-NMPC is able to maintain convergence across all initial conditions.

**Optimality Conditions** To assess the optimality of the solutions of the optimal control problems, we analyze the Karush-Kuhn-Tucker (KKT) condition numbers. Fig. 7 shows that the DQ-NMPC consistently achieves better conditioned solutions near optimal, even in the presence of significant initial errors, commonly encountered at the beginning of the optimization process. This behavior improves the minimization of the cost function while satisfying both the equality and inequality constraints.

**Convergence Across QP solvers** We report the results for both methods in a pose regulation task, evaluating 20 randomly sampled poses with only large pose errors. Two QP solvers were considered, HPIPM [20] and DAQP [22], both commonly used in real-time applications. We present the mean and standard deviation of the number of iterations required by the SQP algorithm in Fig. 8. Despite the presence of large pose

TABLE I

PERFORMANCE METRICS: COMPARISON BETWEEN DQ-NMPC AND BASELINE-NMPC UNDER UNMODELED DYNAMICS AND EXTERNAL DISTURBANCES.

	Position RMSE [m] mean $\pm$ STD		Orientation RMSE [rad] mean $\pm$ STD		Rate of Change $\ \Delta \mathbf{u}_f\ , \ \Delta \mathbf{u}_\omega\ $	
	DQ-NMPC	Baseline-NMPC	DQ-NMPC	Baseline-NMPC	DQ-NMPC	Baseline-NMPC
Disturbance-free ideal model	0.039 $\pm$ 0.005	0.103 $\pm$ 0.039	0.083 $\pm$ 0.011	0.269 $\pm$ 0.121	25.95, 14.17	40.06, 13.26
+%78 Drag	0.136 $\pm$ 0.029	0.268 $\pm$ 0.106	0.182 $\pm$ 0.036	0.322 $\pm$ 0.082	16.92, 6.21	31.90, 11.12
+%20 Mass	0.132 $\pm$ 0.013	0.128 $\pm$ 0.026	0.088 $\pm$ 0.014	0.240 $\pm$ 0.075	29.32, 12.58	47.71, 12.16
+%20 Inertia Matrix	0.039 $\pm$ 0.005	0.081 $\pm$ 0.021	0.088 $\pm$ 0.014	0.239 $\pm$ 0.095	23.75, 18.16	42.39, 34.38
7.07 [N] External Force	0.279 $\pm$ 0.174	0.240 $\pm$ 0.119	0.226 $\pm$ 0.097	0.375 $\pm$ 0.126	30.53, 15.88	46.54, 17.58
0.02 [Nm] External Moment	0.152 $\pm$ 0.012	0.183 $\pm$ 0.080	0.157 $\pm$ 0.052	0.651 $\pm$ 0.560	26.37, 16.10	50.01, 21.56

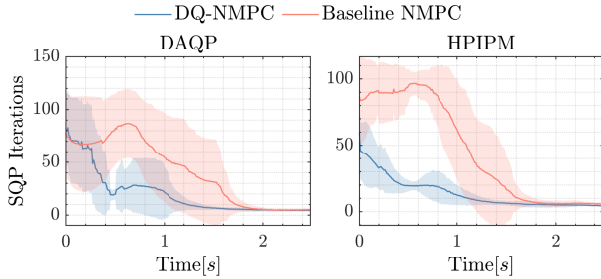


Fig. 8. SQP iterations in the optimization process for pose regulation task considering HPIPM and DAQP as internal QP solvers under large initial pose errors.

errors, the proposed DQ-NMPC consistently required fewer iterations to converge. We attribute this property to the unified modeling of the quadrotor dynamics on the dual-quaternion manifold and the use of a left-invariant pose error projected onto the Lie algebra of dual quaternions, which yields a well-scaled error representation and contributes to improved convergence of the SQP algorithm.

2) *Robustness Study*: Previous simulations were conducted under ideal conditions, assuming perfect knowledge of the model and the absence of disturbances. However, this is not the reality in practical experiments. Therefore, this section studies the performance of DQ-NMPC and Baseline-NMPC under the following scenarios:

- *Unmodeled Dynamics*: We include a model mismatch by considering drag, mass, and inertia coefficients in the quadrotor model.
- *External Disturbances* During the experiments, it is possible to introduce constant external forces, and torques, which can significantly decrease the performance of both controllers.

The simulations were conducted in our custom environment, with both NMPC methods executed using SQP under the Real-Time Iteration (RTI) scheme. The internal QP subproblems were solved using HPIPM [20], with a Gauss-Newton approximation, all implemented within the ACADOS framework [21]. These experiments are carried out considering feasible trajectories, using the circular 16 and Lissajous trajectories, with a maximum velocity of  $V_{max} = 4.68$  m/s.

Table I presents a comprehensive evaluation of the RMSE and the rate of change of control actions in various unmodeled

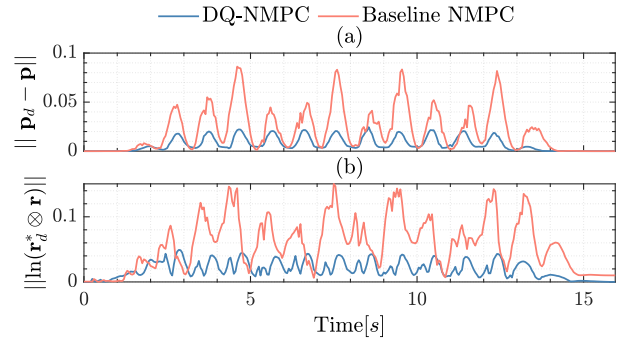


Fig. 9. Tracking performance of a Lissajous trajectory with a maximum speed of up to 4.67 m/s (Disturbance-free and ideal model). **a**: Position error metric. **b** Orientation error metric

dynamic conditions. The DQ-NMPC consistently outperforms the baseline in orientation tracking accuracy, achieving a significantly lower orientation RMSE in all scenarios. This advantage is particularly evident in experiments that involve inertia mismatches, external torques, and aerodynamic drag, where DQ-NMPC demonstrates superior performance. Furthermore, it produces smoother control inputs, which are critical for maintaining the stability of the system. Although the baseline NMPC achieves slightly lower position RMSE in certain situations, especially when the system is subject to mass variations and external forces, its performance decreases in orientation tracking and control smoothness. In general, the DQ-NMPC formulation provides more balanced behavior in a wide range of challenging conditions. Fig. 9 illustrates the tracking results obtained for one of the trajectories in the simulation robustness study including the control error metrics. It is evident that DQ-MPC outperforms the baseline, particularly in reducing orientation errors.

### B. Real-World Experiments

In this section, we present the real-world experiment results. The experiments are conducted in the Aerial-robot Control and Perception lab (ACP) flying space at Worcester Polytechnic Institute. We used a customized quadrotor platform equipped with Nvidia Jetson Orin NX and PixRacer Autopilot board. The quadrotor flies in a netted facility of dimension 11m  $\times$  4.5m  $\times$  3.65m equipped with Vicon motion capture system, which provides the quadrotor state feedback at 240 Hz.

TABLE II

PERFORMANCE METRICS IN REAL-WORLD EXPERIMENTS: POSITION AND ORIENTATION RMSE FOR DIFFERENT TRAJECTORIES AND VELOCITIES, ALONG WITH THE PERCENTAGE IMPROVEMENT OF DQ-NMPC OVER THE BASELINE.

Trajectory	Max Velocity [m/s]	Position RMSE [m]		Orientation RMSE [rad]		Error Reduction [%]	
		DQ-NMPC	Baseline	DQ-NMPC	Baseline	Position	Orientation
Loop	1.0	0.0135 ± 0.0031	0.0210 ± 0.0042	0.0115 ± 0.0057	0.0675 ± 0.0927	35.71	82.96
	1.6	0.0163 ± 0.0046	0.0246 ± 0.0058	0.0277 ± 0.0141	0.0768 ± 0.0439	33.73	63.80
	1.9	0.0146 ± 0.0054	0.0298 ± 0.0075	0.0271 ± 0.0196	0.1034 ± 0.0580	51.00	73.80
Lissajous	4.7	0.0326 ± 0.0187	0.0324 ± 0.0110	0.0465 ± 0.0262	0.0811 ± 0.0348	-0.61	42.66
	5.3	0.0384 ± 0.0238	0.0386 ± 0.0131	0.0506 ± 0.0316	0.0856 ± 0.0335	0.51	40.88
	6.1	0.0443 ± 0.0280	0.0581 ± 0.0347	0.0549 ± 0.0382	0.0864 ± 0.0478	23.74	36.45
	7.5	0.0577 ± 0.0413	0.1314 ± 0.0987	0.0568 ± 0.0432	0.1314 ± 0.0670	56.11	56.77

Both the proposed DQ-NMPC method and the baseline NMPC method are implemented in ROS2 using C++ on a Nvidia Jetson Orin NX at 190 Hz. The optimization problem was solved in ACADOS [21] using SQP with RTI scheme, with HPIPM [20] handling the internal QP subproblems and a Gauss–Newton Hessian approximation over a fixed prediction horizon of 1.5 s. To ensure a fair comparison, the weighting matrices for both controllers were tuned experimentally to achieve comparable performance under low-velocity conditions 4.7 m/s. These parameters were kept fixed across all experiments and were not adjusted based on the specific trajectory or velocities.

Experimental evaluation involves trajectory tracking experiments with circular and Lissajous trajectories, achieving velocities of up to 7.5 m/s. Table II clearly highlights the performance of the proposed DQ-NMPC. DQ-NMPC demonstrates a substantial reduction in orientation tracking errors, achieving improvements of up to 56.77% relative to the baseline formulation considering velocities up to 7.5 m/s. Furthermore, it consistently maintains a lower position RMSE across the evaluated range of velocities. In particular, the performance gap widens significantly at higher velocities, where the baseline formulation exhibits a noticeable degradation in orientation tracking accuracy. These findings underscore the effectiveness of the DQ-NMPC, validating its advantages for high-speed scenarios.

Fig. 10 presents a comparison of boxplots for DQ-NMPC and the baseline NMPC controller when tracking Lissajous trajectories at varying reference maximum velocities. In terms of position RMSE, both methods exhibit similar performance, with DQ-NMPC showing a slightly lower average error. However, in orientation tracking, DQ-NMPC clearly outperforms the baseline. Notably, as the reference velocity increases, the baseline controller’s performance degrades significantly, while DQ-NMPC maintains consistent accuracy.

We attribute the tracking performance of the proposed DQ-NMPC to the left-invariant pose error on the dual-quaternion group, which includes orientation and translation in the desired frame. Additionally, since the pose error belongs to a dual-quaternion group, the error is projected onto its Lie algebra, enabling the design of a quadratic cost function that provides a minimal and symmetric representation. Unlike conventional metrics that lose sensitivity under large orientation deviations [1], [6], our cost maintains sensitivity throughout large

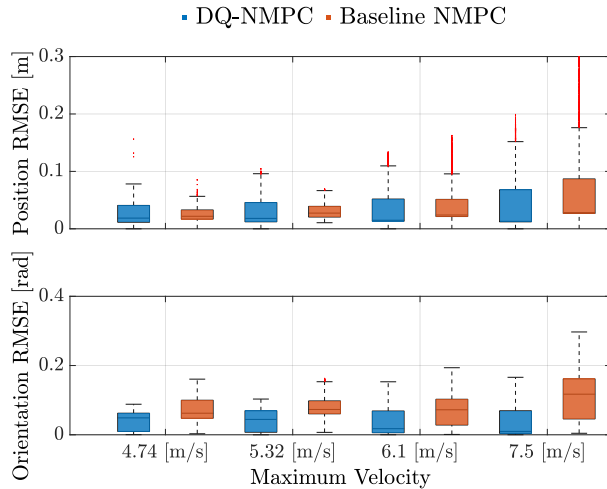


Fig. 10. Box plot of tracking error for position and orientation considering a Lissajous in real-world experiments.

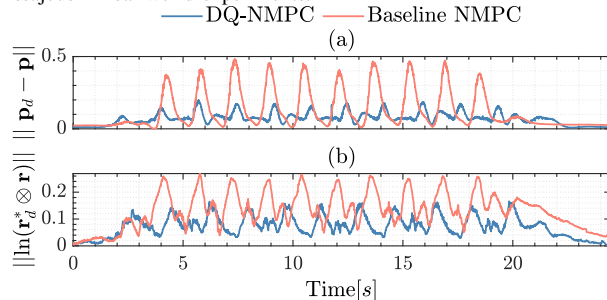


Fig. 11. Tracking performance of a Lissajous trajectory with a maximum speed of up to 7.5 m/s. a: Position error metric. b: Orientation error metric.

orientation errors, ensuring fast convergence of rotational dynamics and accurate trajectory tracking.

The corresponding performance metrics, including position and orientation errors, are shown in Fig. 11a and 11b, respectively. The results demonstrate that the DQ-NMPC controller achieves superior performance while maintaining consistent pose tracking accuracy throughout the experiment.

Finally, we push the limits of the proposed DQ-NMPC framework by conducting experiments with Lissajous trajectories at speeds of up to 13.66 m/s and accelerations up to 4.2 g. Snapshots of the quadrotor executing this high-speed trajectory are shown in Fig. 12a. The corresponding performance metrics are presented in Fig. 12b and Fig. 12c, respectively. The velocity and acceleration profiles achieved by the quadrotor

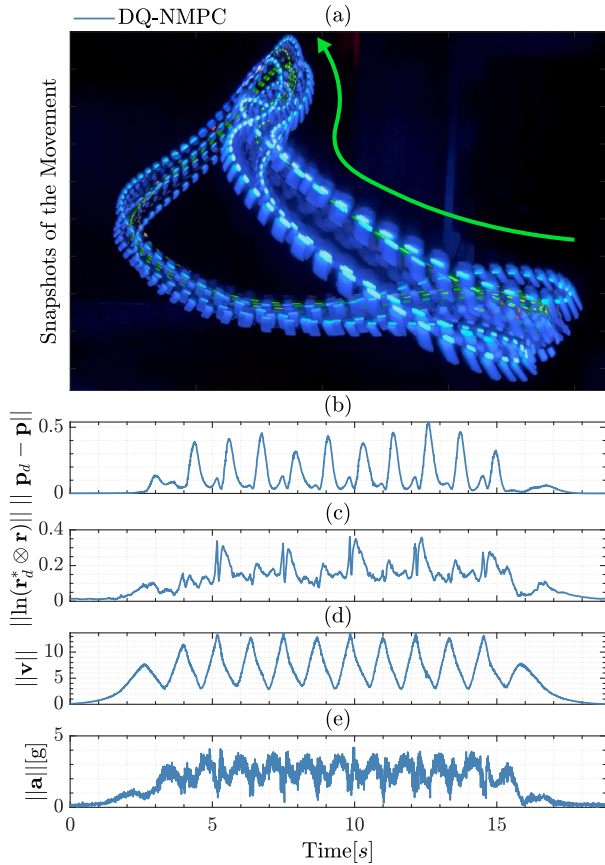


Fig. 12. Tracking performance of a Lissajous trajectory with a maximum speed of up to 13.66 m/s and accelerations up to 4.2 g. **a:** Movement of the quadrotor using the DQ-NMPC. **b:** Position error metric. **c:** Orientation error metric. **d:** Magnitude of the linear velocity during the experiment. **e:** Magnitude of the accelerations during the experiment.

during the experiment are illustrated in Fig. 12d and Fig. 12e, respectively. No baseline NMPC results are reported for this scenario, as the baseline controller was unable to compute a feasible solution at such speeds.

## V. CONCLUSION

This work introduces the DQ-NMPC framework for quadrotor flight. By leveraging unit dual-quaternions, the proposed approach provides a unified and compact representation of quadrotor dynamics. The proposed dual-quaternion-based cost function also simultaneously penalizes both rotation and translation errors. Extensive simulations and real-world experiments demonstrate the effectiveness of the proposed method, showing an improved convergence of the optimization problem and enhanced tracking performance compared to a conventional baseline NMPC method. Future work will also explore the application of the DQ-NMPC framework to other robotic platforms, such as aerial manipulators, fully actuated multi-rotor vehicles, and legged robots, as well as its integration with learning-based methods to enhance adaptability and performance in dynamic and uncertain environments.

## REFERENCES

- [1] S. Sun, A. Romero, P. Foehn, E. Kaufmann, and D. Scaramuzza, "A comparative study of nonlinear mpc and differential-flatness-based control for quadrotor agile flight," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3357–3373, 2022.
- [2] L. F. Recalde, B. S. Guevara, C. P. Carvajal, V. H. Andaluz, J. Varela-Aldás, and D. C. Gandolfo, "System identification and nonlinear model predictive control with collision avoidance applied in hexacopters uavs," *Sensors*, vol. 22, no. 13, 2022.
- [3] E. Tal and S. Karaman, "Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1203–1218, 2021.
- [4] T. Salzmann, E. Kaufmann, J. Arrizabalaga, M. Pavone, D. Scaramuzza, and M. Ryll, "Real-time neural mpc: Deep learning model predictive control for quadrotors and agile robotic platforms," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2397–2404, 2023.
- [5] F. Nan, S. Sun, P. Foehn, and D. Scaramuzza, "Nonlinear mpc for quadrotor fault-tolerant control," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5047–5054, 2022.
- [6] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on  $se(3)$ ," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 5420–5425.
- [7] G. Lu, W. Xu, and F. Zhang, "On-manifold model predictive control for trajectory tracking on robotic systems," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 9, pp. 9192–9202, 2023.
- [8] S. Teng, D. Chen, W. Clark, and M. Ghaffari, "An error-state model predictive control on connected matrix lie groups for legged robot control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 8850–8857.
- [9] J. C. Pereira, V. J. S. Leite, and G. V. Raffo, "Nonlinear model predictive control on  $se(3)$  for quadrotor aggressive maneuvers," *Journal of Intelligent & Robotic Systems*, vol. 101, no. 1, p. 62, 2021.
- [10] M. Kamel, K. Alexis, M. Achteik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on  $so(3)$ ," in *2015 IEEE Conference on Control Applications (CCA)*, 2015, pp. 1160–1166.
- [11] N. Aspragathos and J. Dimitros, "A comparative study of three methods for robot kinematics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 2, pp. 135–145, 1998.
- [12] N. Filipe and P. Tsiotras, "Rigid body motion tracking without linear and angular velocity feedback using dual quaternions," in *2013 European Control Conference (ECC)*, 2013, pp. 329–334.
- [13] L. F. da Cruz Figueredo, B. V. Adorno, and J. Y. Ishihara, "Robust h kinematic control of manipulator robots using dual quaternion algebra," *Automatica*, vol. 132, p. 109817, 2021.
- [14] J. Arrizabalaga and M. Ryll, "Pose-following with dual quaternions," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 5959–5966.
- [15] M. M. Marinho, B. V. Adorno, K. Harada, and M. Mitsuishi, "Dynamic active constraints for surgical robots using vector-field inequalities," *IEEE Transactions on Robotics*, vol. 35, no. 5, pp. 1166–1185, 2019.
- [16] X. Wang and C. Yu, "Unit dual quaternion-based feedback linearization tracking problem for attitude and position dynamics," *Systems and Control Letters*, vol. 62, no. 3, pp. 225–233, 2013.
- [17] Clifford, "Preliminary sketch of biquaternions," *Proceedings of the London Mathematical Society*, vol. s1-4, no. 1, pp. 381–395, 1871.
- [18] S. Teng, W. Clark, A. Bloch, R. Vasudevan, and M. Ghaffari, "Lie algebraic cost function design for control on lie groups," in *IEEE Conference on Decision and Control (CDC)*, 2022, pp. 1867–1874.
- [19] C. Rucker, "Integrating rotations using nonunit quaternions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2979–2986, 2018.
- [20] G. Frison and M. Diehl, "HPIPM: a high-performance quadratic programming framework for model predictive control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6563–6569, 2020.
- [21] R. Verschuere, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.
- [22] D. Arnström, A. Bemporad, and D. Axehill, "A dual active-set solver for embedded quadratic programming using recursive  $l_1^T$  updates," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4362–4369, 2022.