Nonlinear predictive control application on DENSO-VP6242 manipulator using dual quaternions

Arthur da C. Vangasse * **Andressa M. Oliveira** *
Allan de M. Martins *** **Ícaro B. Q. Araújo** * **Heitor J. Savino** **
Glauber R. Leite *

* *Instituto de Computação, Universidade Federal de Alagoas, AL, (e-mail: { vangasse, andressamartins, icaro, grl }@ic.ufal.br).*
** *Cervejaria Ambev, SP, (e-mail: heitor.savino@ambev.com.br).*
*** *Departamento de Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, RN, (e-mail: allan@dee.ufrn.br).*

Abstract: This work presents an application of non-linear model predictive control (NMPC), combined to a kinematic model based in dual quaternion of a comercial robotic manipulator. The 6—DOF serial manipulator, VP-6242 by Denso Robotics was used as reference. Having developed the system's explicit model, optimal control actions are computed in a prediction horizon according to an objective function to be minimized. The optimization problem, considers the structure constraints determined by the robotic system, guaranteeing physical integrity. Dual quaternion algebra adds robustness to the model regarding singularities and computational advantages when compared to traditional homogeneous transformation matrices methods. For NMPC, is possible to deal with multiple variable problems of non-linear nature, which solution is provided by MATLAB MPC Toolbox. Simulation results are presented, showing that the implemented algorithm is capable of generating safe and smooth routes to the highly complex system.

Keywords: Model Predictive Control; Dual Quaternions; Robots manipulators; MIMO Control Systems; Distributed control.

1. INTRODUCTION

It becomes noticeable that the emergent highly complex tasks must resort to more complex robotic systems. Those on the other hand, to become capable of operating in highly dynamic environments, dexterously manipulate inappropriate objects or even interact with humans, shall rely on powerful models and controllers (Adorno and Marinho, 2020).

Topics on kinematic modelling are commonly addressed in robotics textbooks at the starting chapters. Traditionally, rigid motions are decoupled, with rotations being represented by rotation matrices, whereas translations, represented by position vectors. Following through their combination leading to the homogeneous transformation matrices. Such form of exposition can be seen in Siciliano et al. (2009) and Craig (2009).

Adorno (2011) exposes a point of view on kinematic modeling and control based on dual quaternions. The approach puts the choice of dual quaternions over the homogeneous transformation matrices as very well justified, once given that the dual quaternions provide unified representation for robot modeling and control and furthermore, are more compact than the homogeneous transformation matrices. On the matter, Abaunza et al. (2017) also introduces a methodology for the use of dual quaternions on kinematic and dynamic modeling, demonstrated over a quad-rotor with an attached manipulator.

Model predictive control (MPC), on the other hand, is a methodology used for the design of optimal control systems, which assumes a explicit model of a system, optimizing control actions based on predicted outputs, usually associated to constrained multivariable problems (Camacho and Bordons, 2007). In a recent work, Eckhoff et al. (2022) produces contributions on safe human-robot interaction strategies using a MPC framework, which considers constraints based on behaviors that are known to trigger human involuntary motions. Although, due to the expressive nonlinearities of robotic manipulators, the linear models required by conventional MPC, become incapable of efficiently covering the system's behaviour. Elsisi et al. (2021) proposes the use of nonlinear model predictive control (NMPC) for robotic manipulators combined with a neural network algorithm for parameter tuning.

The contribution of this paper combines both overstated approaches in regard of the robotic manipulator Denso VP-6242 6-axis robotic manipulator, which offers a payload of 2.0 kg and a reach of 432 mm, Figure 1 shows the robotic arm located at Instituto de Computação, Universidade Federal de Alagoas. First, the kinematic model based on dual quaternion algebra for Denso VP-6242 was developed. Then, the NMPC problem was written, being aware

ISSN: 2525-8311

DOI: 10.20906/CBA2022/3675

Figure 1. Photography of Denso-VP6242, located at Instituto de Computação, Universidade Federal de Alagoas.

of the designed model and physical constraints imposed by the manipulator's own structure.

This paper is organized as follows: Section 2 expresses the model based in dual quaternion algebra; Section 3 explores how the NMPC problem was built; Section 4 shows the architecture and the developed simulation scenarios; Section 5 presents the given results. Concluding remarks and future work are summarized at the end.

2. DUAL QUATERNION KINEMATICS

Alternatively to homogeneous transformation matrices, Adorno (2011) provides a detailed mathematical background for the dual quaternion algebra, adverting to the desired property of explaining more using fewer and simpler arguments. That approach has the potential to be used in robot modelling and control.

Quaternions algebra was introduced by Hamilton (1844) as an extension for complex numbers, where three imaginary units $\hat{\imath}, \hat{\jmath}, \hat{k}$ are defined with the properties:

$$\hat{\imath}^2 = \hat{\jmath}^2 = \hat{k}^2 = \hat{\imath}\hat{\jmath}\hat{k} = -1. \tag{1}$$

Therefore, the quaternion set is formally defined as:

$$\mathbb{H} \triangleq \{h_1 + \hat{\imath}h_2 + \hat{\jmath}h_3 + \hat{k}h_4 : h_1, h_2, h_3, h_4 \in \mathbb{R}\}. \tag{2}$$

Clifford (1871), on the other hand, introduced the dual number ε with properties:

$$\varepsilon \neq 0, \qquad \varepsilon^2 = 0. \tag{3}$$

And the dual quaternions in the same opportunity, by the set formally defined as:

$$\mathcal{H} \triangleq \{h_1 + \varepsilon h_2 : h_1, h_2 \in \mathbb{H}, \varepsilon \neq 0, \varepsilon^2 = 0\}. \tag{4}$$

With that, the dual quaternion can be written as follows for the representation of rigid motions:

$$\underline{\boldsymbol{h}} = \boldsymbol{r} + \varepsilon\frac{1}{2}\boldsymbol{p}\boldsymbol{r}, \tag{5}$$

where $\boldsymbol{r}$ is an unit norm quaternion representing a rotation, and $\frac{1}{2}\boldsymbol{p}\boldsymbol{r}$ an unbounded norm quaternion indirectly representing a translation.

2.1 Kinematic Model

The pose assigned to the robot's effector is given by its relationship with the configuration in the joint space. For the case of serial manipulators exclusively composed by rotational joints and n links, the transformation $\underline{\boldsymbol{x}}_i^{i-1}$ establishes the pose of link i based on the pose of the previous link of the chain. Therefore, the effector's pose $\underline{\boldsymbol{x}}_E$ is described by the composition of transformations along the chain:

$$\underline{\boldsymbol{x}}_E = \underline{\boldsymbol{x}}_1^0\underline{\boldsymbol{x}}_2^1 \cdots \underline{\boldsymbol{x}}_n^{n-1}. \tag{6}$$

Each intermediate transformation was modeled by using the Denavit-Hartenberg convention in dual quaternion space, which is straightforward and consists in multiplying the four transformations:

$$\underline{\boldsymbol{x}}_{DH} = \boldsymbol{r}_{z,\theta}\underline{\boldsymbol{p}}_{z,d}\underline{\boldsymbol{p}}_{x,a}\boldsymbol{r}_{x,\alpha}. \tag{7}$$

where $\boldsymbol{r}_{z,\theta} = \cos(\theta/2) + \hat{k}\sin(\theta/2)$ representing a θ rotation on z-axis, $\boldsymbol{r}_{x,\alpha} = \cos(\alpha/2) + \hat{\imath}\sin(\alpha/2)$ representing a α rotation on x-axis, $\underline{\boldsymbol{p}}_{z,d} = 1 + \varepsilon(d/2)\hat{k}$ representing a d translation along z-axis, and lastly $\underline{\boldsymbol{p}}_{x,a} = 1 + \varepsilon(a/2)\hat{\imath}$ a a translation along x-axis.

2.2 Kinematic Motion

In this work, kinematic control is used taking into consideration the relationship between the operational space and joint space established by the dual quaternion Jacobian matrix $\boldsymbol{J}$, that satisfies:

$$\dot{\underline{\boldsymbol{x}}}_E = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{8}$$

where $\dot{\underline{\boldsymbol{x}}}_E$ is the first derivative of the dual quaternion that represents the effector's pose and $\dot{\boldsymbol{q}}$ is the joint velocity vector.

Deducing the Jacobian matrix as algorithmically described by Adorno (2011) and focusing on the trajectory following over the operational space, the nonlinear model that describes the manipulator motion was written as:

$$\begin{bmatrix} \dot{\underline{\boldsymbol{x}}}_E \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{J}(\boldsymbol{q}) \\ \boldsymbol{I}_6 \end{bmatrix} \dot{\boldsymbol{q}} \tag{9}$$

Which allows to predict how the end-effector pose behaviors based on different joints configurations. That explicit model used as prediction mechanism is essential for the following predictive control strategy.

3. NONLINEAR MODEL PREDICTIVE CONTROL

Model Predictive Control (MPC) is a methodology of control systems design that uses prediction strategies and optimization based on an explicit model (Camacho and Bordons, 2007), considering the constraints that the system's variables must obey. Usually, the optimization problem considers a limited prediction horizon of states and control actions, being employed to the actual plant only the first control action. The updated problem is solved again at the next iteration, with newly acquired measurements, a characteristic known as receding horizon.

There are many ways to implement a MPC controller, with different combinations of models, prediction strategies, and optimization algorithms. Although classic MPC approaches consider linear models, some systems present nonlinear behavior that do not work properly through linearization, leading to a Nonlinear Model Predictive Control (NMPC) problem. In this work, the NMPC setup suggested by MATLAB MPC Toolbox was used and presents the following characteristics:

- a nonlinear model for the system;
- a quadratic objective function in a finite prediction horizon;
- the future behavior of the variables of interest must follow the reference and is subject to a term that punishes abrupt changes on the input (move suppression);
- the solution of the nonlinear problem provides the optimal inputs.

Considering the system's nonlinear model expressed in Equation (9), the formulations for the NMPC problem, were defined in discrete time as follows:

$$\begin{aligned} \text{minimize} \quad & C(z_k) = C_y(z_k) + C_u(z_k) + C_{\Delta u}(z_k), \\ \text{subjected to} \quad & \boldsymbol{x}(k+1) = f(\boldsymbol{x}(k), \boldsymbol{u}(k)) \\ & \boldsymbol{x}_{\min} \leq \boldsymbol{x}(k) \leq \boldsymbol{x}_{\max}, \\ & \boldsymbol{u}_{\min} \leq \boldsymbol{u}(k) \leq \boldsymbol{u}_{\max}, \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0. \end{aligned} \tag{10}$$

The cost function $C(z_k)$ to be minimized is composed by the sum of the following therms, each representing a particular aspect of the controller's performance:

$$\begin{aligned} C_y(z_k) &= \sum_{i=0}^{N} [\boldsymbol{r}(k+i+1) - \boldsymbol{x}(k+i+1)]^T \\ &\quad \times \boldsymbol{Q}[\boldsymbol{r}(k+i+1) - \boldsymbol{x}(k+i+1)] \\ C_u(z_k) &= \sum_{i=0}^{N} [\boldsymbol{u}(k+i)]^T \boldsymbol{R}_u [\boldsymbol{u}(k+i)] \\ C_{\Delta u}(z_k) &= \sum_{i=0}^{N} [\boldsymbol{u}(k+i) - \boldsymbol{u}(k+i-1)]^T \\ &\quad \times \boldsymbol{R}_{\Delta u}[\boldsymbol{u}(k+i) - \boldsymbol{u}(k+i-1)]. \end{aligned} \tag{11}$$

With all system's variables observable, the output in C_y is given by the state vector, therefore, $\boldsymbol{r}(k) - \boldsymbol{x}(k)$ represents the trajectory error. C_u implements the effort of the control actions $\boldsymbol{u}(k)$. At last, $C_{\Delta u}(z_k)$ implements move suppression, alleviating the rate $\boldsymbol{u}(k) - \boldsymbol{u}(k-1)$ of the input. Matrices $\boldsymbol{Q}$, $\boldsymbol{R_u}$ and $\boldsymbol{R}_{\Delta u}$ are tuneable weight matrices associated to each of those respective properties, where $\boldsymbol{Q}$ is positive semidefinite and the matrices $\boldsymbol{R_u}$ and $\boldsymbol{R}_{\Delta u}$ are positive definite.

Bounding the region where the optimization happens, the constraints are considered rigid and are presented as the kinematic model of the system through function f, which is the discrete version of the presented in Equation (9) using the implicit trapezoidal rule. Also, physical limitations for joint configurations are implemented as $\boldsymbol{x}_{min}$ e $\boldsymbol{x}_{max}$ and velocities as $\boldsymbol{u}_{min}$ e $\boldsymbol{u}_{max}$. It is imposed that the initial conditions are given by the actual state of the system $\boldsymbol{x}_0$. That initial conditions imposition on constraints is a remarkable feature of MPC methodology, which improves safety and motivates its application.

Once assuming the nonlinear function f, the present method diverges from traditional MPC applications, featuring a form of NMPC. Thus recurring to more powerful techniques of prediction and optimization. The NMPC problem was written as a nonlinear programming problem described as:

$$\text{minimize} \quad \boldsymbol{C}(\boldsymbol{x}), \quad \text{subjected to} \begin{cases} c(\boldsymbol{x}) \leq 0, \\ c_{eq}(\boldsymbol{x}) = 0, \\ \boldsymbol{Ax} \leq \boldsymbol{b}, \\ \boldsymbol{A_{eq}x} = \boldsymbol{b_{eq}}, \\ \boldsymbol{l_b} \leq \boldsymbol{x} \leq \boldsymbol{u_b}. \end{cases} \tag{12}$$

Where $c(\boldsymbol{x})$ and $c_{eq}(\boldsymbol{x})$ are given in function of the optimized variables, representing nonlinear inequalities and equalities respectively. The matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$ represent linear equalities relationships, while $\boldsymbol{A_{eq}}$ and $\boldsymbol{b_{eq}}$ linear inequalities relationships. Lastly, $\boldsymbol{l_b}$ represent lower bounds for optimized variables and $\boldsymbol{u_b}$ the upper bounds.

4. SIMULATION SCENARIOS

The implemented algorithms were written in MATLAB environment, which offers a wide set of libraries that made quick development possible avoiding deviations from the purposes of this work. DQ Robotics library (Adorno and Marinho, 2020) covers the dual quaternion algebra, allowing kinematic modeling with proper Denavit-Hartenberg parameters, as disposed in Table 1. Model Predictive Control Toolbox, on the other hand, provides functions and structures for the design of both linear and nonlinear predictive control programs.

Table 1. Table containing Denavit-Hartenberg parameters of Denso VP-6242 with international system measures.

Link	θ_i	d_i	a_i	α_i
1	θ_1	0.125	0	$\frac{\pi}{2}$
2	$\theta_2 + \frac{\pi}{2}$	0	0.210	0
3	$\theta_3 - \frac{\pi}{2}$	0	−0.075	$-\frac{\pi}{2}$
4	θ_4	0.210	0	$\frac{\pi}{2}$
5	θ_5	0	0	$-\frac{\pi}{2}$
6	θ_6	0.070	0	0

In order to achieve the solution of the NMPC problem, the library deploys an optimization solver named *fmincon*, designed to find the minimum of constrained nonlinear multivariable problems, specifically as a nonlinear programming problem. The solver relies on a sequential quadratic programming (SQP) algorithm, which aims to solve the nonlinear programming problem through series

ISSN: 2525-8311

DOI: 10.20906/CBA2022/3675

of approximations of multiple convex optimization sub problems.

In addition to the problem developed in section 3, physical constraints were determined as shown in Table 2, also, minimum and maximum velocities were constrained in the interval between $-\pi$ and π, a stated safe region of applicable control actions. For validation, three scenarios were designed intending to express analogies to recurrent activities where robotic manipulators are traditionally employed. Those are described as the present section progresses.

Table 2. Joint space constraints.

Joint	Minimum Position	Maximum Position
1	$-160°$	$160°$
2	$-120°$	$120°$
3	$-160°$	$-19°$
4	$-160°$	$160°$
5	$-120°$	$120°$
6	$-360°$	$360°$

4.1 Regulation Problem Scenario

This first scenario aims to acknowledge the capability of the controller to conduct the robot's effector from a starting pose, $\boldsymbol{p}_{initial}$ and $\phi_{initial}$, to another given, $\boldsymbol{p}_{final}$ and ϕ_{final}. For practical purposes, trajectory is represented by the final pose, which is fixed through simulation time t_{simul}. Scenario parameters are expressed in Table 3

Table 3. Scene parameters used in regulation problem.

Parameter	Value
$\boldsymbol{p}_{initial}$	$(-0.075, 0, 0.615)$
$\boldsymbol{p}_{final}$	$(0.2, 0, 0.2)$
$\phi_{initial}$	$(0°, 0°, 0°)$
ϕ_{final}	$(0°, 180°, 0°)$
t_{simul}	10s

4.2 Pick and Place Scenario

A pick and place task is implemented creating some association to usual industrial applications. For this task, four quarters are assumed constituting the trajectory sharing equally the simulation time, t_{simul}. Starting at the initial position $\boldsymbol{p}_{initial}$, the first quarter represents the picking at position $\boldsymbol{p}_1$, second quarter represents the return to the starting position, the third quarter represents the placement at $\boldsymbol{p}_2$, with the last quarter represented by another return to initial position. Through the motion execution, orientation is expected to be maintained as a fixed ϕ and each quarter reference is expressed fixed in time as sequential regulation problems. Table 4 contains the parameters used in the simulations.

4.3 Circular Trajectory Scenario

Aiming to imitate tasks like painting and welding, a cyclical trajectory was implemented with a circular shape. The effector, initially positioned on $\boldsymbol{p}_{initial}$ with orientation $\phi_{initial}$, is expected to track the ongoing trajectory, a circumference centered at c of radius r parallel to the yz-plane, with a fixed desired orientation ϕ_d. Table 4 contains the parameters used in the simulations.

Table 4. Scene parameters used in pick and place problem.

Parameter	Value
$\boldsymbol{p}_{initial}$	$(0.21, 0, 0.41)$
$\boldsymbol{p}_1$	$(0.2, 0.2, 0.2)$
$\boldsymbol{p}_2$	$(0.2, -0.2, 0.2)$
ϕ	$(0°, 90°, 0°)$
t_{simul}	20s

Table 5. Scene parameters used in circular trajectory problem.

Parameter	Value
$\boldsymbol{p}_{inicial}$	$(-0.075, 0, 0.615)$
$\phi_{inicial}$	$(0°, 0°, 0°)$
$\boldsymbol{c}$	$(0.3, 0, 0.3)$
r	0.05
ϕ_d	$(0°, 90°, 0°)$
t_{simul}	10s

5. RESULTS

In this section, tuned parameters are shown for each scenario, as also, the controller behavior is observed through input and output data which makes possible to compute performance metrics.

5.1 Regulation Problem Results

Table 6 shows the parameters associated to the NMPC algorithm used to solve the regulation problem from section 4.1. The choice of $\boldsymbol{Q}(k)$ notably prioritizes the quaternionic units related to translation, which demand greater joint efforts on tracking, therefore punishing intensively position errors. Matrices $\boldsymbol{R}$ and $\boldsymbol{R}_{\boldsymbol{\Delta u}}$, are proportional and punish control actions on joints that are closer to the robot's base due to the efforts of moving it's whole body. The size of the prediction horizon N was set empirically based on the trade off between computational cost and performance contributions. At last, and sampling time T_s assumes a plausible value according to Denso VP-6242 specifications.

Table 6. NMPC parameters used in regulation problem.

Parameter	Value
$\boldsymbol{Q}(k)$	diag$(40, 40, 40, 40, 100, 100, 100, 100, 0, 0, 0, 0, 0, 0)$ [1]
$\boldsymbol{R}_u(k)$	diag$(6, 5, 4, 3, 2, 1)$ [1]
$\boldsymbol{R}_{\boldsymbol{\Delta u}}(k)$	diag$(30, 25, 20, 15, 10, 5)$ [1]
N	10
T_s	0.1s

Figure 2 presents in blue, the quaternionic units behaviour over time during the 10 seconds of simulation, compared to the desired trajectory in orange. The graphics show that the system reaches reference before timeout with null steady state error.

Figure 5 shows the control actions representing the velocities assigned to each of the six joints during the simulation, throughout which, no limitations are trespassed, also presenting a smooth behavior. It's important to highlight that once the system reaches reference, it ceases all movement, reaching also a quiescent state.

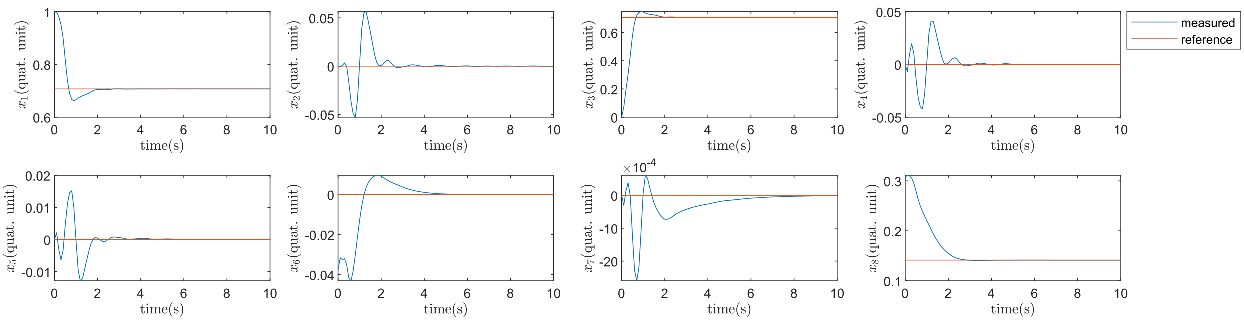

Figure 2. Effector's reference and outputs in dual quaternion resulting from NMPC applied to regulation problem.

5.2 Pick and Place Results

Extending the regulation problem, the pick and place task described in section 4.2 has it's tuning parameters shown in Table 7. Due to the abrupt changes implemented by this trajectory, the weights of $\boldsymbol{Q}(k)$ associated to the quaternionic units that represent translations are incremented.

Table 7. NMPC parameters used in pick and place problem.

Parâmetro	Valor
$\boldsymbol{Q}(k)$	$\mathrm{diag}(40, 40, 40, 40, 200, 200, 200, 200, 0, 0, 0, 0, 0, 0)$ [1]
$\boldsymbol{R_u}(k)$	$\mathrm{diag}(6, 5, 4, 3, 2, 1)$ [1]
$\boldsymbol{R_{\Delta u}}(k)$	$\mathrm{diag}(30, 25, 20, 15, 10, 5)$ [1]
N	10
T_s	0.1s

Figure 3 presents, again in blue, the quaternionic units behaviour over time, compared to the desired trajectory in orange. Peaks of error are observed when the trajectory suggests abrupt and irreproducible movements, however, during each quarter there is at least an instant of time where all of the quaternionic units are synchronized reaching reference.

Figure 6 shows the control actions once again with an expected behavior, smooth inside safe intervals. It can be seen that quiescent state is reached at least once in each quarter between great efforts to respond those abrupt changes of the task.

5.3 Circular Trajectory Results

Considering problems such as the circular trajectory tracking, the urgency to follow a more detailed trajectory is taken into consideration. With that, an even more aggressive matrix $\boldsymbol{Q}(k)$ was proposed, shown in Table 8.

Table 8. NMPC parameters used in circular trajectory problem.

Parameter	Value
$\boldsymbol{Q}(k)$	$\mathrm{diag}(125, 125, 125, 125, 300, 300, 300, 300, \mathrm{zeros}(6))$ [1]
$\boldsymbol{R_u}(k)$	$\mathrm{diag}(6, 5, 4, 3, 2, 1)$ [1]
$\boldsymbol{R_{\Delta u}}(k)$	$\mathrm{diag}(30, 25, 20, 15, 10, 5)$ [1]
N	10
T_s	0.1s

Figure 4 presents the measured quaternionic units in blue, starting at a certain distance from reference in orange, which denotes the ongoing circular trajectory in space. A quick motion is observed at the start of the simulation until the robot reaches the pace defined by the trajectory in about the first two seconds, after that all efforts intend to maintain the desired orientation while executing the circular movement.

At last, the control actions generated by NMPC for the circular trajectory following are presented in Figure 7. And once again, smooth actions are observed, while all limitations are held.

6. CONCLUSION

This work presented a nonlinear model based predictive control (NMPC) application, combined with a dual quaternion modeling algebra for a robotic manipulator. At each iteration of a prediction horizon, optimal control actions

[1] The function diag : $\mathbb{R}^n \rightarrow \mathbb{R}^{n\times n}$ returns a diagonal matrix with elements defined by the arguments.

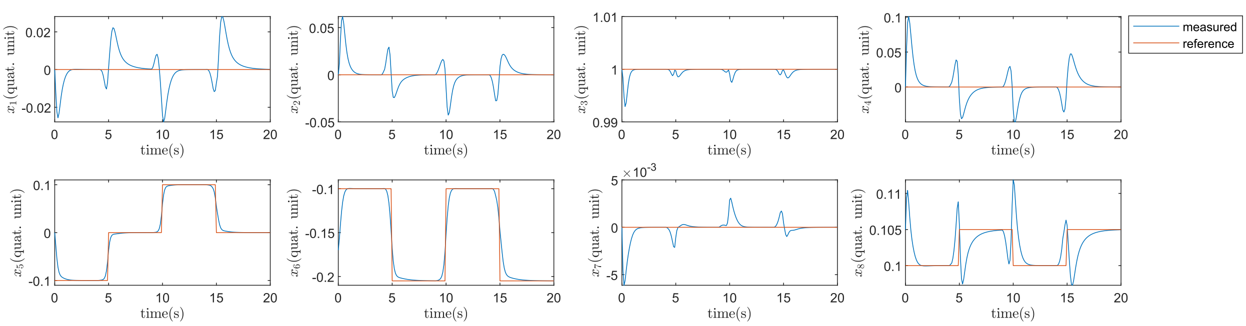

Figure 3. Effector's reference and outputs in dual quaternion resulting from NMPC applied to pick and place problem.

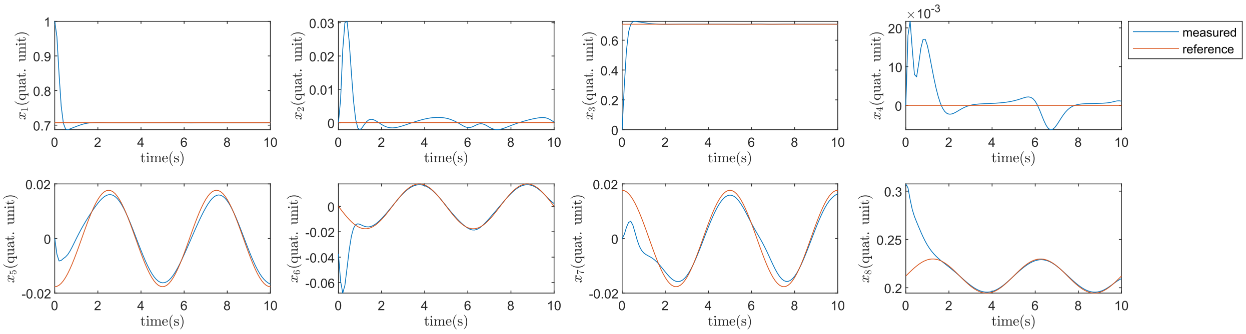

Figure 4. Effector's reference and outputs in dual quaternion resulting from NMPC applied to the circular trajectory.

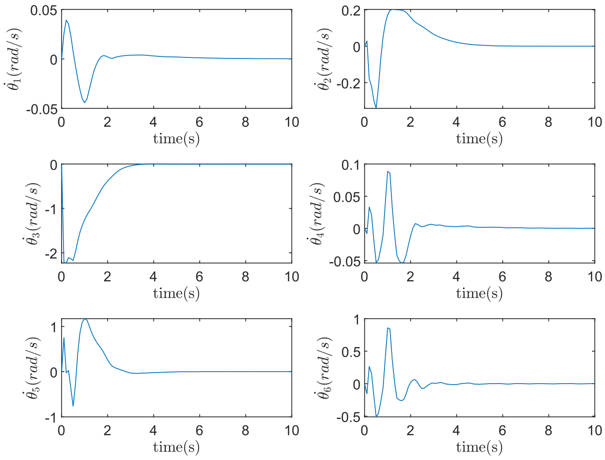

Figure 5. Optimal inputs resulting from NMPC applied to regulation problem.

were computed assuming and complying to all of the constraints imposed by the commercial robotic manipulator physical limitations.

The robot used was fabricated by Denso having 6 degrees of freedom, model VP-6242, which inspired the built nonlinear model. With that, the manipulator became capable to work safely, without singularity scenarios, and presenting smooth movements even when starting far from reference. A great advantage of this technique over other linear approaches is the possibility of extending the prediction horizon without losing the model fidelity.

Future work may consider the deduction of dynamical components for the enhancement of the current nonlinear model. Furthermore, implementations of structural and external obstacles taking advantage of dual quaternion modeling creates some expectations. Also, availing that the developments were made in MATLAB, the connection to Quanser systems used to command the real plant located at Instituto de Computação in Universidade Federal de Alagoas, is highly facilitated.

ACKNOWLEDGES

We would like to thank our research group Easy Engineering and Systems that is part of the Computing Institute (IC) from Federal University of Alagoas (UFAL).

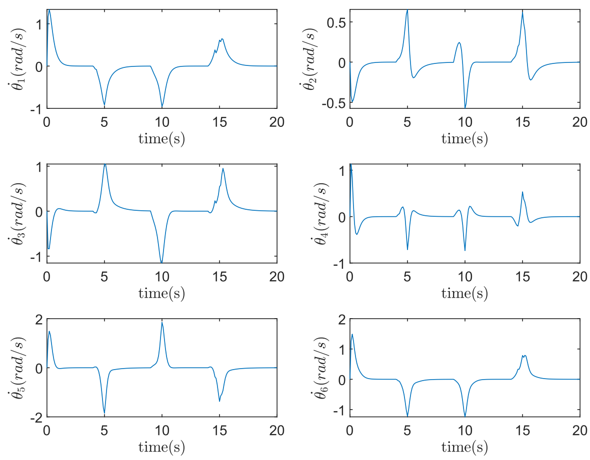

Figure 6. Inputs in blue resulting from NMPC applied to pick and place problem.

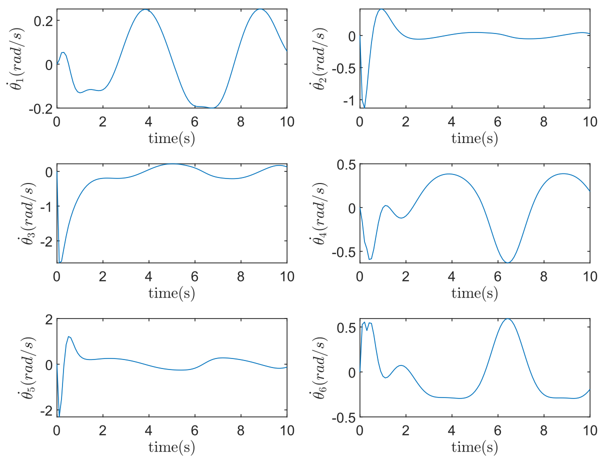

Figure 7. Inputs in blue resulting from NMPC applied to the circular trajectory.

REFERENCES

Abaunza, H., Castillo, P., Victorino, A., and Lozano, R. (2017). Dual quaternion modeling and control of a quad-rotor aerial manipulator. *Journal of Intelligent Robotic Systems*, 88, 267–283. doi:10.1007/s10846-017-0519-4.

Adorno, B.V. (2011). *Two-arm Manipulation: From Manipulators to Enhanced Human-Robot Collaboration*. Theses, Université Montpellier II - Sciences et Techniques du Languedoc. URL https://tel.archives-ouvertes.fr/tel-00641678.

Adorno, B.V. and Marinho, M.M. (2020). Dq robotics: A library for robot modeling and control. *IEEE Robotics & Automation Magazine*, 1–15. doi:10.1109/MRA.2020.2997920. URL https://ieeexplore.ieee.org/document/9136790/.

Camacho, E.F. and Bordons, C. (2007). *Model Predictive control*. Advanced Textbooks in Control and Signal Processing. Springer London, London, 2nd edition. doi:10.1007/978-0-85729-398-5. URL http://link.springer.com/10.1007/978-0-85729-398-5.

Clifford (1871). Preliminary sketch of biquaternions. *Proceedings of the London Mathematical Society*, s1-4(1), 381–395. doi:https://doi.org/10.1112/plms/s1-4.1.381. URL https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s1-4.1.381.

Craig, J.J. (2009). *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India.

Eckhoff, M., Kirschner, R.J., Kern, E., Abdolshah, S., and Haddadin, S. (2022). An mpc framework for planning safe x0026; trustworthy robot motions. In *2022 International Conference on Robotics and Automation (ICRA)*, 4737–4742. doi:10.1109/ICRA46639.2022.9812160.

Elsisi, M., Mahmoud, K., Lehtonen, M., and Darwish, M.M.F. (2021). Effective nonlinear model predictive control scheme tuned by improved nn for robotic manipulators. *IEEE Access*, 9, 64278–64290. doi:10.1109/ACCESS.2021.3075581.

Hamilton, W.R. (1844). Lxxviii. on quaternions; or on a new system of imaginaries in algebra: To the editors of the philosophical magazine and journal. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(169), 489–495.

Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). Modelling, planning and control. *Advanced Textbooks in Control and Signal Processing. Springer,*.