International Journal of Engineering Insights: (2023) Vol. 1, Nro.1, Regular Paper
https://doi.org/10.61961/injei.v1i1.6

Simultaneous motion and shape control of redundant mobile manipulators

María del Carmen Claudio · Viviana Moya · Emanuel Slawiñski · Vicente Mut

Received: 28 August 2023 / Accepted: 29 October 2023

Abstract: This work proposes the design of a simultaneous motion and shape control of a redundant mobile manipulator. The method is based on the construction of a variable weighting matrix that allows to control of specific sections of this type of robot composed of a mobile platform and manipulator arm. The weight matrix allows for prioritizing desired parts of the robot through cost functions that depend on an additional variable defined in this work, which can be changed automatically according to some pre-established criteria or by a human operator if the robot is teleoperated. In addition, simulations are included to show the usefulness of the proposal.

Keywords Weighting Matrix · Redundant Robot · Robot Shape Control · Jacobian Matrix

1 Introduction

The advancement of robotics has facilitated the execution of tasks that previously could only be performed by human operators, either by replacing them in repetitive and tedious tasks or by showing themselves as complementary in dangerous scenarios [1]. In this context, a robot's control actions may be generated internally (from automatic references from sensors) or may come from a distant environment, where redundant robots are generally used, e.g. a mobile manipulator.

Considering that the robot can execute a task in an autonomous way, the difficulty of the mission may require more degrees of freedom (DoF) than apparent, either for safety or design requirements. In this case, the amount of additional degrees of freedom would allow to execution of secondary actions, where appropriate control techniques would allow to use of certain joints (e.g. the mobile platform) more frequently compared to other DoFs (e.g. the robotic manipulator) [2]. Such techniques are developed in order to fulfill additional goals to the primary one, such as obstacle avoidance, singularity avoidance, energy saving, maximum manipulability, and so on. Therefore, having full control over the redundancy characteristics of the system could facilitate the proper execution of the [3] mission.

On the other hand, if in isolated cases of the task, it is required to generate higher magnitude control commands to the mobile platform and lower magnitude commands to the robotic manipulator, these strategies must be embedded in the remote control algorithm. However, handling such a large number of degrees of freedom requires additional physical and mathematical analysis. Based on the literature, two common ways are proposed to solve this problem: section the robot into two main parts and control them independently or relate the whole robotic system (mobile manipulator) through a single Jacobian matrix and control it with some technique.

The first way is to control each section of the robot independently (i.e. in the mobile manipulator, to control either the platform or the robotic manipulator in isolation), treating it as a set of mechanically linked mechanisms where one dynamically perturbs the other [4]. This solution, although obvious and easily implementable, poses discrete activations and deactivations of each part of the robot, additionally needing to generate references to each section of the system. The most notorious disadvantages are the need for non-unified mathematical modeling, controllers for each of the sections, discrete switching that could generate control instabilities, and possibly the involvement of more than one human operator with different robotic controls (the latter two in cases of the remote control).

For the second form, completely activating/deactivating different parts of the robot depending on the situation would not be feasible by directly using the Jacobian matrix that relates Cartesian reference velocities to joint velocities. In this aspect, using strategies such as shape control (as a pri-

María del Carmen Claudio
Inmersoft Technologies
Quito, Ecuador
Tel.: +593 98 787 3246
mcclaudio@inmersoft.com

Emanuel Slawiñski · Vicente Mut
Instituto de Automática, Universidad Nacional de San Juan
San Juan, Argentina

Viviana Moya
Universidad Internacional del Ecuador
Quito, Ecuador

mary objective) to command the entire robotic system in a unified manner is a straightforward solution, however, there are technique-specific constraints to be able to position the end effector in a desired state, given that the reach of the operating end to a desired position would be located in the null space of the homogeneous solution [3]. An appropriate solution to this problem is to modify the behavior of the Jacobian by multiplying it by a weighting matrix, which weighs sections of the matrix to give/remove priority to certain joints. The modification of the behavior of a pondered Jacobian (or weighted) is a concept widely used in the literature. In this aspect, works such as [5] use this technique to limit the positions of the joints with the objective of limiting their achievable range, forcing them to work in safe limits (preventing the actuators from exceeding positions that would damage them). In this type of work, the weight matrix is based on derivable functions that cause zero velocities when the joints reach the safe limits, with a continuous reduction until reaching zero (causing complete immobility in the effector). Similarly, [6] proposes the use of this technique to avoid collisions between parts of the robot, where each link is represented by a protective sphere. The dangerous approach between links (and therefore spheres) causes the deceleration of all the actuators involved, all this based on a function that depends on the diameters and distances between the virtual spheres. In addition to the weight matrix, this body of work has in common the continuous way in which joint positions are constrained, which depends on an internal factor (range of maximum and minimum positions for [5] and collision between sphere surfaces for [6]). In the literature, no proposal is determined that uses this technique to weight robot sections through a weight matrix, specifically using an additional variable that depends on task variables or can be modified by an operator.

The main contribution of this work is the design of a weighting matrix that includes cost functions for the continuous selection of sections of a highly redundant robot, specifically a redundant mobile manipulator robot (MM). In this way, by means of the weight matrix, it is proposed to control either isolated sections of the MM (when the switching variable is at its extremes), or to give execution priority to desired parts of the mechanism (when the variable is set to intermediate values), considering that the switching value is task-dependent and can be modified in real time.

2 Mathematical Foundations

2.1 MM velocity relationships

The differential of the geometric representation with respect to time makes it possible to obtain the relationship between the joint velocity and the Cartesian velocity of the robotic system,

Table 1: DH parameters of the manipulator under consideration

Link	q	d	a	α
1	$\phi + q_1$	h	0	$\pi/2$
2	$q_2 + \pi/2$	0	l_T	0
3	$q_3 + \pi/2$	lb	0	$\pi/2$
4	$q_4 + \pi/2 + atan(30/264)$	0	l_{1a}	0
5	$q_5 + 3\pi/4 - atan(30/258)$	0	$-l_{1b}$	$\pi/2$
6	q_6	$l2$	0	$-\pi/2$
7	q_7	0	0	$\pi/2$
8	q_8	$l3$	0	0
9	0	0	0	$\pi/2$
10	$\pi/2$	0	0	0

$$\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \tag{1}$$

where $\mathbf{J}(\mathbf{q})$ is the matrix relating these velocities and is generally referred to as the Jacobian of the whole robot, while $\dot{\mathbf{x}}$ is the Cartesian velocity with respect to a global reference frame. Fig. 1 shows a redundant mobile manipulator. Then, to take into account the characteristics of the robotic mechanism, one can define a generalized variable of the form

$$\mathbf{q} = \begin{bmatrix}\mathbf{q_M}^T \ \mathbf{q_R}^T\end{bmatrix}^T, \tag{2}$$

where $\mathbf{q_M} \in \mathbb{R}^{n_M}$ represents the integral of the linear and angular velocities of the moving platform and $\mathbf{q_R} \in \mathbb{R}^{n_R}$ represents the angular positions of the robotic arm.

To determine the kinematic behavior of the robotic manipulator, Denavit Hartenberg representation (DH) is used [7]. Table 1 shows the DH parameters to define the geometric relationship of the robotic manipulator used in this work. As for the mobile platform [8], the possibility of independently controlling all four wheels of a skid-steering vehicle greatly extends the applications that these robots can have, especially when the applications must be developed outdoors and with non-regular surfaces. The position of a skid-steering robot [9] can be represented by

$$\mathbf{q_M} = \begin{bmatrix}x \ y \ \phi\end{bmatrix}^T, \tag{3}$$

where $\begin{bmatrix}x \ y\end{bmatrix}^T$ are the coordinates of O_r (arm reference systems) with respect to the global reference frame ($< W >$), while ϕ is the angle of rotation with respect to $< W >$. Kinetically, the mathematical representation of a skid-steering robot can be expressed as follows.

$$\dot{\mathbf{q}}_\mathbf{M} = \mathbf{S}(\mathbf{q_M})\mathbf{v} + \mathbf{A}(\mathbf{q_M})v_y, \tag{4}$$

where $\mathbf{v} = \begin{bmatrix}v_x \ \omega\end{bmatrix}^T$ represents the linear and angular velocity of the robot, v_y is the lateral velocity of the robot (generally produced by sliding), while $\mathbf{S}(\mathbf{q_M})$ and $\mathbf{A}(\mathbf{q_M})$ are

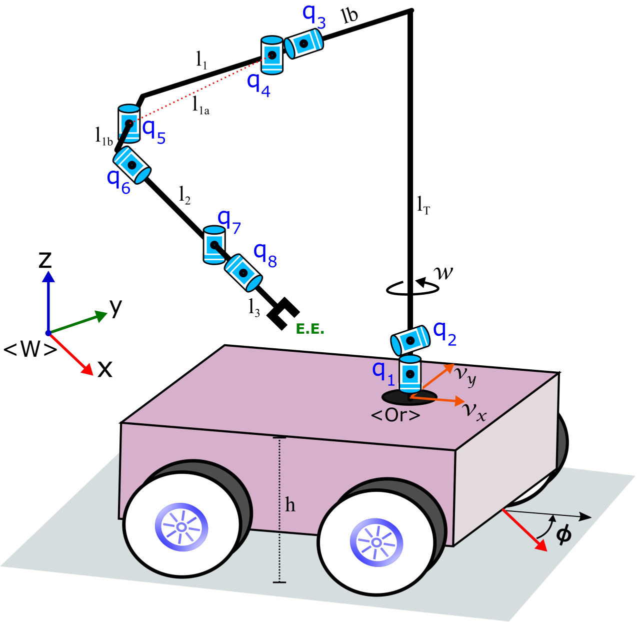

Fig. 1: Mobile Manipulator robot

defined as:

$$\mathbf{S}(\mathbf{q_M}) = \begin{bmatrix} cos(\phi) & -a\sin(\phi) \\ sin(\phi) & a\cos(\phi) \\ 0 & 1 \end{bmatrix}, \tag{5}$$

$$\mathbf{A}(\mathbf{q_M}) = \begin{bmatrix} -sin(\phi) \\ cos(\phi) \\ 0 \end{bmatrix}. \tag{6}$$

In this way, the velocity of the robotic structure of the mobile manipulator can be defined as:

$$\boldsymbol{\eta} = \dot{\mathbf{q}} = \left[\dot{\mathbf{q}}_{\mathbf{M}}^T \; \dot{\mathbf{q}}_{\mathbf{R}}^T\right]^T \in \mathbb{R}^n, \tag{7}$$

where $n = n_M + n_R$, while η_i (con $i = 1, 2, ..., n$) is the i-th velocity of the MM.

Thus, the displacement velocity of the Operating End (O.E) in Cartesian space is represented as follows:

$$\dot{\mathbf{x}}_{\mathbf{R}} = \mathbf{J}(\mathbf{q})\boldsymbol{\eta}, \tag{8}$$

where $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{m\times(n_M+n_R)}$ is a Jacobian matrix that relates the joint velocities to the Cartesian velocities of the complete robotic mechanism.

2.2 Matrix Design $\mathbf{W}$

Assuming that we have a single Jacobian of the whole robot, we need to modify the solutions given by the inverse kinematics based controller. Cost functions governed by a ϵ variable solve the given problem. Thus, the selection of robot sections of interest is achieved through the construction of a diagonal weight matrix $\mathbf{W}$. Depending on the application, the construction of the matrix $\mathbf{W}(\epsilon)$ uses both functions placed on the elements of its diagonal. Consider the case of a mobile manipulator, divided into two parts of interest: 1) the mobile platform, and 2) the robotic manipulator. Thus, a number of α references is required to control the moving platform, while β references are required to control the manipulator.

Thus, the weight matrix for selecting degrees of freedom depending on ϵ for the MM can be constructed with w_i elements. The scalars of w_i are defined by:

$$w_i = \begin{cases} f_1(\epsilon) \; for \;\; i = 1 : \alpha \\ f_2(\epsilon) \; for \;\; i = \alpha + 1 : \alpha + \beta, \end{cases} \tag{9}$$

in this way,

$$\mathbf{W}(\epsilon) = \begin{bmatrix} w_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 \\ 0 & 0 & w_\alpha & 0 & 0 & 0 \\ 0 & 0 & 0 & w_{\alpha+1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{\alpha+\beta} \end{bmatrix}.$$

The solution considers a pair of candidate cost functions denoted by $f_1(\epsilon)$ and $f_2(\epsilon)$, both with different properties given the application requirements.

Let us consider $f_1(\epsilon)$ as a decreasing function that has a large value when ϵ tends to zero, and tends to unity when the adjustable variable ϵ also tends to 1; while we define $f_2(\epsilon)$ as an increasing function that tends to a large value when ϵ tends to unity, and tends to unity when ϵ tends to zero. Among many possibilities, we adopt the following functions:

$$f_1(\epsilon) = \frac{1}{b_1\epsilon + b_2}, \tag{10}$$

$$f_2(\epsilon) = a_1 e^{a_2\epsilon + a_3} + 1, \tag{11}$$

where a_i and b_i are constants to adjust the shape of the functions.

The behavior of the inverse of $\mathbf{W}$ is directly related to the final solution of the controller (as will be seen below). Thus, the change from 0 to 1 in ϵ continuously deactivates the moving platform, while activating the manipulator. At the extremes of ϵ, each of the robots is either activated or completely deactivated.

2.3 Resolution of the redundancy of inverse kinematics

The displacement velocity of an operating end can be calculated using differential kinematics ($\dot{\mathbf{x}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$), however, for practical purposes the parameters set are given by the inverse kinematics solution:

$$\boldsymbol{\eta} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{x}}, \tag{12}$$

where $\mathbf{J}^{\dagger}(\mathbf{q})$ is the pseudo-inverse of $\mathbf{J}(\mathbf{q})$. Given the redundancy of robotic systems, an exact solution for $\boldsymbol{\eta}$ does not normally exist, but an approximation can be found through the use of least squares by calculating $\|\dot{\mathbf{x}} - \mathbf{J(q)}\boldsymbol{\eta}\|$, obtaining the following solution:

$$\boldsymbol{\eta} = \mathbf{J}^{\dagger}(\mathbf{q})\dot{\mathbf{x}} + (\mathbf{I} - \mathbf{J}(\mathbf{q})^{\dagger}\mathbf{J}(\mathbf{q}))\mathbf{z_0}, \tag{13}$$

where $\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J}$ is the projection on the null space generated by $\mathbf{J}(\mathbf{q})$ and $\mathbf{z_0}$ is an arbitrary vector in the homogeneous part of the solution which may include secondary objectives, e.g. operating end orientation, energy saving or maximum manipulability.

Considering that there are two control variables: $\mathbf{x_p}$ and $\mathbf{x_o}$, in our case, we propose to find a solution that allows us to meet both objectives through the redundancy of the robotic system. Similar to how (13) is obtained, a least-squares solution allows to use the null space of the redundant system, having:

$$\begin{aligned}\boldsymbol{\eta} =& \mathbf{J_p^{\dagger}}\dot{\mathbf{x}}_\mathbf{p} + (\mathbf{I} - \mathbf{J_p}^{\dagger}\mathbf{J_p})\tilde{\mathbf{J}}_\mathbf{o}^{\dagger}(\dot{\mathbf{x}}_\mathbf{o} - \mathbf{J_o}\mathbf{J_p^{\dagger}}\dot{\mathbf{x}}_\mathbf{p}) \\ &+ (\mathbf{I} - \mathbf{J_p}^{\dagger}\mathbf{J_p})(\mathbf{I} - \tilde{\mathbf{J}}_\mathbf{o}^{\dagger}\tilde{\mathbf{J}}_\mathbf{o})\mathbf{z}^*.\end{aligned} \tag{14}$$

where $\tilde{\mathbf{J}}_\mathbf{o}^{\dagger} \triangleq \mathbf{J_o}(\mathbf{I} - \mathbf{J_p^{\dagger}}\mathbf{J_p})$, with $\tilde{\mathbf{J}}_\mathbf{o}^{\dagger}$ being the pseudoinverse of $\tilde{\mathbf{J}}_\mathbf{o}$ and $\mathbf{z}^*$ is an arbitrary joint velocity vector that could contain tertiary targets.

Now, based on [10], the constructed weight matrix is used to compute the Moore-Penrose right-handed pseudoinverse. In this way, the inverses of the Jacobian are calculated as follows:

$$\mathbf{J_p^{\dagger}} = \mathbf{W^{-1}}\mathbf{J_p^T}[\mathbf{J_p}\mathbf{W^{-1}}\mathbf{J_p^T}]^{-1}, \tag{15}$$

$$\tilde{\mathbf{J}}_\mathbf{o}^{\dagger} = \mathbf{W^{-1}}\tilde{\mathbf{J}}_\mathbf{o}^\mathbf{T}[\tilde{\mathbf{J}}_\mathbf{o}\mathbf{W^{-1}}\tilde{\mathbf{J}}_\mathbf{o}^\mathbf{T}]^{-1}. \tag{16}$$

In order to correct the numerical error, information obtained from the robot in real time allows us to calculate both position and orientation errors. From (14), results the final closed-loop inverse kinematics solution

$$\boldsymbol{\eta} = \mathbf{J_p^{\dagger}}\dot{\mathbf{x}}_{\mathbf{p_{ref}}} + (\mathbf{I} - \mathbf{J_p}^{\dagger}\mathbf{J_p})\tilde{\mathbf{J}}_\mathbf{o}^{\dagger}(\dot{\mathbf{x}}_{\mathbf{h_{ref}}} - \mathbf{J_o}\mathbf{J_p^{\dagger}}\dot{\mathbf{x}}_{\mathbf{p_{ref}}}), \tag{17}$$

where $\dot{\mathbf{x}}_{\mathbf{p_{ref}}}$ and $\dot{\mathbf{x}}_{\mathbf{h_{ref}}}$ are references in Cartesian coordinates generated by the human operator through a local robot.

2.4 Control references

The ϵ variable can be calibrated considering several factors, automatically or teleoperated. For this work, we assume that the master robot has an additional degree of freedom that allows a continuous online modification of this parameter and that the operator configures it depending on the requirements of the task. On the other hand, the reference to follow is generated by a position controller, based on the position error. Thus, we have:

$$\dot{\mathbf{x}}_{\mathbf{p_{ref}}} = \mathbf{K_p}(\mathbf{e_p}) = \mathbf{K_p}(\mathbf{x_{rd}} - \mathbf{x_r}), \tag{18}$$

with $\mathbf{x_{rd}}$ being the desired positions; $\mathbf{K_p}$ is a positive definite matrix that regulates the position errors. On the other hand, we have established that the redundancy of the MM allows the inclusion of secondary objectives. Keeping in mind that the secondary target is set in the null space of the overall solution, this requirement is fulfilled as long as it does not interfere with the main solution. For our case, the orientation of the end-effector can be modified in-line according to task requirements and defined as follows:

$$\dot{\mathbf{x}}_{\mathbf{h_{ref}}} = \mathbf{K_o}\mathbf{e_o}, \tag{19}$$

where $\mathbf{K_o}$ is a gain matrix and $\mathbf{e_o}$ are the orientation errors defined through the unitary quaternion [11] and are calculated as follows:

$$\mathbf{e_o} = \begin{bmatrix} e_r \ \mathbf{e}_{\upsilon}{}^T \end{bmatrix} = \begin{bmatrix} r r_d + \boldsymbol{\upsilon}^T \boldsymbol{\upsilon}_d \\ r\boldsymbol{\upsilon}_d - r_d\boldsymbol{\upsilon} - \mathbf{S_o}(\boldsymbol{\upsilon})\boldsymbol{\upsilon}_d \end{bmatrix}, \tag{20}$$

where $(r, \boldsymbol{\upsilon})$ and $(r_d, \boldsymbol{\upsilon}_d)$ are the real and desired orientation expressed in quaternions [11], respectively, with the fulfillment that $\mathbf{e_o^T}\mathbf{e_o} = e_r^2 + \mathbf{e}_{\upsilon}{}^T\mathbf{e}_{\upsilon} = 1$, while $\mathbf{S_o}(\boldsymbol{\upsilon})$ is a skew-symmetric matrix defined as:

$$\mathbf{S_o}(\boldsymbol{\upsilon}) \triangleq \begin{bmatrix} 0 & -\upsilon_k & \upsilon_j \\ \upsilon_k & 0 & -\upsilon_i \\ -\upsilon_j & \upsilon_i & 0 \end{bmatrix}. \tag{21}$$

3 Results

The execution of a task is shown in this section to determine the performance of the proposal. For this, three points to be reached are defined, where ϵ is modified in real time at different periods of execution.

To reach the first point, the robot starts from a known position, with a default configuration (Figs. 2, 3). For the first 5 seconds, $\epsilon = 0$ is set, i.e., the platform is fully activated and the manipulator deactivated. In this time frame, position errors tending to zero are present for the X-Y axes (Fig. 4), while the error in Z remains constant since the participation of the MM is required to achieve zero errors. The range of $\mathbf{e_p} = 0$ is performed in the next 5 seconds of execution

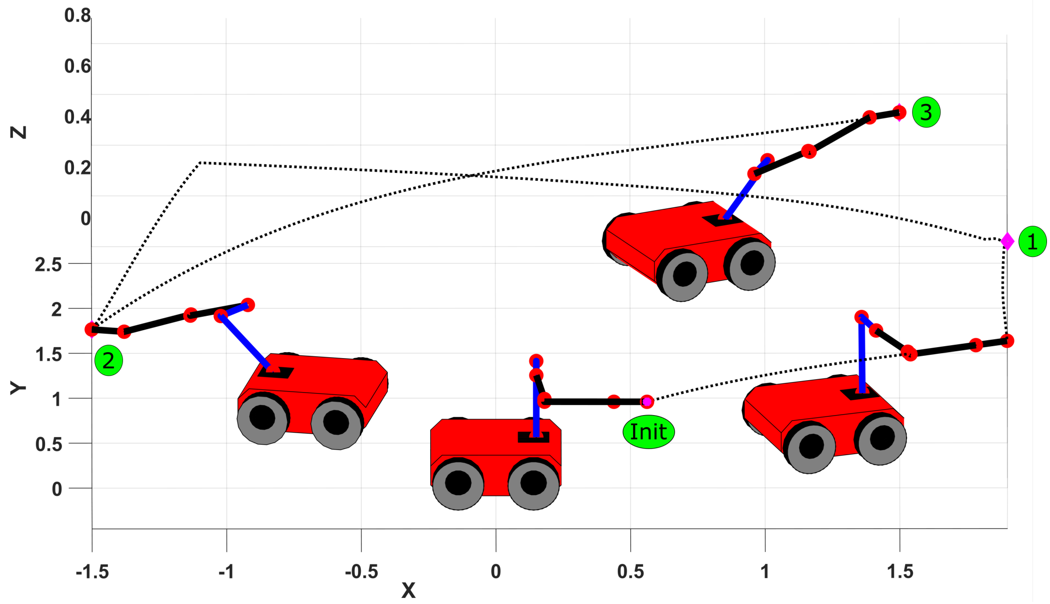

Fig. 2: Reconstruction of execution, lateral view

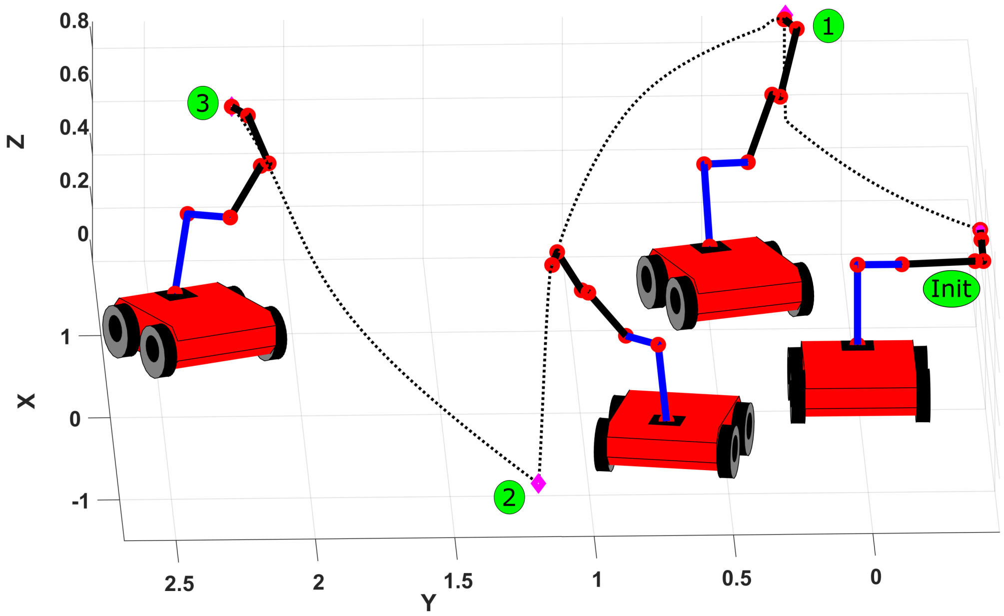

Fig. 3: Reconstruction of execution, front view

since $\epsilon = 1$ is set and the manipulator is activated, while the platform remains stationary. All this behavior of the robotic structure is shown in Figs. 5 and 6. These initial ten seconds show an approach of the platform to the desired point, after which it switches (practically unobtrusively) to the MM.

The second point range runs from seconds 10 to 20. Differently from the first point range, this period considers an involvement of both the platform and the manipulator in an indicated period. In the second ten, the new desired point generates position errors that the controller tries to resolve. From 10 to 12.3 seconds, ϵ enables only the moving platform, i.e. $\epsilon = 0$. Subsequently, the weighting value is $\epsilon = 0.6$, which enables velocities to be induced at all joints of the robot. In this period (at seconds 12.3 to 15), it can be observed how the platform moves backward (Fig. 5) to facilitate the MM reaching the target (Fig. 6), nulling the position errors in all axes (Fig. 4).

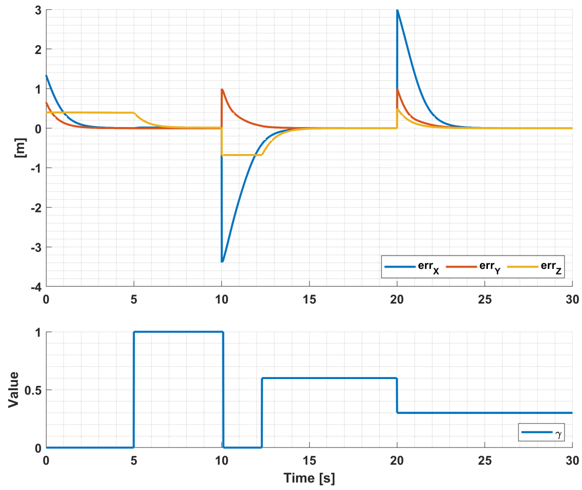

Fig. 4: Position errors

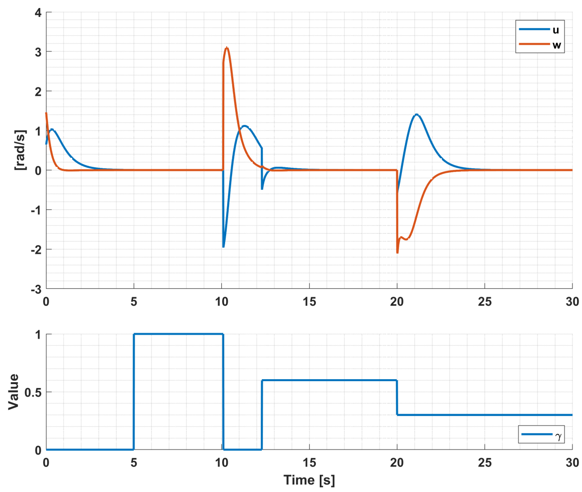

Fig. 5: Platform velocities

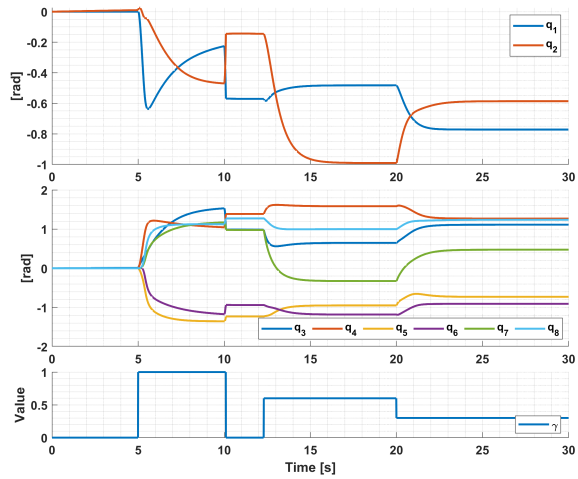

Fig. 6: Robotic manipulator positions

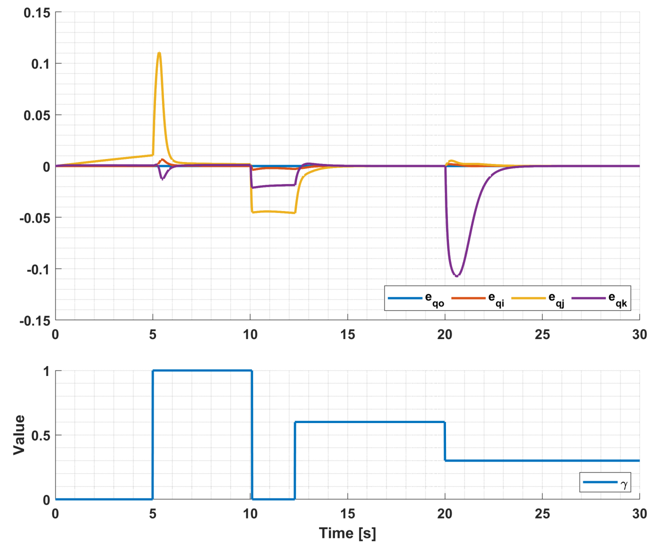

Fig. 7: Orientation errors

Finally, the range of the third point indicates a continuous involvement of both the moving platform and the manipulator. In this case, continuous motion of all joints (Figs. 5 and 6) and a trend of errors (Fig. 4) to zero similar in all axes is observed, given a configuration of $\epsilon = 0.4$.

For their part, quaternion-based orientation errors are shown in Fig. 7. Since the use of redundancy of the robotic system is posited, a desired orientation of the operating end is configured throughout the task. In this case and since this secondary objective is projected onto the null space of the solution, the position range may or may not be achieved. However, given the characteristics of the robotic system and as shown in Fig. 7, the errors tend to zero as long as ϵ allows the manipulator to be activated.

4 Conclusions

The main contribution of this work is the design of a weight matrix that includes cost functions for the continuous selection of sections of a highly redundant robot. In this way, it is proposed to control either isolated sections of a mobile manipulator robot, or by means of the weight matrix, to give higher execution priority to desired parts of the mechanism, based on a task-dependent variable that can be modified in real time.

The activation of parts of a redundant robotic system is possible through the use of a weighting matrix. As shown in the development of this proposal, the design of a matrix that allows to continuously switch sections of interest of the robot while deactivating others is possible. Additionally, since the switching factor can vary in real time, the application of such techniques can be useful for both autonomous and teleoperated control.

Acknowledgements The authors would like to thank the German Student Exchange Service (DAAD), and the Instituto de Automática, a dual-dependent institute: Universidad Nacional de San Juan and CONICET - Argentina.

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. T. Sandakalum and M. H. Ang Jr, "Motion planning for mobile manipulators—a systematic review," *Machines*, vol. 10, no. 2, p. 97, 2022.
2. E. Slawinski, D. Santiago, and V. Mut, "Dual coordination for bilateral teleoperation of a mobile robot with time varying delay," *IEEE Latin America Transactions*, vol. 18, no. 10, pp. 1777–1784, 2020.
3. H. Xing, Z. Gong, L. Ding, A. Torabi, J. Chen, H. Gao, and M. Tavakoli, "An adaptive multi-objective motion distribution framework for wheeled mobile manipulators via null-space exploration," *Mechatronics*, vol. 90, p. 102949, 2023.
4. H. Xing, L. Ding, H. Gao, W. Li, and M. Tavakoli, "Dual-user haptic teleoperation of complementary motions of a redundant wheeled mobile manipulator considering task priority," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 10, pp. 6283–6295, 2022.
5. S. Huang, J. Xiang, W. Wei, and M. Z. Chen, "On the virtual joints for kinematic control of redundant manipulators with multiple constraints," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 65–76, 2017.
6. B. Dariush, Y. Zhu, A. Arumbakkam, and K. Fujimura, "Constrained closed loop inverse kinematics," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2499–2506.
7. P. I. Corke, "A simple and systematic approach to assigning denavit–hartenberg parameters," *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 590–594, 2007.
8. K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *International journal of applied mathematics and computer science*, vol. 14, no. 4, pp. 477–496, 2004.
9. J. Moreno, E. Slawiñski, F. A. Chicaiza, F. G. Rossomando, V. Mut, and M. A. Morán, "Design and analysis of an input–output linearization-based trajectory tracking controller for skid-steering mobile robots," *Machines*, vol. 11, no. 11, p. 988, 2023.
10. Y. Wei, "Motion planning and tracking of a hyper redundant non-holonomic mobile dual-arm manipulator," Ph.D. dissertation, Ecole Centrale de Lille, 2018.
11. B. Xian, M. de Queiroz, D. Dawson, and I. Walker, "Task-space tracking control of robot manipulators via quaternion feedback," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 160–167, 2004.

License