2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Cartesian Impedance Control Generalized to One-Parameter Splines

Ignacio Montesino[1], Juan G. Victores[1], Carlos Balaguer[1], and Alberto Jardon[1]

***Abstract*— Robotic-assisted upper limb rehabilitation has gained significant attention in recent years due to its potential to enhance the recovery process for individuals with motor impairments resulting from neurological conditions and injuries. The main rehabilitation treatments rely on the repetitive execution of a movement of the upper-limb, guided by a therapist to prevent incorrect movements and to provide the necessary support. Many of the exercises performed by therapists can be modeled as a movement in SE(3) space (position and orientation). This movement itself is one-dimensional, as it can be modeled by a one-dimensional curve. To solve a similar problem, some approaches have been proposed in human-robot interaction (HRI) following virtual guides, but are either limited to specific types of curves (e.g. without orientation) or rely on linear control methods with non-intuitive parameters. To address these limitations and enable the use of these methods in physical rehabilitation, this paper extends Cartesian impedance control to splines, which we term path impedance control. It capitalizes on the intrinsic path geometry of end-effector robotic rehabilitation systems. The primary objective of this control algorithm is to emulate the sensation of maneuvering a physical object along a wire, akin to conventional exercise machines; and, in conjunction, provide an intuitive parametrization of rehabilitation exercises. We build on existing virtual guide control strategies using non-linear control and Lie Groups to generalize the control law to any one-parameter SE(3) curve.**

I. INTRODUCTION

In the developed world, strokes stand as a leading cause of disability [1], severely limiting physical capabilities in affected individuals and significantly impacting their quality of life and that of their families [2]. Research consistently favors robot-aided rehabilitation over manual methods, showcasing its ability to enable high-intensity, repetitive, and task-specific training regimens [3]. Two main types of upper limb rehabilitation robots have been developed: end-effector and exoskeleton, with the former demonstrating superior outcomes in promoting active patient engagement [4], [5].

The main challenge in upper-limb rehabilitation lies in providing both active and passive repetitive exercise capabilities [6]. Assist-as-needed (AAN) controllers are tailored to provide support or correction during rehabilitation training tasks based on the patient's functional ability [7], [8], [9]. Various approaches, including impedance control, are used to adaptively assist individuals [10], [11]. Other methods involve using the healthy arm to guide the affected arm or employing learning-based algorithms to model exercise dynamics [12], [13]. Passive exercise utilizes virtual guides to maintain the desired path [14].

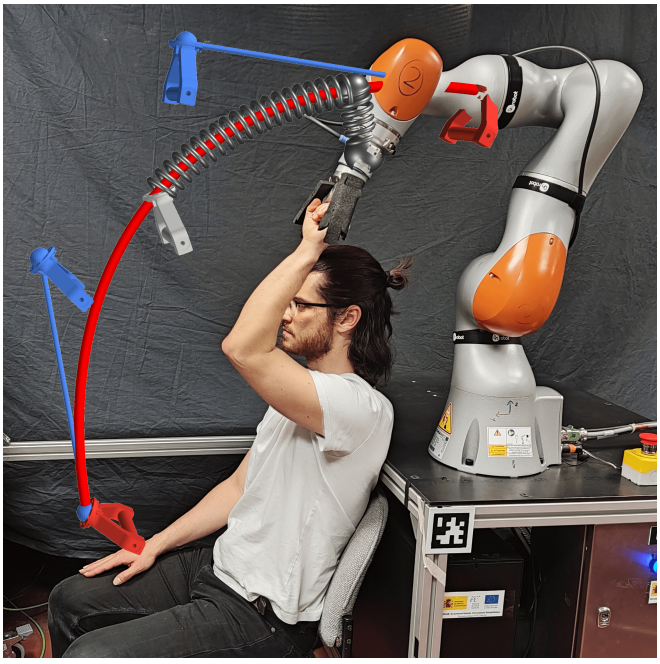

Fig. 1. Visualization of the path impedance control virtual springs, as the patient moves the robot along during therapy.

These control schemes often lack intuitive parameters for physiotherapists, who commonly describe desired robot behaviors geometrically [15]. Our work aims to adapt Cartesian impedance control to any curve, characterizing stiffness and damping along and perpendicular to it (see Figure 1).

For this, we use the virtual guides from HRI [16], and enhance their capabilities by extending Cartesian impedance control to any virtual guide defined as a Lie Group one parameter spline. With this approach, the robot is restricted by a virtual guide that moves along a spline with a given stiffness and damping, which we will call path impedance. These parameters are physically meaningful and can be easily adjusted by the therapist.

The main contributions of the presented work are:

- Extending virtual guides to 6 dimensional rigid body motion, using Lie Group Bézier curves.
- Generalizing Cartesian impedance to parametric space with coefficients in parameter and normal space.
- Validation of the theoretical results in a real robotic

This research has received funding from ROBOASSET, "Sistemas robóticos inteligentes de diagnóstico y rehabilitación de terapias de miembro superior", PID2020-113508RB-I00 financed by AGENCIA ESTATAL DE INVESTIGACION (AEI), the IROPER R&D&I project PLEC2021-007819 funded by MCIN/AEI/10.13039/501100011033 and by the European Union NextGenerationEU/PRTR, and from the i-REHAB project "AI-powered Robotic Personalized Rehabilitation, ISCIII-AES-2022/003041, funded by Instituto de Salud Carlos III (ISCIII) and cofund by European Union".

[1]All researchers are part of RoboticsLab, Carlos III University of Madrid, 28911 Leganés, Spain imontesi@ing.uc3m.es

platform, a KUKA LBR 14, under perturbations.

II. DEFINING LIE GROUP CURVES FOR THE VIRTUAL GUIDES

To define a curve in space, we need a parametric curve. Many such curves exist, for this work we will use Bézier curves, as they are easily defined in $\mathbb{R}^3$ and $SO(3)$, and are standard in computer graphics and robotics. This also opens many possibilities in VR and AR assisted rehabilitation [17]. The required properties of said curve are smoothness and the ability to define a frame along the curve.

A. Defining Bézier curves

Bézier curves are parametric curves defined by a set of control points. Specifically, cubic Bézier curves are defined by four control points $\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ and allow for control of the beginning and end points of the curve as well as the tangent direction of the curve at those points.

From these control points, Bézier curves are defined via the DeCastejau algorithm. The algorithm consists of recursively interpolating between the control points using the parameter t. B_n^i denotes the i-th point in the n-th recursion.

$$B_i^0 = p_i \quad p_i \in \mathbb{R}^3 \tag{1}$$

$$B_i^n(t) = B_{i+1}^{n-1}t + B_i^{n-1}(1-t) \tag{2}$$

$$\mathbf{B}(t) = B_0^3 \tag{3}$$

In $\mathbb{R}^3$, Bézier curves can also be computed using Bernstein polynomials in the form

$$\mathbf{B}(t) = \sum_{i=0}^{3} \mathbf{P}_i B_i^3(t), \quad t \in [0,1] \tag{4}$$

Where $B_i^3(t)$ are the Bernstein polynomials of degree 3.

In addition, the derivatives of the curve can also be computed using Bernstein polynomials as

$$\mathbf{B}'(t) = 3\sum_{i=0}^{2}(\mathbf{P}_{i+1} - \mathbf{P}_i)B_i^2(t), \quad t \in [0,1] \tag{5}$$

$$\mathbf{B}''(t) = 6\sum_{i=0}^{1}(\mathbf{P}_{i+2} - 2\mathbf{P}_{i+1} + \mathbf{P}_i)B_i^1(t) \tag{6}$$

$$\tag{7}$$

These terms can be precomputed, resulting in a very efficient algorithm for computing the curve and its derivatives.

B. Lie Group Bézier curves

Bézier curves can also be extended to Riemannian manifolds and Lie Groups, as shown in [18]. Since the path of the rehabilitation motion is defined as a motion in $\mathbf{SE}(3)$, we will use the extension of Bézier curves to Lie Groups, using an extension of the DeCasteljau algorithm:

$$B_i^0 = g_i \quad g_i \in \mathbf{SE}(3) \tag{8}$$

$$B_i^n(t) = B_{i+1}^{n-1} \cdot \exp(\log((B_{i+1}^{n-1})^{-1}B_{n-1}) \cdot t) \tag{9}$$

$$\mathbf{B}(t) = B_0^3 \tag{10}$$

For this, we will use curves in $\mathbf{SE}(3)$, which are defined by a set of control points g_0, g_1, g_2, g_3 defined as rigid body positions of the end effector.

Finally, we need a means of obtaining a frame in to define the Cartesian impedance on. A standard way of doing this is using the Frenet-Serret formulas. Since we will be tracking the position of the mechanism in R^3, these formulas need only be applied to the position part of the curve.

C. Frenet-Serret Formulas

Using equations 5 and 6 for the derivatives of the Bézier curve we can obtain the frame $F(s) = \{T(s), N(s), B(s)\}$ at any point.

$$T(s) = \frac{\mathbf{B}'(s)}{\|\mathbf{B}'(s)\|} \tag{11}$$

$$N(s) = \frac{\mathbf{B}''(s)}{\|\mathbf{B}''(s)\|} \tag{12}$$

$$B(s) = T(s) \times N(s) \tag{13}$$

The direction of the normal is not stable around at points where the curvature is zero, but since we only care about the perpendicular plane, not any specific direction in it, we can take any perpendicular vector at those points.

III. COMPUTING THE VIRTUAL GUIDE POSITION ON THE CURVE

In HRI, many approaches have been developed over the years to help human operators to follow a path [19]. One especially interesting for rehabilitation is the virtual guide approach [16]. In this approach a mechanical systems dynamics are simulated in real-time to constrain the robot to a desired path. We will now see why this provides us with two main advantages.

In order to find the point on the curve from which to compute the frame we could opt to select the closest point to the robot's end-effector. But naively searching for the closest point has two main drawbacks. The first is the analytical complexity of finding the solution to the optimization problem:

$$t_r = \min_t(\|\mathbf{B}(t) - P\|) \quad t \in [0,1] \tag{14}$$

Which for a cubic Bézier results in a polynomial of degree five, meaning there is no general solution. The second issue is a geometrical one, suppose we are close to a "U-turn" in the curve, as shown in Figure 2.

A small change in the position of the robot's end-effector will result in the closest point jumping from one side of the curve to the other. For multiple reasons (stability, skipping sections, etc.) this behavior is unacceptable.

Our objective is to simulate the behavior of moving a physical object along a wire. We can therefore simulate the object as a particle with mass m and position along the curve u, as shown in Figure 3.

We can connect the particle to the robot's end-effector using a spring with stiffness k and damping b. Then find the

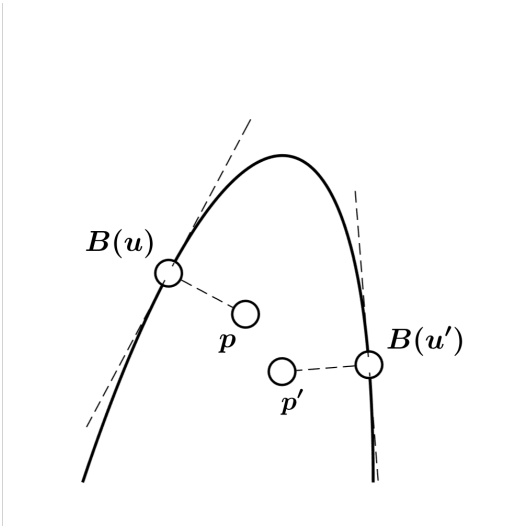

Fig. 2. A small movement in the robot's end-effector (p to p') causes the closest point $B(u)$ to jump from one side of the curve to the other $B(u')$.

force f to apply to this particle by projecting the distance between it and the end-effector onto the tangent vector of the curve.

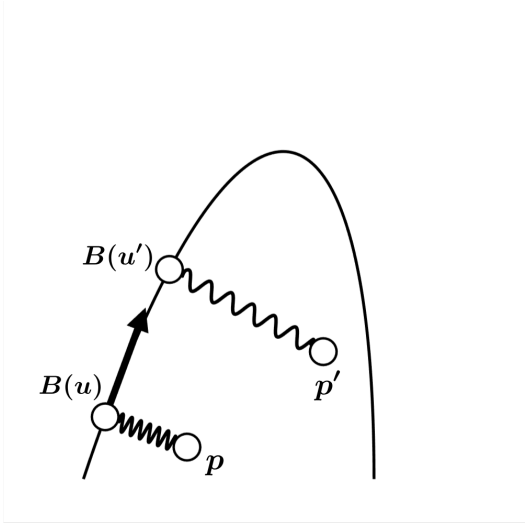

Fig. 3. A virtual particle moves along the curve as it is pulled tangentially by a virtual spring-damper system connected to the robot's end-effector.

This virtual particle will be defined by its state $\mathbf{S} = [u, \dot{u}]^T$, where u is the position of the particle along the curve in parameter space and $\dot{u}$ its velocity. We can iteratively update its state with the following equation:

$$\mathbf{S}' = \mathbf{S} + \Delta t \begin{pmatrix} \dot{u} \\ \ddot{u} \end{pmatrix} \tag{15}$$

Where $\ddot{u}$ is the acceleration of the particle obtained from the spring force:

$$\ddot{u} = \frac{k e_T - b\dot{u}}{m} \tag{16}$$

Where e_T is the projection of the distance between the particle and the end-effector onto the tangent vector of the curve at u.

This updating of the particle state can be done in tandem with the control law for the robot's end-effector.

The details of selecting the parameters m, k and b and updating the position of the particle are explained in the Section IV.

IV. CARTESIAN IMPEDANCE CONTROL GENERALIZED TO SPLINE

Being now equipped with a robot-following frame free of discontinuities, we can proceed to implement path impedance control. Regular Cartesian impedance consist of defining a control law such that the relation between displacements in the end-effector and the forces applied to it is given by the following equation:

$$\mathbf{F} = \mathbf{\Lambda_d}\ddot{\mathbf{x}} + \mathbf{D_d}\dot{\mathbf{x}} + \mathbf{K_d}\mathbf{x} \tag{17}$$

Where $\mathbf{F}$ is a six dimensional wrench containing the force and torque applied to the end-effector, $\mathbf{\Lambda_d}$ is the desired perceived inertia matrix of the end-effector $\mathbf{D_d}$ is the desired damping and $\mathbf{K_d}$ is the stiffness. $\mathbf{x}$ is the pose of the end-effector with respect to the equilibrium position.

The manipulator dynamics can be obtained by Lagrangian mechanics and are of the form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \tau_c + \mathbf{J}^T\mathbf{F}_{ext} \tag{18}$$

Where $\mathbf{M}(\mathbf{q})$ is the joint-space inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the Coriolis and centrifugal forces, $\mathbf{g}$ is the gravity forces and τ_c is the vector of commanded joint torques, all expressed in joint-space. $\mathbf{J}$ is the Jacobian of the end-effector and $\mathbf{F}_{ext}$ is the external wrench applied to the end-effector.

In [20] a method for obtaining a stable impedance parameters is described. By setting $\mathbf{\Lambda_d} = (\mathbf{J}^{-1})^T\mathbf{M}\mathbf{J}^{-1}$, the control law will no longer depend on the external forces, since they will produce the correct accelerations for our desired mass.

Since force rehabilitation movements are made at slow speeds, we can neglect the Coriolis forces. We can also negate the gravity forces by adding $-\mathbf{g}(\mathbf{q})$ to our control law.

The desired stiffness matrix is designed to be diagonal, with the diagonal elements $k_x, k_y, k_z, k_{\theta_x}, k_{\theta_y}, k_{\theta_z}$. Finally, for the design of the damping matrix we use the formulation in [21], which allows the user to specify a damping ratio from 0 meaning undamped to 1 meaning critically damped. For an in depth explanation of the damping matrix design see [21].

The problem of path impedance can be seen as Cartesian impedance in a frame that is moving along a path. We can obtain a change of frame matrix T by computing the frame $F(s)$ at virtual particle position.

$$T = \begin{bmatrix} F(s) & 0 \\ 0 & F(s) \end{bmatrix} \tag{19}$$

and its inverse:

$$T^{-1} = \begin{bmatrix} F(s)^{-1} & 0 \\ 0 & F(s)^{-1} \end{bmatrix} \tag{20}$$

Here 0 is a 3×3 zero matrix.

We can then define the desired inertia matrix in the new frame as:

$$\mathbf{J}' = T^{-1}\mathbf{J} \tag{21}$$

$$\mathbf{\Lambda_d}' = (\mathbf{J}')^{-T}\mathbf{M}\mathbf{J}'^{-1} \tag{22}$$

In this new frame our desired stiffness matrix is:

$$\mathbf{K_d}' = \begin{pmatrix} k_T & 0 & 0 & 0 & 0 & 0 \\ 0 & k_N & 0 & 0 & 0 & 0 \\ 0 & 0 & k_N & 0 & 0 & 0 \\ 0 & 0 & 0 & k_R & 0 & 0 \\ 0 & 0 & 0 & 0 & k_R & 0 \\ 0 & 0 & 0 & 0 & 0 & k_R \end{pmatrix} \tag{23}$$

We are forced to set a single stiffness value for both perpendicular directions, and a single one for the tangent direction. This is due to the fact that the frame $F(s)$ has no guaranteed normal direction at points where the curvature is zero, as seen in subsection II-C.

The damping matrix $\mathbf{D_d}'$ is obtained in an identical fashion to the Cartesian impedance case, but using the new inertia matrix $\mathbf{\Lambda_d}'$ and the new stiffness matrix $\mathbf{K_d}'$.

The new control law is then:

$$\mathbf{F}' = \mathbf{\Lambda_d}'\ddot{\mathbf{x}}' + \mathbf{D_d}'\dot{\mathbf{x}}' + \mathbf{K_d}'\mathbf{x}' \tag{24}$$

But now, if we simply define $\mathbf{x}' = T^{-1}\mathbf{x}$, we will obtain the distance between the end-effector and the virtual particle in the new frame. To obtain a geometrically meaningful control law, we need to define a more useful $\mathbf{x}'$. Figure 4 illustrates the desired behavior. Where $B(t)$ is the reference point on the path, p is the current position of the end-effector and $B(u)$ is the position of the virtual particle along the path.

The definition of $\mathbf{x}'$ is then:

$$\mathbf{x}'_p = \begin{pmatrix} L \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + F(s)^{-1}(B(u) - p) \tag{25}$$

Here B refers to the position part of the curve, since the curve is defined as a motion in $\mathbf{SE}(3)$. Note that since $(B_p(u) - p)$ is perpendicular to the first component of the frame $F(s)$ (the tangential component), the first component of $F(s)^{-1}(B(u) - p)$ is 0.

To give some intuition on the behavior of the control law, imagine holding the end-effector in place. The virtual particle follows the robot's end-effector along the path, and since the end-effector is not moving, it remains in place. The wrench felt by the end-effector is an attraction force towards the path, at the point of the virtual particle. Only the tangential component, dependent on L, grows as the reference point $B(t)$ moves along the path.

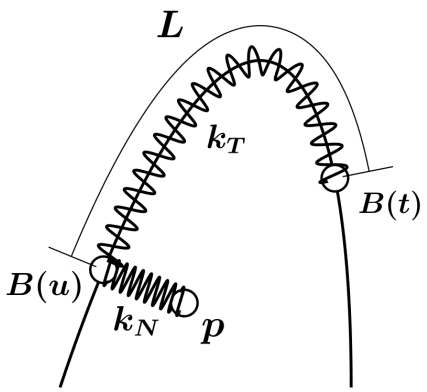

Fig. 4. Path impedance control behaves as a spring damper perpendicular to the curve and one along it.

We have not given a definition for L. The natural choice would be the arc-length from $B(u)$ to $B(t)$, but this would require computing the arc-length of the curve at every time step. For Bézier curves, this is a very expensive operation, since it requires computing the derivative of the curve and then integrating it.

In fact, to obtain a passive controller, we do not need to explicitly compute the arc-length. We can set the tangential force to be proportional to the velocity along the curve $B(u)$ with parameter D_t. Thus obtaining a passive viscous friction force.

With this in mind, we use the parameter distance $(u - t)L_B$, multiplied by an approximation of the length of the curve L_B. This means that L will not grow uniformly as t increases, but we believe this is a design choice to be made for each specific application and parametric curve type. The only remaining that of choosing the parameters k, b and m for the virtual particle we had postponed in Section III.

For Bézier curves, we have found that a mass of $m = 0.05kg$ and a stiffness of $k = 2000N/m$ with a damping ratio of $\xi_D = 0.7$, $d = 2\xi_D m\sqrt{\frac{k}{m}}$ works well for all cases. Any cubic Bézier curve is parametrized by four control points, and the parameter space is always $[0, 1]$, so the complexity of the curve is limited, and the parameter space does not change in magnitude.

To simulate the particle movement, each time we command the robot (500Hz) we also update the position of the virtual particle. This is fast enough for the dynamics of the virtual mechanism to be well-behaved.

V. EXPERIMENTAL SETUP

To validate the stability of our control law, a set of experiments have been performed in simulation, and then on the real robot. For validation on the real robot, we have used a KUKA LBR iiwa 14 R820, as shown in Figure 6.

The curve selected as a reference path for the experiment is a Bézier curve of degree 3 with control points shown in Eq. 9. Each control is a 4×4 matrix representing an element of $SE(3)$. These four control points define the curve shown in Figure 5.

$$g_0 = \begin{pmatrix} -1.00 & -0.00 & 0.00 & 0.53 \\ -0.00 & 1.00 & 0.00 & -0.00 \\ -0.00 & 0.00 & -1.00 & 0.31 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

$$g_1 = \begin{pmatrix} -0.95 & -0.00 & -0.31 & 0.53 \\ 0.05 & 0.99 & -0.15 & 0.05 \\ 0.30 & -0.16 & -0.94 & 0.51 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

$$g_2 = \begin{pmatrix} -0.99 & -0.00 & -0.16 & 0.53 \\ 0.04 & 0.97 & -0.23 & 0.05 \\ 0.15 & -0.23 & -0.96 & 0.11 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

$$g_3 = \begin{pmatrix} -1.00 & -0.00 & 0.00 & 0.63 \\ -0.00 & 1.00 & 0.00 & 0.20 \\ -0.00 & 0.00 & -1.00 & 0.31 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{pmatrix}$$

It defines a path around the user with a rotation of the wrist to simulate a rehabilitation exercise. We can see the evolution of both position and orientation. In Figure 6, we can see a virtual representation of the virtual mechanism during interaction with the patient. Where the forces acting on the end-effector are represented as springs.

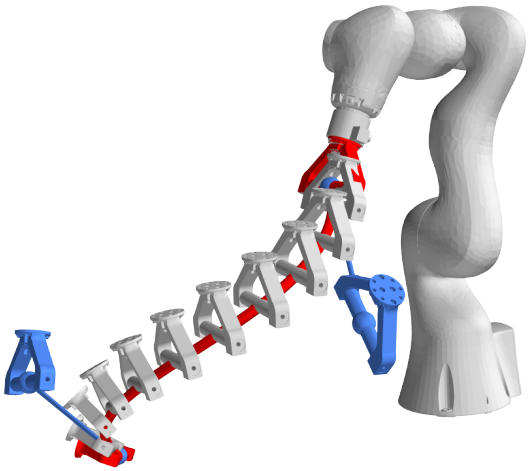

Fig. 5. Bézier curve of degree 3 with control points g_0, g_1, g_2 and g_3. The start and end poses are shown in red and the control poses in blue.

Fig. 6. The virtual mechanism pulls the user perpendicularly towards the curve (yellow) and tangentially to the target (gray). In rotation space, it corrects the orientation to that of the closest point (yellow).

The performed experiments have been the following:

1) Simulated perturbations to compare the control law with traditional impedance control.
2) The control law tested on the real robot with human robot interaction and perturbations.

VI. EXPERIMENTAL RESULTS

In this section, we present the results of the experiments as described in section V. First, the perturbation analysis in the simulated environment.

A. Comparison of Cartesian and the presented path impedance control under simulated perturbations

In the following three-dimensional graph in Figure 7, we see the effect of two perturbations at two different times. Three different results from control methods are shown: traditional Cartesian control at two different stiffness values (high at $1500N/m$, and low at $200N/m$), as well as path impedance control with a tangential stiffness of $200N/m$ and a normal stiffness of $1500N/m$.

The first perturbation is at a 45° angle to the curve, and the second is perpendicular to the curve. At first perturbation, and we see that the presented path impedance control recovers as quick as high impedance control, as its normal stiffness is the same. Since we have stayed perpendicular to the curve, the closest point has not changed and the rotation is not affected as seen in Figure 8.

At the second perturbation, we see that the maximum distance from the curve is the same for both high impedance

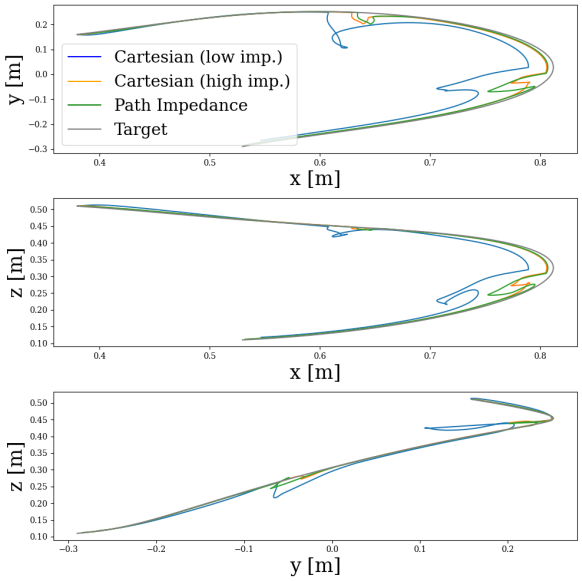

Fig. 7. Effects of perturbations on position controlled by the different algorithms. Diedric perspectives of the 3D plot.

and path impedance control. path impedance control however stretches farther along the tangent direction due to its lower stiffness in that direction, with a profile more similar to the low impedance control. In rotation, path impedance control has a bigger variation as it tracks the rotation at the closest point, not at the target.

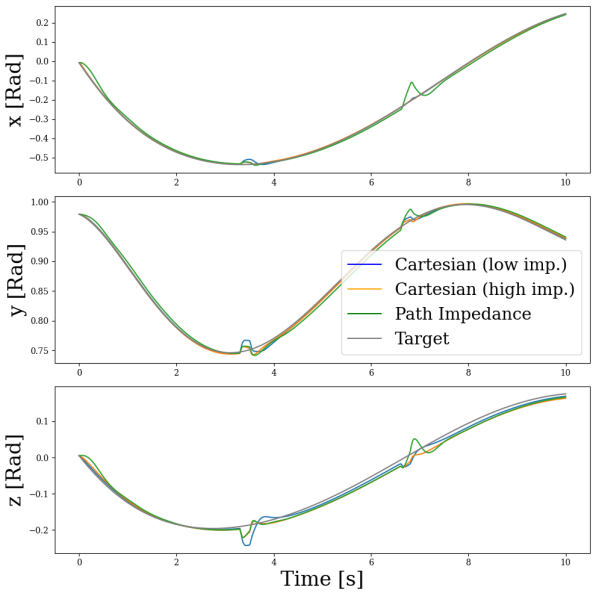

Fig. 8. Perturbations effects on quaternion components of rotation, controlled by the different algorithms.

B. Testing on a real robot

We perform a test on a real robot using the spline seen in Figure 6 forwards and backwards. Since this is a real interaction with a human, we set the Cartesian impedance control to a low stiffness of $200N/m$. This mimics real exercises where we have to strictly follow the path, but we cannot exert excessive force along the path where the damaged muscles are being activated.

Figure 9 shows the results of the experiment. While low impedance Cartesian control does not follow the path, path impedance control follows the path closely. This a more subjective test as user interaction is not easily repeatable. But nonetheless it shows the robustness of the control law to perturbations, while maintaining low stiffness where needed.

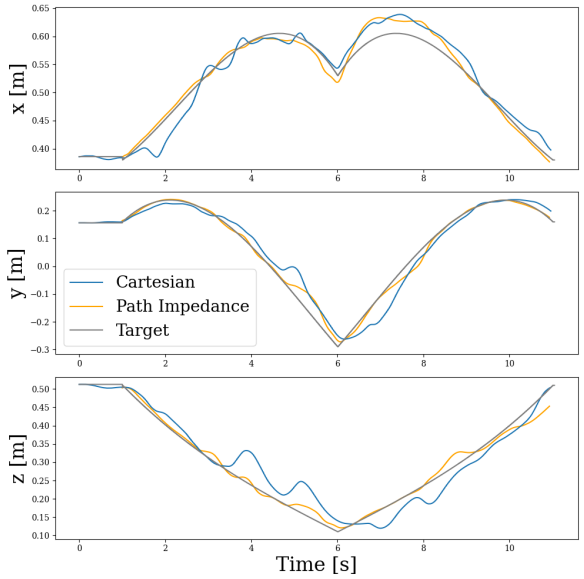

Fig. 9. Performance of the path-based impedance control on the real robot.

If instead of helping the user reach a target along a path we wanted to instead oppose the user's movement to engage in active rehabilitation, we could simply set the stiffness of the tangential impedance to zero, and substitute it with a velocity damping term.

VII. CONCLUSION AND FUTURE WORK

In this work, we have presented a novel control law, path impedance, for human-robot interaction in rehabilitation robotics, one that is based on the geometric characteristics of the path of the rehabilitation motion. By extending it to generic parametric curves in $\mathbf{SE}(3)$ we hope to open the possibility for other curves (e.g. Pythagorean hodograph curves) and other applications in VR and AR assisted rehabilitation.

We have shown that our control law is robust to perturbations and that it is able to maintain the end effector on the path as well as high impedance control. As well as allowing more freedom of movement along the path with low impedance.

In the future, we would like to further develop more types of path based control laws. For example, the one required by isokinetic exercises, where the robot sets a maximum velocity along the path no matter how much force is exerted by the user.

Funding

The research leading to these results has received funding from the Spanish Ministry of Economy and Competitiveness as part of the project: "ROBOASSET: Intelligent robotic systems for assessment and rehabilitation in upper limb therapies" (PID2020-113508RB-I00), funded by AEI/10.13039/501100011033,

The research leading to these results has also received funding from and by the project: "iREHAB: AI-powered Robotic Personalized Rehabilitation", (ISCIII-AES-2022/003041), funded by Instituto de Salud Carlos III (ISCIII) and cofunded by EU.

Also we have received the support of the RoboCity2030-DIH-CM Madrid Robotics Digital Innovation Hub ("Robótica aplicada a la mejora de la calidad de vida de los ciudadanos. fase IV"; S2018/NMT-4331), funded by "Programas de Actividades I+D en la Comunidad de Madrid" and cofunded by Structural Funds of the EU.

References

[1] Davide G. de Sousa, Lisa A. Harvey, Simone Dorsch, and Joanne V. Glinsky. Interventions involving repetitive practice improve strength after stroke: A systematic review. *Journal of Physiotherapy*, 64(4):210–221, October 2018.

[2] Mohammad Najafi, Mojtaba Sharifi, Kim Adams, and Mahdi Tavakoli. Robotic assistance for children with cerebral palsy based on learning from tele-cooperative demonstration. *International Journal of Intelligent Robotics and Applications*, 1(1):43–54, February 2017.

[3] Anil Babu Payedimarri, Matteo Ratti, Riccardo Rescinito, Kris Vanhaecht, and Massimiliano Panella. Effectiveness of Platform-Based Robot-Assisted Rehabilitation for Musculoskeletal or Neurologic Injuries: A Systematic Review. *Bioengineering*, 9(4):129, March 2022.

[4] Paweł Maciejasz, Jörg Eschweiler, Kurt Gerlach-Hahn, Arne Jansen-Troy, and Steffen Leonhardt. A survey on robotic devices for upper limb rehabilitation. *Journal of Neuroengineering and Rehabilitation*, 11:3, January 2014.

[5] Hassan M. Qassim and W. Z. Wan Hasan. A Review on Upper Limb Rehabilitation Robots. *Applied Sciences*, 10(19):6976, January 2020.

[6] Kyung Eun Lee, Muncheong Choi, and Bogja Jeoung. Effectiveness of Rehabilitation Exercise in Improving Physical Function of Stroke Patients: A Systematic Review. *International Journal of Environmental Research and Public Health*, 19(19):12739, October 2022.

[7] Pengyu Zhang, Jie Zhang, and Jiangming Kan. A Research on Manipulator-Path Tracking Based on Deep Reinforcement Learning. *Applied Sciences*, 13(13):7867, January 2023.

[8] Eric T. Wolbrecht, Vicky Chan, David J. Reinkensmeyer, and James E. Bobrow. Optimizing compliant, model-based robotic assistance to promote neurorehabilitation. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, 16(3):286–297, June 2008.

[9] Leigang Zhang, Shuai Guo, and Qing Sun. An Assist-as-Needed Controller for Passive, Assistant, Active, and Resistive Robot-Aided Rehabilitation Training of the Upper Extremity. *Applied Sciences*, 11, December 2020.

[10] Brahim Brahmi, Maarouf Saad, Cristobal Ochoa Luna, Philippe S. Archambault, and Mohammad H. Rahman. Passive and active rehabilitation control of human upper-limb exoskeleton robot with dynamic uncertainties. *Robotica*, 36(11):1757–1779, November 2018.

[11] Brahim Brahmi, Mohamed Hamza Laraki, Maarouf Saad, M. H. Rahman, Cristobal Ochoa-Luna, and Abdelkrim Brahmi. Compliant adaptive control of human upper-limb exoskeleton robot with unknown dynamics based on a Modified Function Approximation Technique (MFAT). *Robotics and Autonomous Systems*, 117:92–102, July 2019.

[12] Lufeng Chen, Jing Qiu, Xuan Zou, and Hong Cheng. A Dual-Arm Participated Human-Robot Collaboration Method for Upper Limb Rehabilitation of Hemiplegic Patients. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12617–12623, May 2023.

[13] Kamran Maqsood, Jing Luo, Chenguang Yang, Qingyuan Ren, and Yanan Li. Iterative learning-based path control for robot-assisted upper-limb rehabilitation. *Neural Computing and Applications*, May 2021.

[14] X. Zhang, A. Behal, D.M. Dawson, and J. Chen. A novel passive path following controller for a rehabilitation robot. In *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, volume 5, pages 5374–5379 Vol.5, December 2004.

[15] Jon Arrizabalaga and Markus Ryll. Pose-Following with Dual Quaternions, August 2023.

[16] Luc Joly, L.D. Joly, Claude Andriot, and C. Andriot. Imposing motion constraints to a force reflecting telerobot through real-time simulation of a virtual mechanism. *null*, 1995.

[17] Ignacio Montesino, Juan G Victores, Carlos Balaguer, and Alberto Jardon. Extending Piecewise Bézier Fitting Methods to SO(3) and SE(3) for Robot Paths. Workshop on Geometric Representations at ICRA 2024, June 2024.

[18] F. C. Park and B. Ravani. Bézier Curves on Riemannian Manifolds and Lie Groups with Kinematics Applications. *Journal of Mechanical Design*, 117(1):36–40, March 1995.

[19] Stuart A Bowyer, Stuart A. Bowyer, Brian Davies, Brian L. Davies, Brian L. Davies, Ferdinando Rodriguez y Baena, and Ferdinando Rodriguez y Baena. Active constraints/virtual fixtures: A survey. *IEEE Transactions on Robotics*, 2014.

[20] Christian Ott. Cartesian Impedance Control: The Rigid Body Case. In Christian Ott, editor, *Cartesian Impedance Control of Redundant and Flexible-Joint Robots*, Springer Tracts in Advanced Robotics, pages 29–44. Springer, Berlin, Heidelberg, 2008.

[21] A. Albu-Schaffer, C. Ott, U. Frese, and G. Hirzinger. Cartesian impedance control of redundant robots: Recent results with the DLR-light-weight-arms. In *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, volume 3, pages 3704–3709 vol.3, September 2003.