Towards Impedance control on Curves for Upper-Limb Rehabilitation

I. Montesino, J.G. Victores, A. Jardon

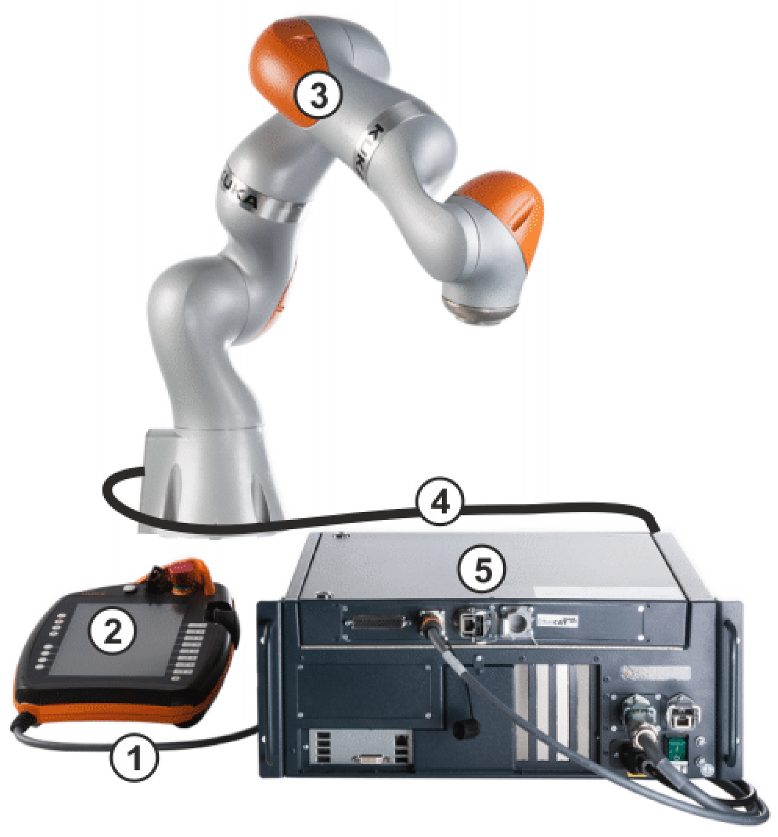
Universidad Carlos III de Madrid

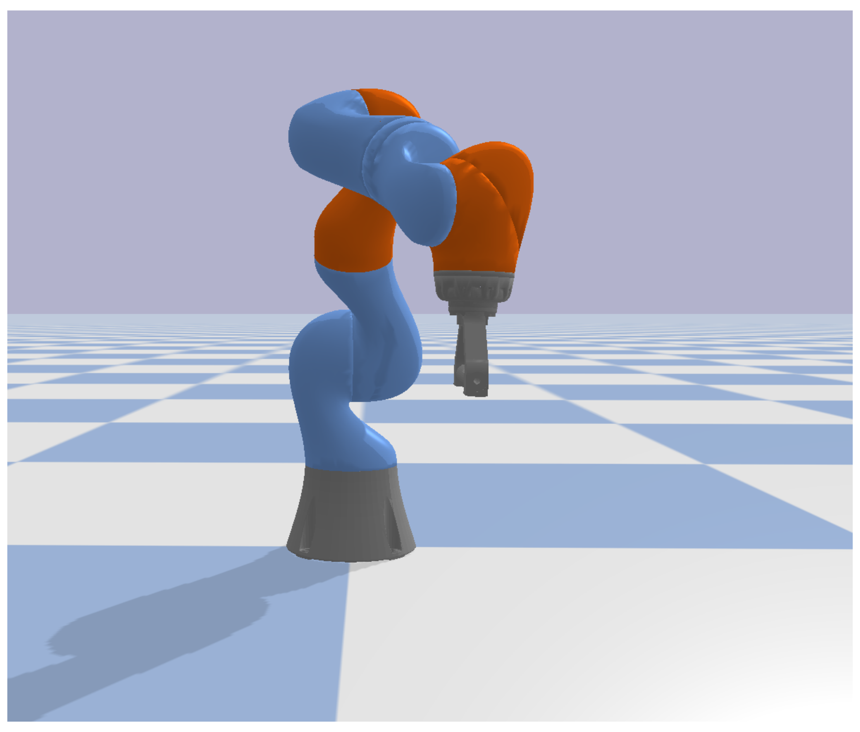
Introduction

Robotic rehabilitation is on the rise as physiotherapists require tools to accurately measure diffrent ailments. Some main challenges include robustness of the solutions to greater effort ranges, flexibility, repeatability and easy individualization of the treatment.
To overcome these challenges, three main challenges must be overcome:

- A robust and capable robotic platform must be used. Capable of fast adaptation to changes in requirements and ensured safety.
- A control method that mimics a human therapists' movement.
- A trajectory capture system that is both precise and robust to perturbations.

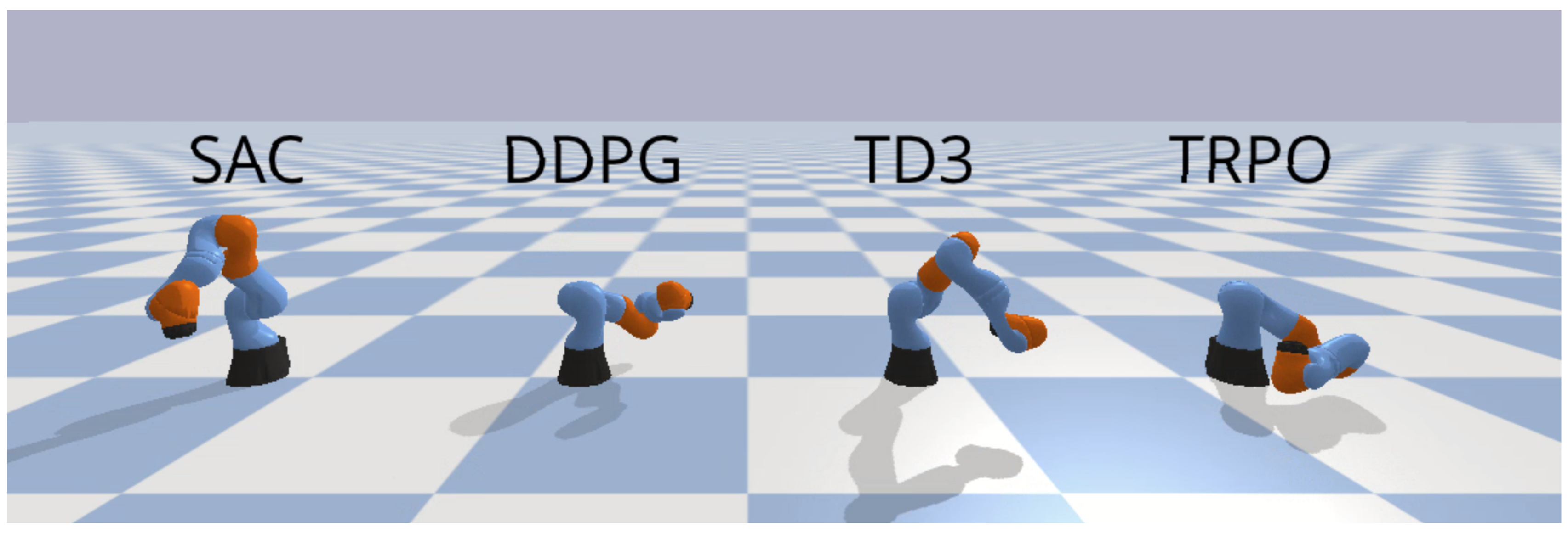
Robotic platform





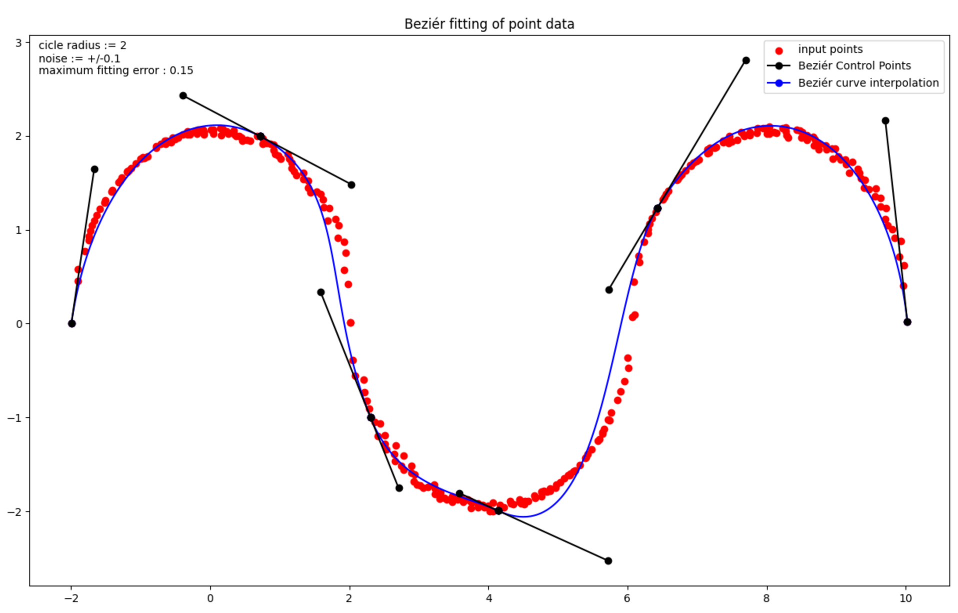
The robotic platform consists of a LBR IIWA 14 collaborative robot capable of torque sensing in all of its joints. Since it's out of the box controller provides little flexibility, two software packages have been developed.

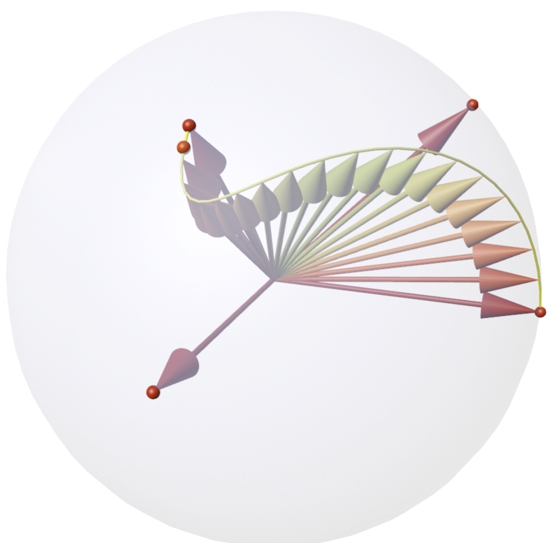
- *iiwa-fri-driver*: A real time abstracting driver in C++ binded to Python
- *iiwa-fri-gym*: A set of control algorithms and gym environments both for the real robot and an accurately simulated robot.



Geometric Curve Fitting

Once having created an appropriate platform, the next step is to identify the rehabilitation trajectories as geometric curves. For this purpose, Bézier curves are chosen.

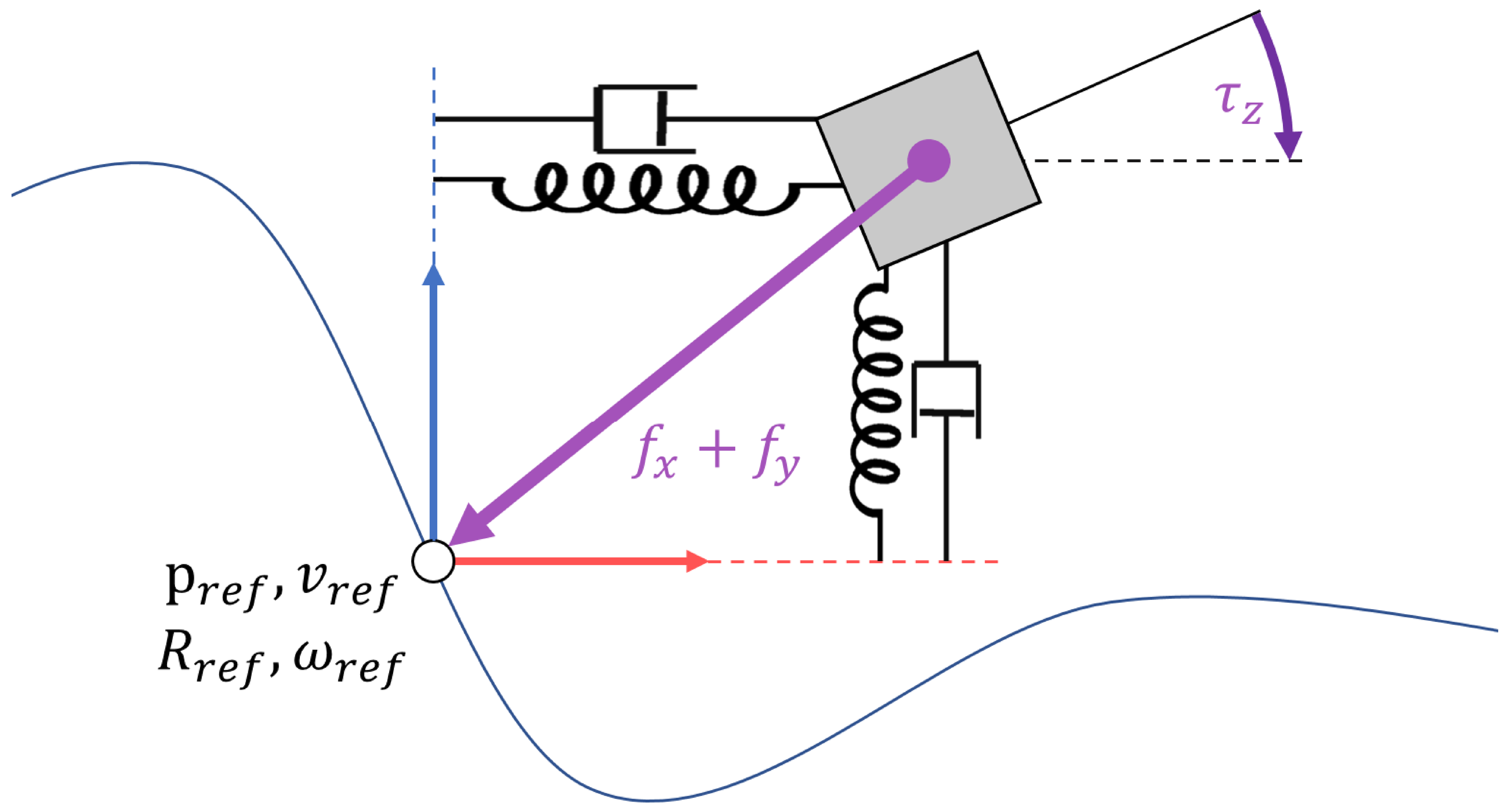




While multiple algorithms exist to obtain $2D$ Bézier curves from point data, its extension to $SE(3)$ is non-trivial. A measure has to be defined, as well as a robust analog for the *lerp* operation. To the left is shown a fitting of a 2D trajectory, and on the right an SO(3) one.

Cartesian Impedance

Then, we perform Cartesian impedance for a safe trajectory following. That is, move the robot's end-effector to behave in a spring -like manner.



$$(f, \tau) = K \cdot e + D \cdot \dot{e} + \Lambda \cdot \ddot{e}$$

For this, we choose a stiffness in each cartesian direction (3 rotation and 3 translation)

$$K = I \cdot (K_x, K_y, K_z, K_a, K_b, K_c) \text{ in } N/m, N.m$$

And a damping factor between 0.0 and 1.0 in the same manner.

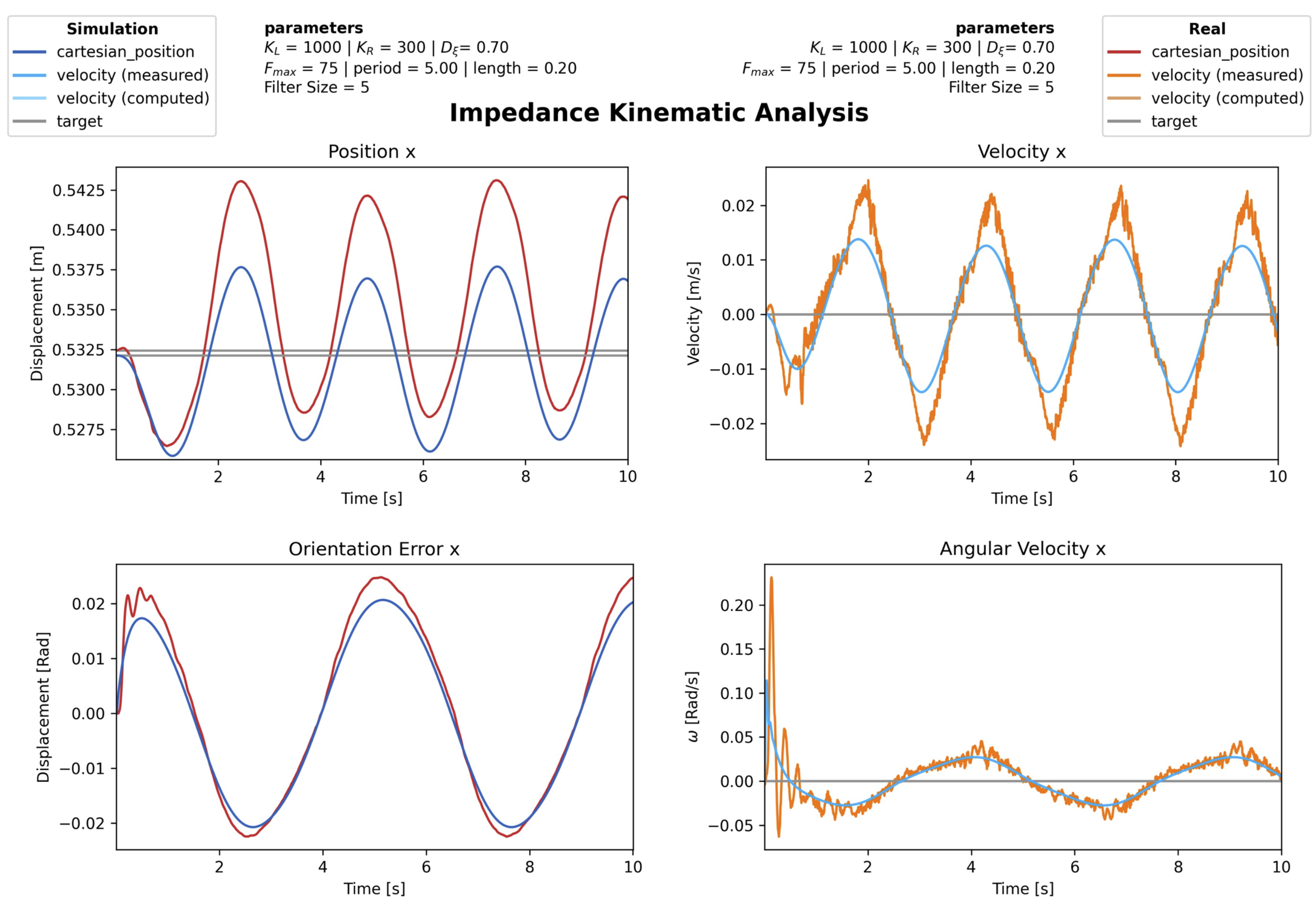
$$D_\xi = I \cdot (\xi_x, \xi_y, \xi_z, \xi_a, \xi_b, \xi_c) \text{ in } N/m, N.m$$

Then to find the appropriate D matrix, we use Cholesky decomposition of the Cartesian mass matrix M to then find the generalized eigenvalues o K with respect to M. These will also be the main damping directions.

$$\Lambda = J^{+^T} M_q J^{+} \qquad Q \rightarrow \Lambda = QQ^*$$

$$K_d = Q^{-1} K Q^{-1^T} \qquad D = 2 \cdot Q D_\xi K_d Q^T$$

Results



These results show the correct tracing of the given trajectory as well as the similarity between the simulated dynamics an the measured ones.

Acknowledgements

The research leading to these results has received funding from the projects ROBOASSET PID2020-113508RB-I00, ALMA H2020-EIC-FETPROACT-2019 and IROPER - Robótica inteligente para necesidades personales