The Journal of Open Source Software

A C++ Implementation of a Cartesian Impedance Controller for Robotic Manipulators

Matthias Mayr [1,2] **and Julian M. Salt-Ducaju** [2,3]

1 Department of Computer Science, Faculty of Engineering (LTH), Lund University, Sweden **2** Wallenberg AI, Autonomous Systems and Software Program (WASP), Sweden **3** Department of Automatic Control, Faculty of Engineering (LTH), Lund University, Sweden

DOI: 10.21105/joss.05194

Software

- Review
- Repository
- Archive

Editor: Adi Singh

Reviewers:

- @JHartzer
- @ShravanTata

Submitted: 20 December 2022
Published: 23 January 2024

Summary

A Cartesian impedance controller is a type of control strategy that is used in robotics to regulate the motion of robot manipulators. This type of controller is designed to provide a robot with the ability to interact with its environment in a stable and compliant manner. Impedance control increases the safety in contact-rich environments by establishing a mass-spring-damper relationship between the external forces acting on the robot and the variation from its reference defined by a set of coordinates that describe the motion of a robot. As a consequence, the controlled robot behaves in a compliant way with respect to its external forces, which has the added benefit of allowing a human operator to interact with the robot, such as manually guiding it.

In this package, we provide a C++ implementation of a controller that allows collaborative robots:

1. To achieve compliance in Cartesian task-frame coordinates;
2. To allow for joint compliance in the nullspace of task-frame coordinates; and
3. To be able to apply desired forces and torques to their environments, e.g., for direct force control.

This package can be used in any torque-controlled robotic manipulator. Its implementation in Robot Operating System (ROS) integrates it into `ros_control` (Chitta et al., 2017) and can automatically utilize the Unified Robot Description Format (URDF) description of the robot's geometry.

Statement of Need

Modern robotics is moving more and more past the traditional robot systems that have hard-coded paths and stiff manipulators. Many use cases require the robots to work in semi-structured environments. These environments impose uncertainties that could cause collisions. Furthermore, many advanced assembly, manufacturing, and household scenarios such as insertions or wiping motions require the robot to excert a controlled force on the environment. Finally, the robot workspace is becoming increasingly shared with human workers to leverage both agents and allow them to complement each other.

An implementation of compliant control for robotic manipulators is an attractive solution for robots in contact-rich environments. To cover the wide variety of tasks and scenarios, we think that it needs to fulfill the following criteria:

1. Dynamically adapt the end-effector reference point.
2. Dynamically adjust the robot's impedance (i.e., its ability to resist or comply with external forces).

3. Apply commanded forces and torques (i.e., a wrench) with the end-effector of the robot.
4. Command a joint configuration and apply it in the nullspace of the Cartesian robotic task.
5. Execute joint-space trajectories.

A complete implementation with respect to items 1–5 above of compliance for torque-commanded robotic manipulators is not available, and the existing solutions `franka_ros` (Franka Emika, 2017a) and `libfranka` (Franka Emika, 2017b) as well as the `KUKA` FRI Cartesian impedance controller can only be used for a single type of robotic manipulator:

	KUKA FRI controller	franka_ros	libfranka	**This package**
Reference Pose Update	(x)[1]	x		**x**
Cartesian Stiffness Update		x		**x**
Cartesian Wrench Update	(x)[1]			**x**
Nullspace Control	?	x		**x**
Kinesthetic Teaching	(x)[2,3]	x	(x)[3]	**x**
Trajectory Execution				**x**
Multi-Robot Support				**x**

This implementation offers a base library that can easily be integrated into other software and also implements a `ros_control` controller on top of the base library for the popular ROS middleware. The base library can be used with simulation software such as DART (Lee et al., 2018). It is used for contact-rich applications such as wiping a surface Mayr et al. (2023). Furthermore it is used in several research papers such as those by Mayr, Ahmad, et al. (2022); Mayr, Hvarfner, et al. (2022); Ahmad et al. (2022); and Ahmad et al. (2023) that explore reinforcement learning as a strategy to accomplish contact-rich industrial robot tasks.

The Robot Operating System (ROS) is an open-source middleware that is widely used in the robotics community for the development of robotic software systems (Quigley et al., 2009). Within ROS, an implementation of compliant control is available for position-commanded and velocity-commanded robotic manipulators with the `cartesian_controllers` package (Scherzinger et al., 2017). However, if a robotic manipulator supports direct control of the joint torques, e.g., the `KUKA LBR iiwa` or the `Franka Emika Robot (Panda)`, torque-commanded Cartesian impedance control is often the preferred control strategy, since a stable compliant behavior might not be achieved for position-commanded and velocity-commanded robotic manipulators (Lawrence, 1988).

Control Implementation

The gravity-compensated rigid-body dynamics of the controlled robot can be described, in the joint space of the robot $q \in \mathbb{R}^n$, as (Siciliano & Khatib, 2016):

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} = \tau_{\mathrm{c}} + \tau^{\mathrm{ext}} \tag{1}$$

where $M(q) \in \mathbb{R}^{n\times n}$ is the generalized inertia matrix, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ captures the effects of Coriolis and centripetal forces, $\tau_{\mathrm{c}} \in \mathbb{R}^n$ represents the input torques, and $\tau^{\mathrm{ext}} \in \mathbb{R}^n$ represents the external torques, with n being the number of joints of the robot. Since the proposed controller was evaluated using robots that are automatically gravity-compensated (`KUKA LBR iiwa` and `Franka Emika Robot (Panda)`), the gravity-induced torques have not been included in Equation 1. However, the proposed controller can be used in robots that

[1] An FRI connection can send either joint position updates or wrench updates
[2] Reaching a joint limit triggers a safety stop
[3] Can be implemented by setting the Cartesian stiffness to zero

are not automatically gravity-compensated by adding a gravity-compensation term to the commanded torque signal, τ_c.

Moreover, the torque signal commanded by the proposed controller to the robot, τ_c in Equation 1, is composed by the superposition of three joint-torque signals:

$$\tau_\mathrm{c} = \tau_\mathrm{c}^\mathrm{ca} + \tau_\mathrm{c}^\mathrm{ns} + \tau_\mathrm{c}^\mathrm{ext} \tag{2}$$

where

- $\tau_\mathrm{c}^\mathrm{ca}$ is the torque commanded to achieve a Cartesian impedance behavior (Hogan, 1985) with respect to a Cartesian pose reference in the m-dimensional task space, $\xi^\mathrm{D} \in \mathbb{R}^m$, in the frame of the end-effector of the robot:

  $$\tau_\mathrm{c}^\mathrm{ca} = J^\mathrm{T}(q)\left[-K^\mathrm{ca}\Delta\xi - D^\mathrm{ca}J(q)\dot{q}\right] \tag{3}$$

  with $J(q) \in \mathbb{R}^{m\times n}$ being the Jacobian relative to the end-effector (task) frame of the robot, and $K^\mathrm{ca} \in \mathbb{R}^{m\times m}$ and $D^\mathrm{ca} \in \mathbb{R}^{m\times m}$ being the virtual Cartesian stiffness and damping matrices, respectively. Also, the Cartesian pose error, $\Delta\xi$ in Equation 3 is defined as $\Delta\xi_\mathrm{tr} = \xi_\mathrm{tr} - \xi_\mathrm{tr}^\mathrm{D}$ for the translational degrees of freedom of the Cartesian pose and as $\Delta\xi_\mathrm{ro} = \xi_\mathrm{ro}\left(\xi_\mathrm{ro}^\mathrm{D}\right)^{-1}$ for the rotational degrees of freedom.
- $\tau_\mathrm{c}^\mathrm{ns}$ is the torque commanded to achieve a joint impedance behavior with respect to a desired configuration and projected in the null-space of the robot's Jacobian, to not affect the Cartesian motion of the robot's end-effector (Ott, 2008):

  $$\tau_\mathrm{c}^\mathrm{ns} = \left(I_n - J^\mathrm{T}(q)(J^\mathrm{T}(q))^\dagger\right)\tau_0 \tag{4}$$

  with the superscript † denoting the Moore-Penrose pseudoinverse matrix (Khatib, 1995)[4] given by $J^\dagger = (J^\mathrm{T}J)^{-1}J^\mathrm{T}$ (Ben-Israel & Greville, 2003), and τ_0 being an arbitrary joint torque formulated to achieve joint compliance,

  $$\tau_0 = -K^\mathrm{ns}(q - q^\mathrm{D}) - D^\mathrm{ns}\dot{q} \tag{5}$$

  where $K^\mathrm{ns} \in \mathbb{R}^{n\times n}$ and $D^\mathrm{ns} \in \mathbb{R}^{n\times n}$ are the virtual joint stiffness and damping matrices, respectively.
- $\tau_\mathrm{c}^\mathrm{ext}$ is the torque commanded to achieve the desired external force command in the frame of the end-effector of the robot, $F_\mathrm{c}^\mathrm{ext}$:

  $$\tau_\mathrm{c}^\mathrm{ext} = J^\mathrm{T}(q)F_\mathrm{c}^\mathrm{ext} \tag{6}$$

Safety Measures

As described in Figure 1, there are several safety measures that have been implemented in the controller to achieve a smooth behavior of the robot:

Filtering

The proposed controller allows the online modification of relevant variables: ξ^D, K^ca and D^ca in Equation 3, K^ns and D^ns in Equation 5, and $F_\mathrm{c}^\mathrm{ext}$ in Equation 6. However, for a smoother behavior of the controller, the values of these variables are low-pass filtered. The update law at each time-step k is:

$$\alpha_{k+1} = (1-a)\alpha_k + a\alpha^\mathrm{D} \tag{7}$$

where α^D is the desired new variable value and $a \in (0, 1]$ is defined in such a way that a user-defined percentage of the difference between the desired value α^D and the variable value at the time of the online modification instruction, α_0, is applied after a user-defined amount of time.

[4] The Moore–Penrose pseudoinverse is computationally cheap and allows a null-space projection disregarding the dynamics of the robot. However, not using the dynamics of the robot to fomulate a pseudoinverse matrix for null-space projection may cause that a non-zero arbitrary torque, τ_0 in Equation 4, generates interfering forces in the Cartesian space if the joint of the robot are not in a static equilibrium ($\dot{q} = \ddot{q} = 0$).

Saturation

To increase safety in the controller, some of the filtered variables (the stiffness and damping factors K^{ca}, D^{ca}, K^{ns}, and D^{ns}, and the desired external force command $F_{\mathrm{c}}^{\mathrm{ext}}$) can be saturated between user-defined maximum and minimum limits, i.e., for an example variable α:

$$\alpha_{\mathrm{min}} \leq \alpha \leq \alpha_{\mathrm{max}} \tag{8}$$

Rate Limiter

The rate of the commanded torque, τ_{c} in Equation 2, can be limited. For two consecutive commands at times k and $k+1$:

$$\Delta\tau_{\mathrm{max}} \geq \|\tau_{\mathrm{c},k+1} - \tau_{\mathrm{c},k}\| \tag{9}$$

Block Diagram

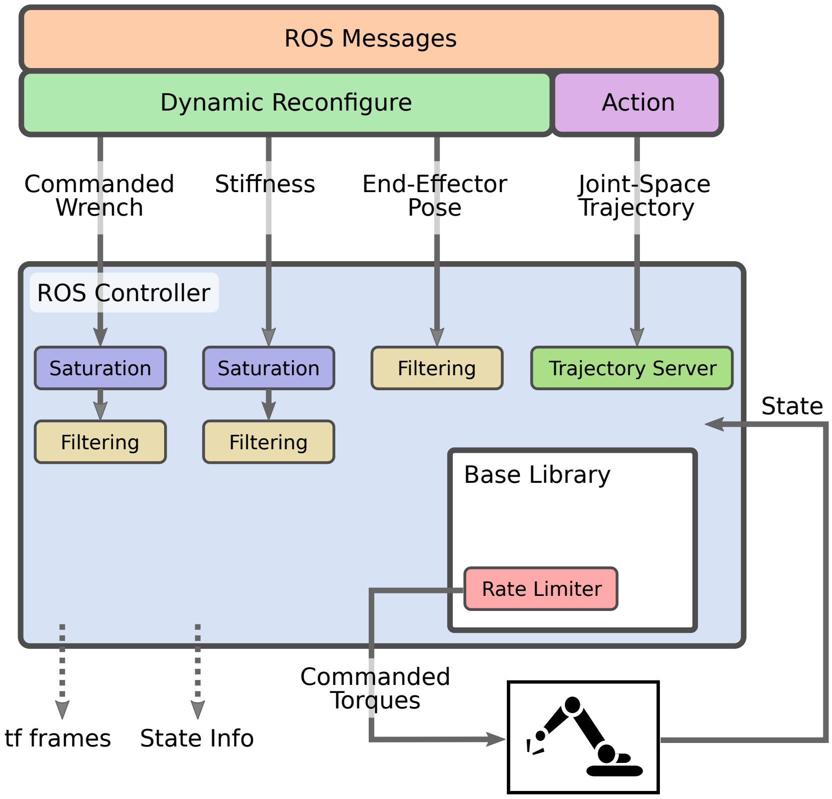

Figure 1: Block diagram of the controller.

Acknowledgements

We thank Björn Olofsson and Anders Robertsson for the discussions and feedback as well as Konstantinos Chatzilygeroudis for the permission to use the RBDyn wrapper code. Finally, we thank Oussama Chouman for his contributions during his thesis project.

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation. The authors are members of the ELLIIT Strategic Research Area at Lund University.

References

Ahmad, F., Mayr, M., & Krueger, V. (2023). Learning to adapt the parameters of behavior trees and motion generators (BTMGs) to task variations. *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 10133–10140. https://doi.org/10.1109/IROS55552.2023.10341636

Ahmad, F., Mayr, M., Topp, E. A., Malec, J., & Krueger, V. (2022). Generalizing behavior trees and motion-generator (BTMG) policy representation for robotic tasks over scenario parameters. *2022 IJCAI Planning and Reinforcement Learning Workshop*.

Ben-Israel, A., & Greville, T. N. (2003). *Generalized inverses: Theory and applications*. Springer-Verlag, New York, USA. https://doi.org/10.1007/b97366

Chitta, S., Marder-Eppstein, E., Meeussen, W., Pradeep, V., Rodríguez Tsouroukdissian, A., Bohren, J., Coleman, D., Magyar, B., Raiola, G., Lüdtke, M., & Fernández Perdomo, E. (2017). Ros_control: A generic and simple control framework for ROS. *The Journal of Open Source Software*. https://doi.org/10.21105/joss.00456

Franka Emika. (2017a). *Franka Emika Panda – franka_ros Documentation*. https://frankaemika.github.io/docs/franka_ros.html.

Franka Emika. (2017b). *Franka Emika Panda – libfranka Documentation*. https://frankaemika.github.io/docs/libfranka.html.

Hogan, N. (1985). Impedance control: An approach to manipulation: Parts I–III. *J. Dynamic Syst., Measurement, and Control*, *107*(1), 1–24. https://doi.org/10.1115/1.3140702

Khatib, O. (1995). Inertial properties in robotic manipulation: An object-level framework. *The International Journal of Robotics Research*, *14*(1), 19–36. https://doi.org/10.1177/027836499501400103

Lawrence, D. A. (1988). Impedance control stability properties in common implementations. *IEEE International Conference on Robotics and Automation (ICRA)*, 1185–1190. https://doi.org/10.1109/ROBOT.1988.12222

Lee, J., Grey, M. X., Ha, S., Kunz, T., Jain, S., Ye, Y., Srinivasa, S. S., Stilman, M., & Liu, C. K. (2018). DART: Dynamic animation and robotics toolkit. *The Journal of Open Source Software*, *3*(22), 500. https://doi.org/10.21105/joss.00500

Mayr, M., Ahmad, F., Chatzilygeroudis, K., Nardi, L., & Krueger, V. (2022). Skill-based multi-objective reinforcement learning of industrial robot tasks with planning and knowledge integration. *2022 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, 1995–2002. https://doi.org/10.1109/ROBIO55434.2022.10011996

Mayr, M., Ahmad, F., Duerr, A., & Krueger, V. (2023). Using knowledge representation and task planning for robot-agnostic skills on the example of contact-rich wiping tasks. *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 1–7. https://doi.org/10.1109/CASE56687.2023.10260413

Mayr, M., Hvarfner, C., Chatzilygeroudis, K., Nardi, L., & Krueger, V. (2022). Learning skill-based industrial robot tasks with user priors. *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 1485–1492. https://doi.org/10.1109/CASE49997.2022.9926713

Ott, C. (2008). *Cartesian impedance control of redundant and flexible-joint robots*. Springer, Berlin, Germany.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., & others. (2009). ROS: An open-source robot operating system. *IEEE International Conference on Robotics and Automation (ICRA): Workshop on Open Source Software*.

Scherzinger, S., Roennau, A., & Dillmann, R. (2017). Forward dynamics compliance control (FDCC): A new approach to cartesian compliance for robotic manipulators. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4568–4575. https://doi.org/10.1109/IROS.2017.8206325

Siciliano, B., & Khatib, O. (2016). *Springer handbook of robotics*. Springer, Berlin, Germany. https://doi.org/10.1007/978-3-319-32552-1