Comparing cascaded real-time controllers for an extended KUKA LBR iiwa robot during physical human-robot interaction

Vergleich kaskadierter Echtzeitregelungen eines erweiterten KUKA LBR iiwa-Roboters für physische Mensch-Roboter-Interaktion

Murali Karnam, Philippe Cattin, Georg Rauter, Nicolas Gerig
University of Basel, Department of Biomedical Engineering, 4123 Allschwil, Switzerland
{murali.karnam, philippe.cattin, georg.rauter, nicolas.gerig}@unibas.ch

Extended Abstract

Introduction

Medical robots require accurate motion control that is safe for patients, surgeons, the environment, and the robot itself. Motion commands for such robots can be generated by taking input from multiple modalities (teleoperation, physical interaction, or pre-planned paths from medical image data). Off-the-shelf robots are now used for surgical applications; for example, the KUKA LBR iiwa Med during Laserosteotomy (CARLO by Advanced Osteotomy Tools, Switzerland).
External control of robots is necessary to (a) react to changes in the environment, (b) select a suitable inverse kinematics solution, or (c) extend the robot with additional active degrees of freedom (DoF). KUKA LBR iiwa robots allow limited external control options; either position or impedance control with DirectServo interface at a maximum control rate of 0.5 kHz, or impedance control with an additional torque overlay at 1 kHz with the Fast Robot Interface (FRI by KUKA). Typically, external commands are calculated and commanded from a computer using the Robot Operating System (ROS) [1] or MATLAB [2], which are soft real-time systems. The lag between commanded and observed joint trajectories of up to 0.250 s has been reported [1] when commanding the robot from ROS over FRI.
We aim at minimizing joint control lag while externally controlling the KUKA LBR iiwa robot with an extended DoF for surgery in real-time through FRI. We compare the robot's performance with two control schemes – high-level closed-loop and high-level open-loop – for external position control, wherein the feedback over FRI is used or not used for controlling the robot.

Materials and Methods

We use a KUKA LBR iiwa robot mounted on a linear axis (AL2412 motor controlled with a Bechkoff AX5106 servo drive). High-level control of these components is executed on a CX-2020 (Beckhoff Automation GmbH) real-time computer as shown in Figure 1. The robot speed is limited to 0.25 m/s in the task space (ISO 10218-1) for safe human-robot interaction. In the joint space, the speed is limited to 0.25 m/s for linear and 5 rad/s for rotational joints.

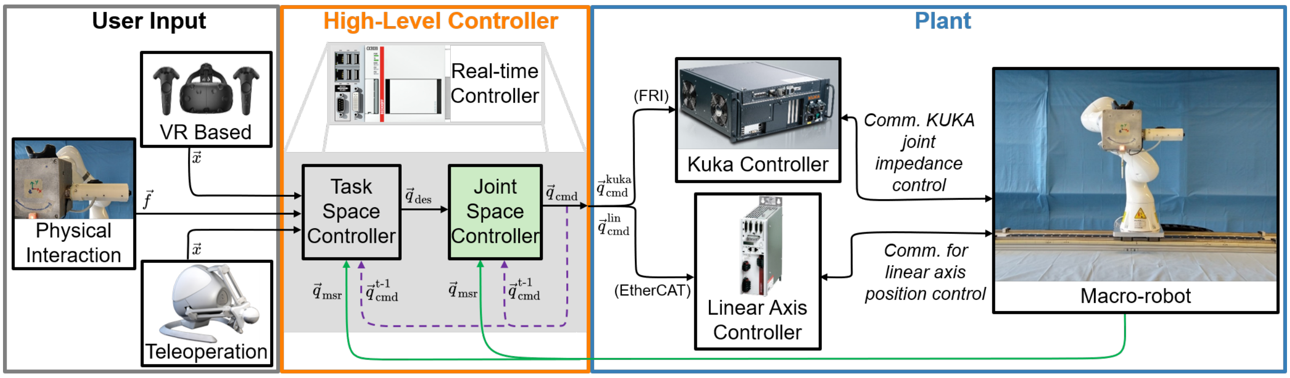

Figure 1 Architecture for real-time control of the 8-DoF macro-robot (KUKA LBR iiwa on a linear axis). Users can control it with three different input devices. The joint position command ($\vec{q}_{\text{cmd}}$) is sent to the KUKA controller over FRI and to the linear drive over EtherCAT at 1 kHz. The FRI client (C++) and the high-level controller (MATLAB Simulink) are compiled and run in the TwinCAT3 real-time environment. The focus of this work is the joint space controller (in green box), that calculates safe joint position commands.

To ensure commands sent to the robot by the high-level controller satisfy the safety limits, we use a linear joint space controller that follows the desired task space velocity:

$$\vec{q}_{\text{cmd}} = \begin{cases} \vec{q}_{\text{des}} & \text{if } q_{\text{des},i} - \hat{q}_i \le c_i \vee i \in [1,8] \\ \hat{\vec{q}} + \frac{(\vec{q}_{\text{des}}-\hat{\vec{q}})}{\|\vec{q}_{\text{des}}-\hat{\vec{q}}\|_\infty} c_* & \text{otherwise, where } c_* = \min\limits_i c_i \vee \{i \mid q_{\text{des},i} - \hat{q}_i > c_i\}, \end{cases}$$

This work may be used under a Creative Commons Attribution-ShareAlike 4.0 License (CC BY-SA 4.0) .

$\hat{\vec{q}}$ is the joint position feedback, and $\vec{c}$ is the maximum allowable joint increment per time step for each joint. The two controllers differ in the joint position feedback; for high-level closed-loop control $\hat{\vec{q}} = \vec{q}_{\text{msr}}$, and for high-level open-loop control $\hat{\vec{q}} = \vec{q}^{\,\text{t-1}}_{\text{cmd}}$ (the commanded position at the previous time step). The difference in the performance of the controllers (both running at 1 kHz) depends on the plant dynamics and the delay in feedback to the controller. In high-level closed-loop control, we estimate an approximate cycle delay of less than 5 ms for the measured position from low-level joint electronics to reach the controller and the updated command sent back to the robot joint.
The two controllers are compared with the bandwidth and step response of the 2nd joint of the robot at a pose with the highest inertia and no influence of gravity ($\vec{q} = [0, 0, \pi/2, 0, 0, 0, 0, 0]^\top$). A sinusoidal joint command ($q_{\text{des},2} = 0.1\sin(2\pi f t)$) is used to evaluate the bandwidth with the frequency, $f \in [0.1, 1]$ Hz. The step response is evaluated for a step of 0.05 rad. Both controllers were evaluated using the theoretical maximum joint increment per time step with a safety factor of 1.5, $c_2 = 1.8 \times 10^{-4}$rad. Additionally, to compensate the delay with the high-level closed-loop control, a $c_2^{\text{ex}} = 2.5 \times 10^{-3}$rad was experimentally searched as the maximum without going into a safety stop, and evaluated.

Results

The bandwidth of the high-level closed-loop controller was observed to be less than high-level open-loop control (see Figure 2). Using the experimentally tuned c_2^{ex} for high-level closed-loop control resulted in a similar bandwidth for both controllers. For small steps, the response of the high-level open-loop was faster (see Figure 3) taking 0.36 s to settle within an error of 1% and an overshoot of 3.9%, compared to the high-level closed-loop with a settling time of 0.47s and overshoot error of 4.92%. The step response is similar for larger steps.

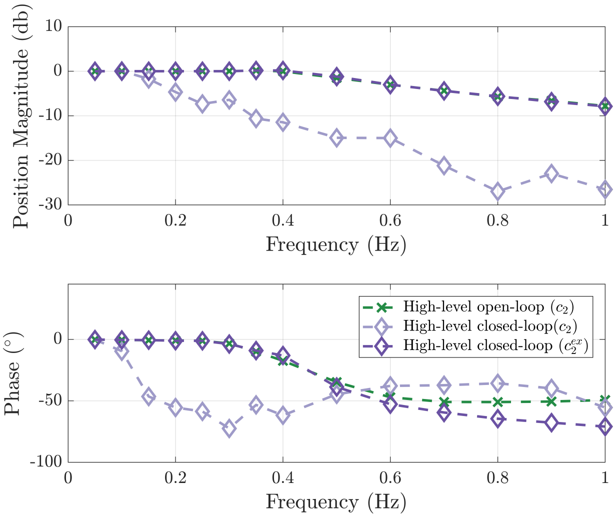

Figure 2 Bandwidth comparison for the 2nd joint of the macro-robot. The high-level closed-loop underperforms with c_2, but with the larger c_2^{ex}, it has a similar performance as the high-level open-loop control.

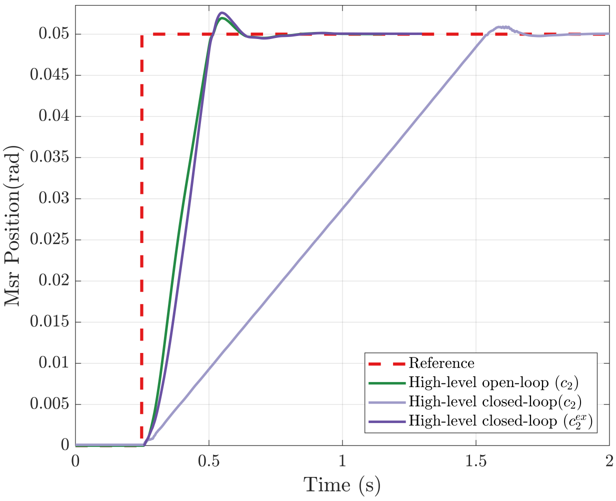

Figure 3 Step response of 2nd joint shows that high-level open-loop control performs better than high-level closed-loop control for chosen step size. We can also observe that the slope of the response is faster with high-level closed-loop control.

Discussion and Conclusions

To achieve the same bandwidth as the high-level open-loop control, 15 times larger c_2^{ex} was needed for high-level closed-loop control. We assume that larger increments in joint commands are needed to compensate for the delay in communicating over FRI, which cannot be reduced any further. For high-level open-loop control, c_2 that was calculated with a safety factor on the theoretical limit, was experimentally verified to be close to the safety limit. When only internal robot sensing is available, we suggest using a high-level open-loop controller for improved performance and simplicity of finding $\vec{c}$. If sensing is augmented with external tracking, $\vec{c}$ must be carefully chosen for a high-level closed-loop controller to achieve good performance.

Literatur

[1] V. Chawda and G. Niemeyer. Toward controlling a kuka lbr iiwa for interactive tracking. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1808–1814, 2017.

[2] M. Safeea and P. Neto. Kuka sunrise toolbox: Interfacing collaborative robots with matlab. *IEEE Robotics Automation Magazine*, 26(1):91–96, 2019.

DuEPublico
Duisburg-Essen Publications online

In: Siebte IFToMM D-A-CH Konferenz 2021

This text is made available via DuEPublico, the institutional repository of the University of Duisburg-Essen. This version may eventually differ from another version distributed by a commercial publisher.

DOI: 10.17185/duepublico/74060
URN: urn:nbn:de:hbz:464-20210217-132230-8

This work may be used under a Creative Commons Attribution-ShareAlike 4.0 License (CC BY-SA 4.0) .