Model-based Cartesian Impedance Control

Real Robot Challenge Phase II Report

Anonymous Author(s)[1]

[1]Affiliation
Email: Email

Abstract

For this phase, our model-based cartesian impedance was improved three ways: A refined, holistic formulation, task decomposition for complex manipulations and Bayesian optimization of the hyperparameters. Combined, these three additions dramatically improved grasping and manipulation performance on the real system. Videos:
sites.google.com/view/robotchallenge-modelbased

1 Introduction

In Phase I, we proposed a model-based Cartesian impedance controller (CIC) [1, 2, 3, 4] of the TriFinger robotic platform [5] for the Real Robot Challenge (RRC). The motivation for such a structured approach is balance between inductive biases and adaptivity. Encoding prior knowledge like the dynamics model and goal state avoids excessive learning, while the adaptivity of the impedance structure should allow for sim-to-real transfer without explicit consideration. While purely data-driven robot learning is a noble pursuit, the time and resource limitations of the RRC encourage maximal usage of prior knowledge. In Phase I, the controller had several limitations. For one, each finger was controlled independently. This severely limited the ability to perform orientation control, which is best achieved by moments generated by several fingers acting in a coordinated fashion. Since the CIC was 'derisked' in Phase I for position control, a richer formulation was derived that considers all three fingers.

However, there was still a remaining limitation. In the RRC, the task does not incorporate the rotational symmetry of the cube due to the colour matching requirement. As such, rotations of up to 180° could be required in each axis. Such manipulations cannot be achieved through one interaction, and so some degree of *planning* is required to achieve complex manipulations. We had not anticipated this factor initially, and relied on 'reset' heuristics to carry out new interactions as progress stalled. To improve this respect, we perform 'task decomposition' in order to break the manipulation down into simpler components. This allowed our CIC to carrying out transform primitives, e.g. rolling by 90°.

The last limitation was in hyperparameter selection. In Phase I, these were hand-tuned, building on intuition from control theory. For Phase II, we performed Bayesian optimization (BO) [6] to improve these parameters, both in simulation and the real system. This transformed our structured approach into black-box policy search of a structured controller. Videos illustrating the performance of the presented approach are provided [1].

The report is structured as follows: Section 2 explains the improved controller formulation, Section 3 details the task decomposition algorithm for 'primitive'-like manipulation, and Section 4 describes the use of Bayesian optimization. Section 5 provides the team's outlook for Phase III.

2 Cartesian Model-based Impedance Control

In this section, we summarize the controller and explain the improved holistic formulation.

Grasping and Position Control For the desired cartesian position of the ith fingertip $\boldsymbol{X}_i$, we define $\bar{\boldsymbol{X}}_i$ to be the error between this tip position and the cube's centre of mass $\boldsymbol{X}_c$, so $\bar{\boldsymbol{X}}_i = \boldsymbol{X}_c - \boldsymbol{X}_i$. We then define an impedance controller for $\bar{\boldsymbol{X}}_i$, a second order ODE that can be easily interpreted as a mass spring damper system with parameters $\{\boldsymbol{M}, \boldsymbol{D}, \boldsymbol{K}\}$,

$$\boldsymbol{M}\ddot{\bar{\boldsymbol{X}}}_i + \boldsymbol{D}\dot{\bar{\boldsymbol{X}}}_i + \boldsymbol{K}\bar{\boldsymbol{X}}_i = \boldsymbol{f}_i. \tag{1}$$

For the task of grasping the cube, each finger is controlled independently. Since the cube's center of mass

[1] https://sites.google.com/view/robotchallenge-modelbased

is estimated using a vision-based system, the damping factor was zeroed for fast control. Converting this cartesian space control law back to joint coordinates results in $\boldsymbol{\tau}_{1,i} = \boldsymbol{M}(\boldsymbol{q})\boldsymbol{J}^{-1}\ddot{\boldsymbol{X}}$, where $\boldsymbol{\tau}_{1,i}$ denotes the torques to be applied to finger i, $\boldsymbol{q}$ the joint configuration and $\boldsymbol{J}$ the Jacobian. The natural adaptivity of this impedance control law ensures a stable grasp of the cube.

To perform cube pose control, we follow the ideas presented in previous CIC literature [4] and design a proportional control law that perturbs the cube's center based on the goal position $\boldsymbol{X}_g$, $\hat{\boldsymbol{X}}_c = \boldsymbol{X}_c + K_1(\boldsymbol{X}_g - \boldsymbol{X}_c)$. Replacing $\boldsymbol{X}_c$ by $\hat{\boldsymbol{X}}_c$ in the equation above results in our impedance-based grasping and position control law. It ensures that the cube is grabbed and held (mostly by choosing the stiffness $\boldsymbol{K}$) and moves the cube to the desired goal location. K_1 determines the evolution of the reference position and allows to control how fast the cube moves to its target.

Force-based, Holistic Components Despite the functionality of the initial controller, it did not consider the fingers as a whole, and so was limited in controlling orientation of the cube. Contact forces were also passively applied rather than explicitly considered. To incorporate these additional considerations, we superimpose four torques. First is the already introduced position control and gravity compensation, which is added with three contact and rotational terms explained in the following, such that $\boldsymbol{\tau}_i = \sum_{j=1}^{4} \boldsymbol{\tau}_{j,i}$.

To also allow directly specifying the force applied by each finger, we introduce an additional component $\boldsymbol{\tau}_{2,i} = \boldsymbol{J}^{\intercal}\boldsymbol{F}_{2,i}$, where $\boldsymbol{F}_{2,i}$ is the force applied by finger i. We chose $\boldsymbol{F}_{2,i}$ to be in the direction of the surface normal of the face where finger i touches the cube ($\boldsymbol{F}_{2,i} = K_2\boldsymbol{d}_i$). However, to not counteract the impedance controller, the resulting force of this component $\boldsymbol{F}_{\text{res}} = \sum_i \boldsymbol{F}_{2,i}$ should account to zero. We ensure this by solving

$$-\boldsymbol{F}_{\text{res}} = [\boldsymbol{J}^{-\intercal}, \boldsymbol{J}^{-\intercal}, \boldsymbol{J}^{-\intercal}][\boldsymbol{\tau}_{3,1}, \boldsymbol{\tau}_{3,2}, \boldsymbol{\tau}_{3,3}]^{\intercal}, \qquad (2)$$

for $\boldsymbol{\tau}_{3,i}$. All previous components ensure a stable grasp closure. This is essential for the following orientation control law. Neglecting the cube's exact shape, we model the moment that is exerted onto the cube as $\boldsymbol{\Omega} = \sum \boldsymbol{r}_i \times \boldsymbol{F}_{4,i} = \sum \boldsymbol{S}_{r_i}\boldsymbol{F}_{4,i}$, where $\boldsymbol{r}_i = -\bar{\boldsymbol{X}}_i/|\bar{\boldsymbol{X}}_i|_2$ denotes the vector pointing from the cube's center towards the finger position, $\boldsymbol{S}_{r_i}$ the respective skew-symmetric matrix, and $\boldsymbol{F}_{4,i}$ an additional force that should lead to the desired rotation. The goal is now to realize a moment proportional to the current rotation errors, which are provided in the form of an axis of rotation $\boldsymbol{r}_{\boldsymbol{\phi}}$ and its magnitude ϕ. Thus, the control law yields $\boldsymbol{\Omega} = K_3\phi\boldsymbol{r}_{\boldsymbol{\phi}}$. We achieve $\boldsymbol{\Omega}$ by solving

$$\boldsymbol{\Omega} = [\boldsymbol{S}_{r_1}\boldsymbol{J}^{-\intercal}, \boldsymbol{S}_{r_2}\boldsymbol{J}^{-\intercal}, \boldsymbol{S}_{r_3}\boldsymbol{J}^{-\intercal}][\boldsymbol{\tau}_{4,1}, \boldsymbol{\tau}_{4,2}, \boldsymbol{\tau}_{4,3}]^{\intercal} \quad (3)$$

for $\boldsymbol{\tau}_{4,i}$. Associated parameters are tuned using Bayesian optimization.

Finger Placement We further combine the previously introduced control law with a simple finger placement heuristic. We place the fingers such that three out of the four faces that are perpendicular to the ground plane are covered. To obtain stable position control, the face which is closest to the goal location is not assigned any finger, ensuring that we can push the cube to the target. During orientation control this heuristic might be adapted. In case the desired axis of rotation is parallel to the ground plane, we place the fingers such that both faces that intersect with the axis of rotation are covered, thereby greatly simplifying this task.

3 Task Decomposition of Manipulations

As previously mentioned, the impedance controller does not provide the functionality to perform multiple interactions with the object. Additionally, given the limited maneuverability of the fingers, achieving some level 4 goal poses in a single interaction is impossible. Therefore, it is necessary to decompose the desired manipulation into a series of transformations that the controller *can* perform. This constitutes as a weak form of planning, as it assumes that the CIC is capable of performing a set of 'primitive' object manipulations.

For all levels, before enabling the control law, the fingers are moved to the vicinity of the desired grasping points using via points and inverse kinematics position control. This ensures that when activating the

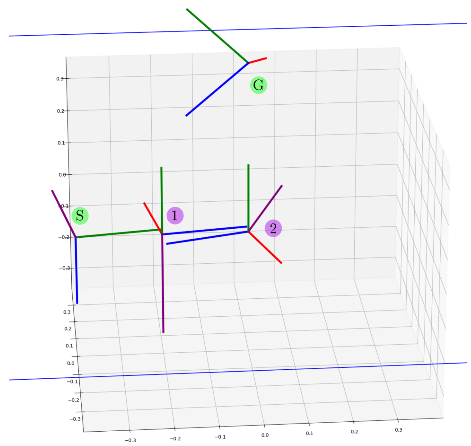

Figure 1: Illustration of a transform decomposition. Moving from frame S to frame G, frames 1 and 2 are intermediate transform in one axis. In purple, we display the rotation vector to move to the next frame.

manipulation controllers, contact is made in unison to prevent undesired movements during the grasp.

In terms of manipulation, for levels 1-3 which do not consider orientation, we use two 'primitive' controllers. One capable of moving the cube in the ground plane to the desired location and the other one stabily lifting the cube to its desired position. Switching between the controllers is done through a threshold based on the distance to the goal location.

To tackle level 4, which requires multiple interactions with the cube and includes orientation, we propose a hierarchical structure, in which the task is first decomposed into simpler goals and then the robot tries to achieve these individual goals sequentially exploiting the associated 'primitive' control law. We frame the task sequentiation as a linear concatenation of homogeneous transformations

$$\boldsymbol{H}_w^g = \boldsymbol{H}_2^g \boldsymbol{H}_1^2 \boldsymbol{H}_s^1 \boldsymbol{H}_w^s = \boldsymbol{H}_2^g \boldsymbol{H}_1^2 \boldsymbol{H}_w^1 = \boldsymbol{H}_2^g \boldsymbol{H}_w^2 \quad (4)$$

where, $\boldsymbol{H}_w^g$ is the target pose, $\boldsymbol{H}_w^s$ is the starting pose and $\boldsymbol{H}_w^1$ and $\boldsymbol{H}_w^2$ some intermediate poses to ease the problem. See Figure 1 for an illustration of the frames. To compute $\boldsymbol{H}_w^2$, we frame it as the shortest distance pose from $\boldsymbol{H}_w^g$ s.t. one of the axis vectors is normal to the floor. This constraint minimizes the additional rotation that is needed when lifting the cube from the ground. Moving from S frame to the second frame 2 is done with an additional via point. First, we rotate the cube to ensure that the normal vector in the desired pose 2 and the normal vector in pose 1 match. Then, we apply a rotation parallel to the floor to move from pose 1 to pose 2.

4 Bayesian Optimization of Hyperparameters

From the formulation detailed in Section 2, the CIC mainly depends on six hyperparameters $(\boldsymbol{K}, K_1, K_2, K_3)$. The overall control strategy contains more parameters, such as via points for approaching the desired grasp locations or thresholds that define when to switch to the next 'primitive' controller. Bayesian optimization was used to perform sample-efficient black-box optimization, using the BoTorch library [7]. Optimizing the hyperparameters can also be viewed as correcting for modelling error through parameter tuning. BO was first performed in simulation, where it was found that the contact-based hyperparameters (impedance stiffness and the force scalar) were miscalibrated. While previously the stiffnesses were the same in each dimension, having different xy and z values was found to dramatically improve performance.

When applying BO on the real system, only the key contact-based parameters were optimized to reduce the dimensionality of the search space.

5 Outlook

We have presented our updated cartesian impedance controller approach, which has now evolved into black-box policy search with a structured controller. We look forward to seeing how this approach compares to more learning- or planning-based strategies. One extension not carried out in this phase was incorporating model learning. This was shelved due to the effectiveness of the provided model on the real system. Phase III presents the added challenge of manipulating non-axisymmetric objects. We hope that our proposed approach is flexible enough to scale to this setting and assume that it might be necessary to improve upon the finger placement as this greatly influences which types of manipulations are possible.

References

[1] Kevin M. Lynch and Frank C. Park. *Modern Robotics: Mechanics, Planning, and Control.* 2017.

[2] Thomas Wimböck, Christian Ott, Alin Albu-Schäffer, and Gerd Hirzinger. Comparison of object-level grasp controllers for dynamic dexterous manipulation. *The International Journal of Robotics Research*, 2012.

[3] Luigi Biagiotti, Hong Liu, Gerd Hirzinger, and Claudio Melchiorri. Cartesian impedance control for dexterous manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.

[4] Martin Pfanne, Maxime Chalon, Freek Stulp, Helge Ritter, and Alin Albu-Schäffer. Object-level impedance control for dexterous in-hand manipulation. *IEEE Robotics and Automation Letters*, 2020.

[5] Manuel Wüthrich, Felix Widmaier, Felix Grimminger, Joel Akpo, Shruti Joshi, Vaibhav Agrawal, Bilal Hammoud, Majid Khadiv, Miroslav Bogdanovic, Vincent Berenz, Julian Viereck, Maximilien Naveau, Ludovic Righetti, Bernhard Schölkopf, and Stefan Bauer. Trifinger: An open-source robot for learning dexterity, 2020.

[6] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, 2012.

[7] Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: Programmable Bayesian Optimization in PyTorch. *arxiv e-prints*, 2019.