Compliant Robotic Manipulation

TTK4550

Martin Hagen Myrestrand

December 17th 2020

This is a report presented as a part of the specialization project TTK4550 at the Faculty of Information Technology and Electrical Engineering

NTNU – Trondheim
Norwegian University of
Science and Technology

Abstract

Compliant robotic manipulation is an area with growing importance in the field of robotic control, playing a crucial part in various safety-critical applications. This work is aimed at developing a framework for safe force-constrained and compliant interactions, enabling execution of complex tasks requiring human-level adaptability. Currently, Variable Impedance Control (VIC) is a prominent approach for achieving such adaptable compliant manipulation skills. Especially, there is an increasing focus on learning based VIC, performing online optimization with respect to the variable stiffness and damping of the controller. However, these methods usually demand a lot of interactions to achieve good performance. In this work, along with the future master thesis, we aim to implement and assess a framework for learning based VIC, utilizing efficient sampling methods. Specifically, we aim to evaluate our approach in the use case of performing robotic ultrasound examinations on humans, focused on performing a motion trajectory while maintaining a desired contact-force with various safety constraints present.

This project work, being the foundation for the continuing master thesis, covers the preparatory work, constituting three main contributions; literature review, setting up a framework for the Franka Emika Panda robot, and implementing and testing two fundamental approaches within impedance control.

In terms of the framework of the Panda robot, a software-environment has been set up, allowing swift and intuitive control, both in a Gazebo-based simulator and in an experimental setup. Further, regarding the implementation of impedance control, two different control laws have been investigated. One allowing-, and one avoiding inertia shaping. The implemented controllers were tested in simulations and experiments, and were analysed for their performance. Early testing proved the inertia shaping impedance controller to be unstable as a result of non-representative estimates of the external wrench. The controller avoiding inertia shaping avoided this obstacle. Not being dependent on the sensed external wrench, it provided a predictable compliant behavior, avoiding high contact forces. Also, it performed well at having the end-effector establish contact with a dummy human torso, both limiting the contact forces and being able to hold a desired pose. For the immediate further work, we would like to investigate methods of improving the estimates of the external wrench, enabling the development of sample efficient learning based VIC.

Table of contents

Mathematical notations

$\boldsymbol{q} \in \mathbb{R}^{n\times 1}$	joint angles
$\dot{\boldsymbol{q}} \in \mathbb{R}^{n\times 1}$	joint velocities
$\boldsymbol{p} \in \mathbb{R}^{3\times 1}$	position
$\boldsymbol{v} \in \mathbb{R}^{6\times 1}$	velocity, consisting of translational and rotational velocity
$\Delta\boldsymbol{p}_{de} \in \mathbb{R}^{3\times 1}$	error in position
$\Delta\boldsymbol{v}_{de} \in \mathbb{R}^{6\times 1}$	error in velocity
$\boldsymbol{h}_e \in \mathbb{R}^{6\times 1}$	external wrench
$\boldsymbol{h}_c \in \mathbb{R}^{6\times 1}$	applied wrench
$\boldsymbol{\tau} \in \mathbb{R}^{n\times 1}$	applied torque
$\boldsymbol{x} \in \mathbb{R}^{6\times 1}$	pose, consisting of position and orientation

Acronyms

DOF Degrees of freedom. 4, 6, 7, 9, 12

FCI Franka Control Interface. ii, vi, 12–14, 26

RL Reinforcement learning. 1, 11

ROS Robot Operating System. v, 12, 13, 16, 26

VIC Variable Impedance Control. i, 1, 11, 27, 28

Glossary

Gazebo Gazebo is an open-source 3D robotics simulator.. vi, 13–17, 26, 28

MoveIt! Motion planning framework running on top of ROS utilizing some of their common tools like the ROS Visualizer (Rviz) and the ROS robot format (URDF).. 13, 14

URDF URDF, or Unified Robotic Description Format, is an XML specification used in academia and industry to model multibody systems [1].. 13

List of Figures

List of Tables

1 Introduction

This section consists of the two subsections *Motivation* and *Outline*, and provides an introduction to the task at hand. The first part is aimed at giving an overview of the current status in the field, and through this explain the motivation behind the work described in this thesis. The second part has the purpose of describing the structure of the report itself.

Motivation

The work described in the thesis is performed as a part of Sintef's project ROMO. The full working title being "RObotics for Moving Objects within manufacturing and healthcare – enabling the future internet of things". The essence of it being development of methods and algorithms that allow robots to interact with, and handle moving objects in real time [6]. The healthcare application of it require manipulators to achieve safe interaction with humans. Automatic ultrasound examinations, where the robot must be able to both find and follow organs and other structures is an example of this. Focused on this use case, the scope of the thesis is limited to have a manipulator follow a desired trajectory while ensuring a compliant behavior when interacting with the environment.

Compliant robotic manipulation is a hot topic in the area of robotic control. The motivation being safe force-constrained interactions with the environment and performing complex tasks, requiring human-like skills of adaptability. Today, the main part of the research is focused on developing control methods that achieve this high level of adaptability, often falling under the category of Variable Impedance Control (VIC). Especially, there is an increasing focus within learning-based VIC, utilizing Machine Learning techniques such as Reinforcement Learning, to perform online optimization with respect to the variable stiffness and damping parameters of the controller. However, these methods usually demand a lot of interactions to achieve good performance. Therefore there have also been an increasing focus directed at sampling efficiency in the field. An example of this is Li et al.'s development of a data-efficient RL method, based on a probabilistic Gaussian Process (GP) model of the system.

As the end goal of this work, including the upcoming master thesis, we aim to implement and assess a learning-based VIC, utilizing efficient sampling methods. Specifically, we want to test it in the above mentioned use case of performing ultrasound examinations on humans, focused on performing a motion trajectory while maintaining a desired contact-force. To get there, we first have to do some preparatory work, laying the foundation for the master thesis. This thesis covers this preparatory work, constituting three main contributions; literature review, setting up a framework for the Franka Emika Panda robot, and implementing and testing two fundamental approaches within impedance control.

Outline

The report is divided into a total of seven sections including this one. In the next one, the reader will be provided with the necessary theoretical background within robotic interaction control and dynamics of robot manipulators. Then the robot-hardware and software central to the task at hand will be reviewed. Thereafter the implementation is presented, followed by the results of simulations and experiments. Finally, the reader is presented with the discussion and conclusion.

2 Background

In this section, the theoretical background forming the foundation of this work is presented. To begin with, some general preliminary concepts are reviewed. This is followed by a presentation of robot manipulator dynamics. The rest of the section is focused on robotic interaction control and related aspects, including indirect force control, impedance control and variable impedance control.

2.1 General concepts

Orientation in 3D space

In this report, three different representations of orientation and rotation are treated.
Rotation matrices, *Euler angles* and *quaternions* respectively. A rotation matrix, describing a three dimensional rotation, transforming the frame from orientation a to orientation b is denoted $\boldsymbol{R}_a^b \in \mathbb{R}^{3\times3}$. Being an orthogonal matrix per definition, the reverse rotation $\boldsymbol{R}_b^a$ can be found as $\boldsymbol{R}_a^{b^{-1}} = \boldsymbol{R}_a^{b^T}$. The rotation matrix operates as a transformation matrix, e.g. with the ability to represent positions and velocities with respect to different frames, Σ. In particular, in this project it is used to transform forces, positions, velocities, accelerations and rotations to the appropriate frame, whether it may be in the base frame or the end-effector frame.
Euler angles, $\begin{pmatrix}\alpha & \beta & \gamma\end{pmatrix}^T \in \mathbb{R}^{3\times1}$, are offering a more minimal form of representation, describing three simple rotations relative to the axes of its own moving coordinate frame Σ. When operating with Euler angles, the order of the rotations matter, and there are several conventions among the 12 possible orders of rotations. Z-Y-X, Z-Y-Z and Z-X-Z are commonly used [7].
A drawback of the Euler angle based representation is occurrence of singularities in the vector notation. The *quaternion* representation is extremely useful as it avoids this problem. A quaternion has the form

$$\boldsymbol{\epsilon} = \eta + \epsilon_1 i + \epsilon_2 j + \epsilon_3 k\,. \tag{1}$$

$\eta, \epsilon_1, \epsilon_2$ and ϵ_3 being scalars, and i, j and k being operators satisfying the following combinatory rules

$$ii = jj = kk = -1, \qquad ij = k,\; jk = i,\; ki = j, \qquad ji = -k,\; kj = -i,\; ik = -j. \tag{2}$$

The conjugate of the quaternion is defined as

$$\tilde{\boldsymbol{\epsilon}} = \eta - \epsilon_1 i - \epsilon_2 j - \epsilon_3 k\,, \tag{3}$$

and is useful for defining the *unit quaternion*, satisfying $\tilde{\boldsymbol{\epsilon}}\boldsymbol{\epsilon} = \eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1$. Often η and $\begin{pmatrix}\epsilon_1 & \epsilon_2 & \epsilon_3\end{pmatrix}^T$ is referred to as the scalar- and vector part of the quaternion respectively.

The skew-symmetric operator

The *skew-symmetric operator*, $\boldsymbol{S}(*)$, takes a vector $\begin{pmatrix} v_1 & v_2 & v_3 \end{pmatrix}^T$ as input, and produces a skew-symmetric matrix

$$\boldsymbol{S}(\boldsymbol{v}) = \begin{pmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{pmatrix}, \tag{4}$$

corresponding to the vector product of $\boldsymbol{v}$.

2.2 Robot manipulator dynamics

Jacobian

The jacobian, $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{6\times n}$, is a mapping between joint space and task space based on the forward kinematics of the manipulator. Accordingly, it is used to calculate the cartesian (task space/operational space) velocity as a function of joint velocities

$$\boldsymbol{v}_e = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \tag{5}$$

and reversely the actuator torque $\boldsymbol{\tau}$ as a function of the task space wrench $\boldsymbol{h}_c$ from the controller

$$\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{h}_c. \tag{6}$$

Task space formulation of the dynamic model

For the task at hand, it is most convenient to consider the task space formulation of the dynamical system. For a rigid 6-DOF robot manipulator in contact with the environment, the model can be described by

$$\boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{v}}_e + \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{v}_e + \boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{h}_c - \boldsymbol{h}_e, \tag{7}$$

where $\boldsymbol{h}_c$ is the controller's force output, $\boldsymbol{h}_e$ is the external wrench, $\boldsymbol{\Lambda}(\boldsymbol{q}) \in \mathbb{R}^{6\times 6}$ is the cartesian inertia matrix, $\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{6\times 6}$ is the wrench caused by centrifugal and Coriolis effects, and $\boldsymbol{\eta}(\boldsymbol{q}) \in \mathbb{R}^{6\times 1}$ is the wrench of the gravitational effects. The cartesian inertia matrix, $\boldsymbol{\Lambda}(\boldsymbol{q})$, is calculated as

$$\boldsymbol{\Lambda}(\boldsymbol{q}) = (\boldsymbol{J}\boldsymbol{H}(\boldsymbol{q})^{-1}\boldsymbol{J}^T)^{-1}, \tag{8}$$

where $\boldsymbol{H}(\boldsymbol{q}) \in \mathbb{R}^{n\times n}$ is the symmetric and positive-definite joint space inertia matrix. This inertia matrix is representing the mass distribution of the manipulator and is highly state-dependent. By additionally knowing the joint space formulation of the centrifugal and Coriolis effects, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, the corresponding wrench, $\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, is

$$\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{J}^{-T}\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\boldsymbol{J}^{-1} - \Lambda(\boldsymbol{q})\dot{\boldsymbol{J}}\boldsymbol{J}^{-1}. \tag{9}$$

The last component of (7), the wrench of the gravitational effects, is found as $\boldsymbol{\eta}(\boldsymbol{q}) = \boldsymbol{J}^{-T}\boldsymbol{g}(\boldsymbol{q})$, where $\boldsymbol{g}(\boldsymbol{q})$ is the joint space quantity.

2.3 Robotic Interaction Control

Robotic interaction control is the concept of controlling the interaction between a manipulator and its environment. Combined with methods of motion control, manipulators can be made capable of following desired trajectories while ensuring a *compliant behavior* with respect to external forces, providing safe and stable control. Interaction control also make it possible to perform more advanced manipulation tasks, requiring adaptability that is enabled through force feedback. The methods to achieve interaction control is divided into two broad approaches, *passive-* and *active interaction control* (Fig. 1). In the passive approach, the compliant behavior is inbuilt into the agent's hardware. For example by structural compliance of the joints, links, position servo or end-effector. A big drawback with this method is its lack of flexibility as the compliant structure must be adapted to fit each special use case. Also, it can only handle small deviations from its pre-programmed trajectory [8]. Active interaction control is a more sophisticated approach where the control system is responsible for securing compliance. Although usually being slower and more expensive than the passive one, it is capable of overcoming the previously mentioned drawbacks of passive interaction control [8]. In this thesis we specifically focus on active interaction control.

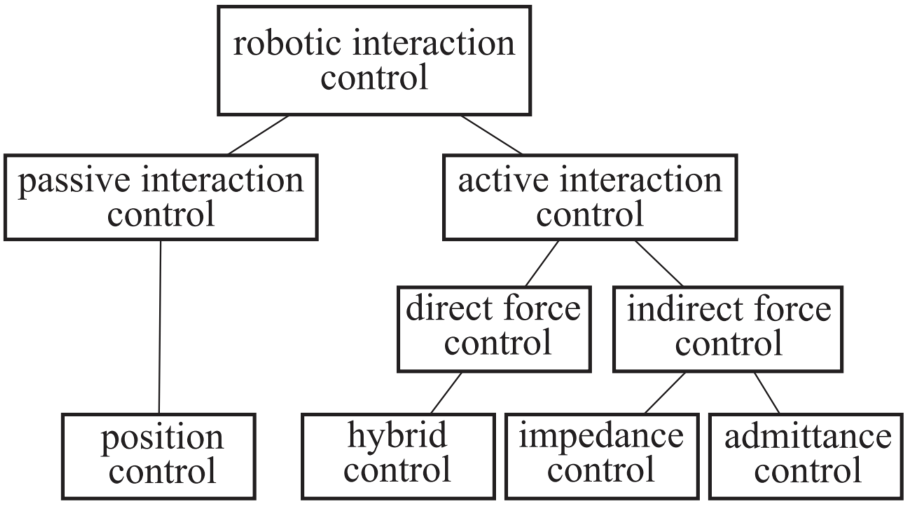


Figure 1: Interaction control classification as proposed in [2] obtained from [3]

2.4 Indirect Force Control

Active interaction control can be divided into two categories, *indirect-* and *direct force control.* The latter is recognized by the use of force feedback in closed-loop. Opposed to the indirect method, it is achieving force regulation, controlling the contact force and moment to a desired value. Without any explicit closed force feedback loop, the indirect method is instead achieving force control through motion control [8] e.g. by changing the reference position to comply with the interaction force. *Impedance control* and *Admittance control* are examples of approaches within indirect motion control.

Stiffness Control

Before introducing impedance control, we need to start with the corner stone of indirect control, *Stiffness control.* To begin with the basics, the position and orientation of the end-effector (*pose*) can be described by the vector $\boldsymbol{x}_e = \begin{pmatrix} \boldsymbol{p}_e^T & \boldsymbol{\phi}_e^T \end{pmatrix}^T \in \mathbb{R}^{6\times1}$, where $\boldsymbol{p}_e$ is the position and $\boldsymbol{\phi}_e$ is a set of Euler angles describing the orientation. Similarly, $\boldsymbol{x}_d$ is defined as the desired pose. Assuming a constant $\boldsymbol{x}_d$, the end-effector's deviation from the desired pose is, $\Delta\boldsymbol{x}_{de} = \boldsymbol{x}_d - \boldsymbol{x}_e$, and the velocity error is, $\Delta\dot{\boldsymbol{x}}_{de} = -\dot{\boldsymbol{x}}_e = -\boldsymbol{A}^{-1}(\boldsymbol{\phi}_e)\boldsymbol{v}_e$, with

$$\boldsymbol{A}(\boldsymbol{\phi_e}) = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{T}(\boldsymbol{\phi_e}) \end{pmatrix}, \tag{10}$$

$\boldsymbol{I} \in \mathbb{R}^{3\times3}$ being the identity matrix, $\boldsymbol{0} \in \mathbb{R}^{3\times3}$ being the null matrix and $\boldsymbol{T} \in \mathbb{R}^{3\times3}$ being the mapping function $\boldsymbol{\omega}_e = \boldsymbol{T}(\boldsymbol{\phi_e})\dot{\boldsymbol{\phi}}_e$, where $\boldsymbol{\omega}_e$ is the angular velocity of the end-effector. If we now introduce the following proportional-derivative motion control law with gravity-compensation

$$\boldsymbol{h}_c = \boldsymbol{A}^{-T}(\boldsymbol{\phi}_e)\boldsymbol{K}_P\Delta\boldsymbol{x}_{de} - \boldsymbol{K}_D\boldsymbol{v}_e + \boldsymbol{\eta}(\boldsymbol{q}), \tag{11}$$

where $\boldsymbol{K}_P \in \mathbb{R}^{6\times6}$ and $\boldsymbol{K}_D \in \mathbb{R}^{6\times6}$ are symmetric positive-definite matrices, we find the following asymptotically stable equilibrium

$$\boldsymbol{h}_e = \boldsymbol{A}^{-T}(\boldsymbol{\phi}_e)\boldsymbol{K}_P\Delta\boldsymbol{x}_{de}. \tag{12}$$

According to (12), the end-effector will behave as a 6-DOF spring in respect of the external wrench $\boldsymbol{h}_e$. (12) also show that the matrix $\boldsymbol{K}_P$ is playing the role of an *active stiffness*, controlling the compliance of the manipulator. The choice of $\boldsymbol{K}_P$ is then important for ensuring a suitable elastic behaviour. For example, you can design this stiffness matrix so that you have high positional accuracy in the xy-plane, and allowing more compliance in the z-direction, reducing interaction-forces along this axis. Establishing a second-order dynamical relationship between the deviation of the end-effector pose and the external wrench is what we define as stiffness control [8].

Mechanical Springs

The compliant behavior of two elastically coupled rigid bodies, A and B, with coinciding reference frames Σ_a and Σ_b, can near the equilibrium be described by the linear mapping

$$\boldsymbol{h}_b^b = \boldsymbol{K}\delta\boldsymbol{x}_{ab}^b = \begin{pmatrix} \boldsymbol{K}_t & \boldsymbol{K}_c \\ \boldsymbol{K}_c^T & \boldsymbol{K}_o \end{pmatrix} \delta\boldsymbol{x}_{ab}^b, \tag{13}$$

where $\boldsymbol{h}_b^b$ is the elastic wrench applied to body B, expressed in Σ_B, while being exposed to an infinitesimal twist displacement $\delta\boldsymbol{x}_{ab}^b$, also expressed in Σ_B [8]. $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ in (13) is the symmetric positive-semidefinite stiffness matrix, describing the behavior of an ideal 6-DOF spring. It consists of the *translational stiffness*, $\boldsymbol{K}_t \in \mathbb{R}^{3\times 3}$, the *rotational stiffness*, $\boldsymbol{K}_o \in \mathbb{R}^{3\times 3}$, and the *coupling stiffness*, $\boldsymbol{K}_c \in \mathbb{R}^{3\times 3}$. In the case of a symmetric coupling stiffness matrix, there is a maximum decoupling between rotation and translation. Accordingly, there is a *center of stiffness* at the point where the bodies' reference frames coincide. Similarly we can define a *center of compliance* if the compliance matrix, $\boldsymbol{C} = \boldsymbol{K}^{-1}$, has symmetric off-diagonal blocks. If the center of stiffness and the center of compliance coincide, there is no coupling between translation and rotation. Meaning that a relative translation of the bodies results in a pure force along an axis, through the center off stiffness. And also meaning that a relative rotation would lead to a pure torque about an axis through the center of stiffness.

Geometrically consistent active stiffness

To realize a geometrically consistent 6-DOF active stiffness, it is required to find an appropriate control law with the correct proportional control action. In the case of having a finite displacement of the end-effector frame Σ_e in respect to the desired frame Σ_d, the resulting control action can be interpreted as the elastic wrench applied on the end-effector. As to guarantee asymptotic stability in terms of Lyapunov, there is need for a fitting potential elastic energy function. The expression for mechanical stiffness in (13) is simplified by assuming that the coupling stiffness matrix is zero. The simplification means that the potential elastic energy can be expressed as the sum of translational- and rotational potential energy. The translational potential energy is defined as

$$V_t = \frac{1}{2}\Delta\boldsymbol{p}_{de}^T \boldsymbol{K}'_{Pt} \Delta\boldsymbol{p}_{de}, \tag{14}$$

with

$$\boldsymbol{K}'_{Pt} = \frac{1}{2}\boldsymbol{R}_d \boldsymbol{K}_{Pt} \boldsymbol{R}_d^T + \frac{1}{2}\boldsymbol{R}_e \boldsymbol{K}_{Pt} \boldsymbol{R}_e^T, \tag{15}$$

where $\boldsymbol{K}_{Pt} \in \mathbb{R}^{3\times 3}$ is a symmetric positive-definite matrix. By using $\boldsymbol{K}'_{Pt}$ instead of $\boldsymbol{K}_{Pt}$ in (14) the potential energy is guaranteed to be port symmetric, also in the case of finite placements [8]. The following power $\dot{V}_t$ becomes

$$\dot{V}_t = \Delta\dot{\boldsymbol{p}}_{de}^{e^T} \boldsymbol{f}_{\Delta t}^e + \Delta\boldsymbol{\omega}_{de}^{e^T} \boldsymbol{m}_{\Delta t}^e, \tag{16}$$

where $\Delta\dot{\boldsymbol{p}}^{eT}_{de}$ is the time derivative of the positional deviation, and $\boldsymbol{\omega}^{eT}_{de}$ is the error in angular velocity, both with respect to the end-effector frame. The vectors $\boldsymbol{f}^{e}_{\Delta t}$ and $\boldsymbol{m}^{e}_{\Delta t}$ are the the elastic force and moment respectively, applied to the end-effector at a finite position displacement $\Delta\boldsymbol{p}^{e}_{de}$. Expressed in the base frame they are computed as

$$\boldsymbol{f}_{\Delta t} = \boldsymbol{K}'_{Pt}\Delta\boldsymbol{p}_{de}, \qquad \boldsymbol{m}_{\Delta t} = \boldsymbol{K}''_{Pt}\Delta\boldsymbol{p}_{de}, \tag{17}$$

with

$$\boldsymbol{K}''_{Pt} = \frac{1}{2}\boldsymbol{S}(\Delta\boldsymbol{p}_{de})\boldsymbol{R}_{d}\boldsymbol{K}_{Pt}\boldsymbol{R}^{T}_{d}, \tag{18}$$

where $\boldsymbol{S}(\Delta\boldsymbol{p}_{de}) \in \mathbb{R}^{3\times 3}$ is the skew matrix of the positional displacement vector $\Delta\boldsymbol{p}_{de} \in \mathbb{R}^{3\times 1}$. In sum, the elastic wrench caused by a pure translational displacement is denoted

$$\boldsymbol{h}_{\Delta t} = \begin{pmatrix}\boldsymbol{f}^{T}_{\Delta t} & \boldsymbol{m}^{T}_{\Delta t}\end{pmatrix}^{T}. \tag{19}$$

The rotational potential energy in turn can be defined as

$$V_o = 2\boldsymbol{\epsilon}^{eT}_{d_e}\boldsymbol{K}_{Po}\boldsymbol{\epsilon}^{e}_{de}, \tag{20}$$

where $\boldsymbol{\epsilon}^{e}_{de} \in \mathbb{R}^{3\times 1}$ is the vector part of the unit quaternion extracted from the rotation $\boldsymbol{R}^{e}_{d}$, taking the end-effector to the desired orientation. Similar to $\boldsymbol{K}_{Pt}$ in (18), $\boldsymbol{K}_{Po}$ is a symmetric positive-definite matrix $\in \mathbb{R}^{3\times 3}$. Since $\boldsymbol{\epsilon}^{e}_{d_e} = -\boldsymbol{\epsilon}^{e}_{d_e}$, the function V_o is port symmetric. $\dot{V}_o$ yields

$$\dot{V}_o = \Delta\boldsymbol{\omega}^{eT}_{de}\boldsymbol{m}^{e}_{\Delta o}, \tag{21}$$

with

$$\boldsymbol{m}_{\Delta o} = \boldsymbol{K}'_{Po}\boldsymbol{\epsilon}_{de}, \tag{22}$$

where

$$\boldsymbol{K}'_{Po} = 2\boldsymbol{E}^{T}(\boldsymbol{\eta}_{de}, \boldsymbol{\epsilon}_{de})\boldsymbol{R}_{e}\boldsymbol{K}_{Po}\boldsymbol{R}^{T}_{e}, \tag{23}$$

and $\boldsymbol{E}(\boldsymbol{\eta}_{de}, \boldsymbol{\epsilon}_{de}) = \boldsymbol{\eta}_{de}\boldsymbol{I} - \boldsymbol{S}(\boldsymbol{\epsilon}_{de})$. Accordingly a finite orientation displacement $\boldsymbol{\epsilon}_{de}$ produces an elastic wrench

$$\boldsymbol{h}_{\Delta o} = \begin{pmatrix}\boldsymbol{0}^{T} & \boldsymbol{m}^{T}_{\Delta o}\end{pmatrix}^{T}, \tag{24}$$

equivalent to a pure moment. The total elastic wrench can then be expressed as

$$\boldsymbol{h}_{\Delta} = \boldsymbol{h}_{\Delta t} + \boldsymbol{h}_{\Delta o}, \tag{25}$$

in line with (19) and (24), as a function of both a finite position- and orientation displacement. Discarding the high order infinitesimal terms, (25) yields the mapping

$$\boldsymbol{h}_e^e = \boldsymbol{K}_P \delta \boldsymbol{x}_{de}^e = \begin{pmatrix} \boldsymbol{K}_{Pt} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{Po} \end{pmatrix} \delta \boldsymbol{x}_{de}^e. \tag{26}$$

This shows that $\boldsymbol{K}_P$ represents an ideal spring with respect to the frame of the end-effector, the origin being the center of stiffness. Also in the case of large displacements, the geometrical and physical meaning of $\boldsymbol{K}_{Pt}$ and $\boldsymbol{K}_{Po}$ remains the same. Meaning e.g. that the upper left element in $\boldsymbol{K}_{Pt} \in \mathbb{R}^{3\times 3}$ will always decide the stiffness in x-direction with respect to Σ_e.

2.5 Impedance Control

In order to achieve a desired *dynamic* behavior, stiffness control is not sufficient. The problem extends to achieving a desired second-order system with 6-DOF, characterized by a certain mass, damping and stiffness. Achieving such a system, known as mechanical *impedance*, can be tedious as the dynamics depends on the nonlinear and coupled ones of the manipulator [8]. By doing the acceleration-resolved approach associated with motion control, one aim to decouple and linearize the nonlinear robot dynamics at the acceleration level.
Also, in the presence of a force and torque sensor (FT sensor) measuring $\boldsymbol{h}_e$, a more complete form of impedance control can be implemented by enabling inertia shaping, meaning that you are able to determine the apparent inertia of the closed system. By casting the control law

$$\boldsymbol{h}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\boldsymbol{\alpha} + \boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{\eta}(\boldsymbol{q}) + \boldsymbol{h}_e, \tag{27}$$

into the dynamic model in (7), it reduces to $\dot{\boldsymbol{v}}_e = \boldsymbol{\alpha}$, $\boldsymbol{\alpha}$ being the control input with the meaning of an acceleration with respect to the base frame. Identifying $\dot{\boldsymbol{v}}_e = \bar{\boldsymbol{R}}_e^T \dot{\boldsymbol{v}}_e^e + \dot{\bar{\boldsymbol{R}}}_e^T \boldsymbol{v}_e^e$ with

$$\bar{\boldsymbol{R}}_e = \begin{pmatrix} \boldsymbol{R}_e & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}_e \end{pmatrix}, \tag{28}$$

choosing

$$\boldsymbol{\alpha} = \bar{\boldsymbol{R}}_e^T \boldsymbol{\alpha}^e + \dot{\bar{\boldsymbol{R}}}_e^T \boldsymbol{v}_e^e, \tag{29}$$

leads to $\dot{\boldsymbol{v}}_e^e = \boldsymbol{\alpha}^e$, with the control input $\boldsymbol{\alpha}^e$ having the meaning of an acceleration relative to the end-effector frame. Now, setting

$$\boldsymbol{\alpha}^e = \dot{\boldsymbol{v}}_d^e + \boldsymbol{K}_M^{-1}(\boldsymbol{K}_D \Delta \boldsymbol{v}_{de}^e + \boldsymbol{h}_{\Delta}^e - \boldsymbol{h}_e^e), \tag{30}$$

the closed loop expression is found to be

$$\boldsymbol{K}_M \Delta \dot{\boldsymbol{v}}^e_{de} + \boldsymbol{K}_D \Delta \boldsymbol{v}^e_{de} + \boldsymbol{h}^e_\Delta = \boldsymbol{h}^e_e, \tag{31}$$

$\boldsymbol{K}_M \in \mathbb{R}^{3\times3}$ and $\boldsymbol{K}_D \in \mathbb{R}^{3\times3}$ being symmetric positive-definite matrices, $\Delta\dot{\boldsymbol{v}}^e_{de}$ and $\Delta\boldsymbol{v}^e_{de}$ being the error in acceleration and velocity, and $\boldsymbol{h}^e_\Delta$ being the elastic wrench defined in (25), all relative to Σ_e. $\boldsymbol{K}_M$, having the meaning of the apparent inertia matrix, is like $\boldsymbol{K}_P$ and $\boldsymbol{K}_D$ an adjustable parameter. With no external wrench working on the manipulator, this control scheme guarantees that that the end-effector frame Σ_e asymptotically follows the desired frame Σ_d. In the presence of interaction, the compliant behavior of the end-effector is described by (31), limiting the contact wrench at the expense of a finite displacement in position and orientation.
However, this control scheme is dependent on the availability of precise and steady measurements of the external wrench $\boldsymbol{h}_e$. If that is not the case, one has to avoid inertia shaping, choosing the apparent inertia to be equal to the natural cartesian inertia of the robot. The simplified control law

$$\boldsymbol{h}_c = \boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{v}}_d + \boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}})\boldsymbol{v}_d + \boldsymbol{K}_D\Delta\boldsymbol{v}_{de} + \boldsymbol{h}_\Delta + \boldsymbol{\eta}(\boldsymbol{q}), \tag{32}$$

still achieves a desired impedance behaviour through the resulting closed loop expression

$$\boldsymbol{\Lambda}(\boldsymbol{q})\Delta\dot{\boldsymbol{v}}_{de} + (\boldsymbol{\Gamma}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{K}_D)\Delta\boldsymbol{v}_{de} + \boldsymbol{h}_\Delta = \boldsymbol{h}_e. \tag{33}$$

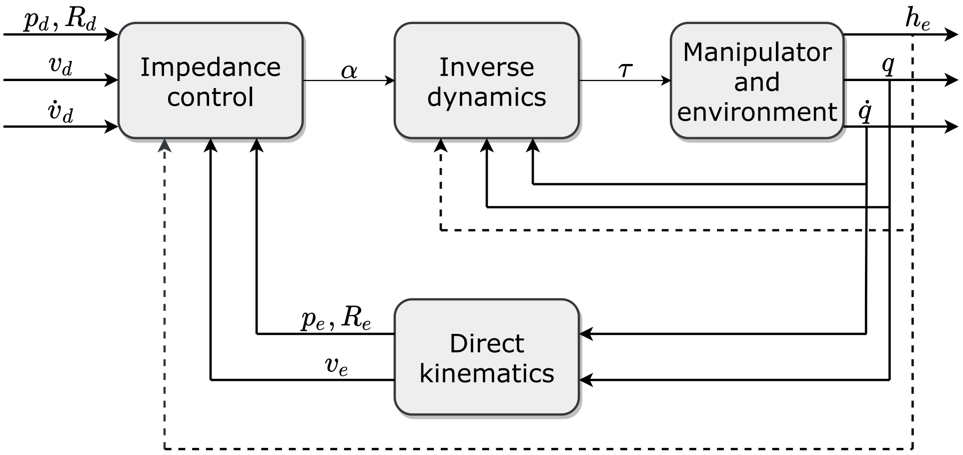


Figure 2: Block diagram of the suggested impedance controllers. The inertia shaping controller being the only one having a feedback of the external wrench $\boldsymbol{h}_e$ (dotted lines)

2.6 Variable Impedance Control

The big drawback of using the standard impedance control is its lack of adaptability. Even with a lot of effort on finding the optimal values for stiffness, inertia and damping parameters, it is highly difficult to frame a controller that performs well in diverse environments or even at different areas of the same environment. Usually it is ideal to decide in advance which kind of environment the controller is to perform optimally, most often an environment with either high or low stiffness. If a manipulator is able to adapt its stiffness and/or damping as it encounters new environments, one can ensure smoother and more predictable interactions. This is the fundamental idea behind *Variable Impedance Control (VIC)*. In use cases such as robotic ultrasound, VIC can be used to adapt the dynamical properties of the interaction between the end-effector (ultrasound probe) and the environment as it touches an area with different stiffness. One of the main advantages to such an approach is the increased ability to avoid moments of large interacting forces when the environmental stiffness increases abruptly. Also in the presence of disturbances such as movements caused by heartbeat and breathing in the ultrasound use case, VIC has the potential to optimize the manipulator's dynamics accordingly. To sum up, compared to the standard impedance control, VIC provides a control approach that enables a higher level of flexibility and stability when interacting with the environment.

VIC is a very broad approach and there is no specific recipe for how to implement it. However, you can divide VIC into two categories based on how the dynamic properties are adjusted, namely *Adaptive VIC* and *Learning-based VIC*. The adaptive VIC is recognized by its fixed adaptive update laws, and there are multiple approaches in this direction. Lee and Buss for example, implemented a variable stiffness based on previous force tracking errors [9]. A similar, yet different technique is tested by Duan et al. where the stiffness is set to zero and the damping coefficient is adjusted instead. A third approach is assessed by Liu et al. where the error in force feedback is used to adjust the compliant position of the end-effector [10]. A drawback with these methods however, is that they rely on accurate models of the system to work well [11]. Similar to adaptive methods, the learning methods can also be divided into multiple subcategories. Key to all of them is the utilization of machine learning, where the controllers aim to optimize automatically by analyzing their performance and adjusting the parameters accordingly. Different learning techniques such as *Imitation learning*, *Iterative learning* and *Reinforcement learning (RL)* are utilized. [12] and [13] are examples of research articles examining the latter, more specifically with the algorithm *Policy Improvement with Path Integrals*. [14] on the other hand relied on stiffness estimations computed from human demonstrations. However, these methods usually demand a lot of interactions to achieve good performance. Therefore, there has been an increasing focus directed at sampling efficiency in the field. An example of this is Li et al.'s development of a data-efficient RL method based on a probabilistic Gaussian Process (GP) model of the system [15].

3 Robot-hardware and software

This section aims to make the reader aware of the hardware and software that is enabling the implementations and assessments treated in Section 4 and 5. First, the basic characteristics of the robot will be presented, followed by the interfaces allowing intuitive and straight forward implementation of control.

3.1 Franka Emika Panda Robot

The Panda robot, manufactured by Franka Emika, is a collaborative lightweight robot designed to act like an agile human arm. It has 7-DOF, a payload of 3 kg, a weight of 18 kg and a radius (reach) of 855 mm [16]. Through its joint torque sensors and forward kinematics, it is capable of calculating an estimate of the external wrench at the end-effector. In addition to the robot arm itself, the setup also consists of an external activation device, a shop floor controller and a shared network switch, connecting everything to a workstation running the respective control-scripts (Fig. 3).

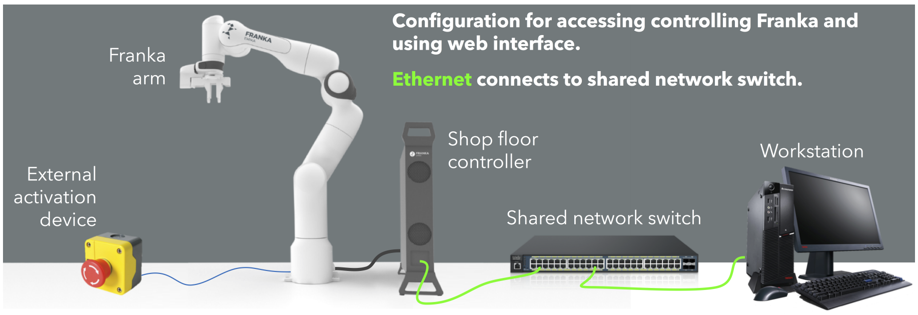

Configuration for accessing controlling Franka and using web interface.
Ethernet connects to shared network switch.
Franka arm
External activation device
Shop floor controller
Shared network switch
Workstation

Figure 3: Setup for controlling the Panda robot using web interface [4]

3.2 Robot Operating System

Robot Operating System (ROS) is a flexible framework for writing robot software. More specifically it is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behaviour across a wide variety of robotic platforms [17]. In sum, these components provide services such as hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes and package management [18]. Today, there exists several versioned sets of ROS packages, known as ROS *Distributions*. In this setup, ROS Melodic Morenia is the one being used, released in May 2018. This Distribution is primarily targeted at Ubuntu 18.04 (Bionic Beaver).

3.3 Franka Control Interface

Franka Control Interface (FCI) allows a fast and direct low-level bidirectional connection to the arm and hand [5]. The schematic overview of this interaction is depicted in Fig. 4. This figure also shows the components that make up the FCI, namely the C++ library *libfranka* and the ROS integration *franka_ ros*. Out of the two, libfranka is the one providing low-level control of the Franka robots. It is handling the network communication with the control unit (Fig. 4) and is also providing interfaces to easily execute commands, read robot states and access the model library [19]. *franka_ ros* on the other hand, is a metapackage for all Franka Emika ROS packages, providing the robots with ROS support [20]. In addition to its ROS contribution, it also ensures support for the motion planning framework MoveIt!. Also *franka_ description*, a collection of URDF models and 3D meshes of the Franka robots, is a part of this component, having a utility that extends beyond ROS [21].

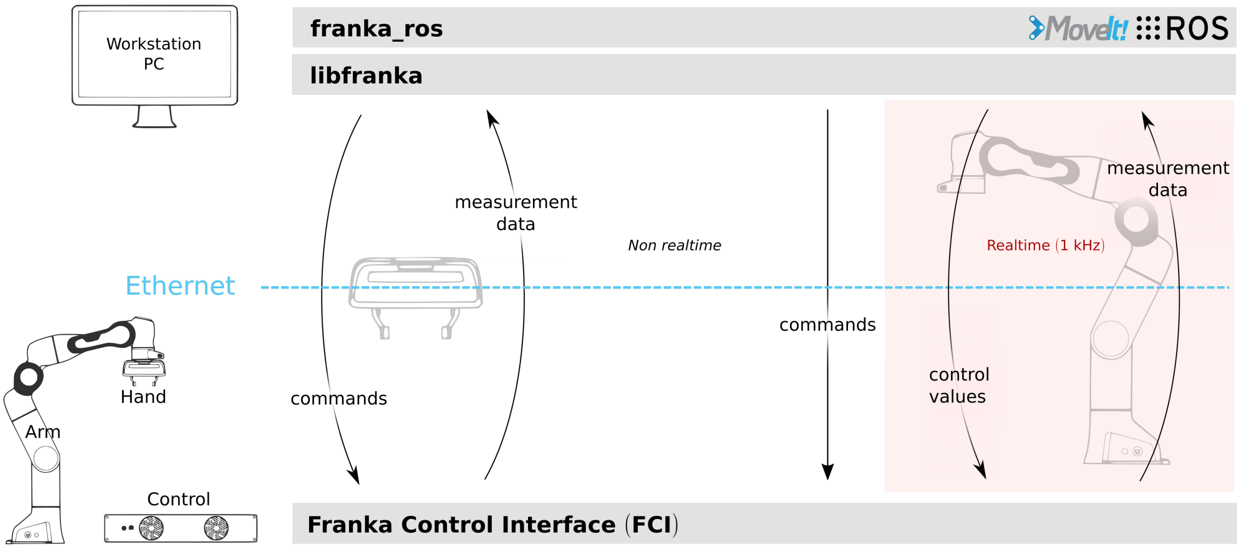


Figure 4: Schematic overview of the FCI's workflow [5]

3.4 Extensions to FCI

With the goal of making it easier and more intuitive to make controllers for the Panda robot, several independent developers have created their own, extended interfaces. Saif Sidhik, behind the github account "justagist" is providing such an interface. His repositories *franka_ ros_ interface*, *panda_ robot* and *panda_ simulator* [22] are built upon the FCI, and make up a unified interface for controlling the Franka Emika Panda robot (Fig. 5).

franka_ ros_ interface

Out of the three repositories, *franka_ ros_ interface* is the one working closest to the FCI. In short, it is a ROS package extending the *franka_ ros* to expose more information about the robot, and providing low-level control of the robot using ROS and Python API. [23]. To do so, it is dependent on the package *franka_ panda_ description*, operating as an extension of *franka_ ros*' package *franka_ description*. This package is providing the dynamics parameters for the robot arm and is also providing support for simulation in Gazebo.

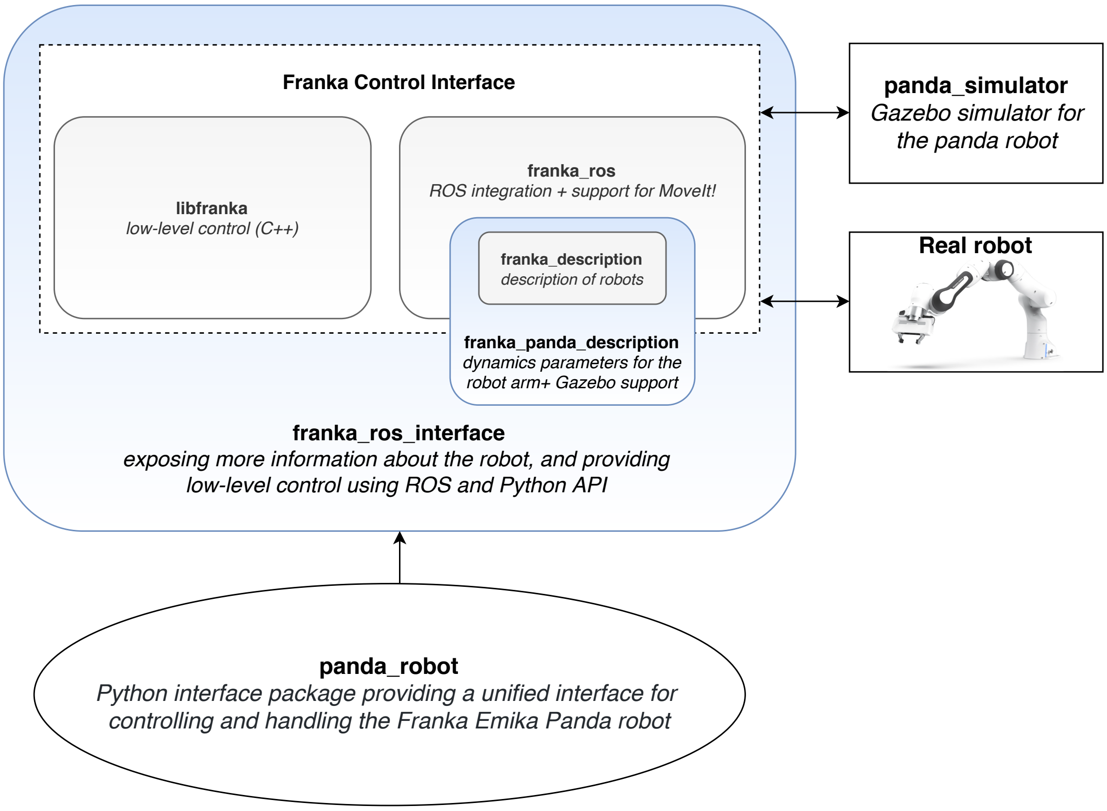

Franka Control Interface
libfranka
low-level control (C++)
franka_ros
ROS integration + support for MoveIt!
franka_description
description of robots
franka_panda_description
dynamics parameters for the robot arm+ Gazebo support
franka_ros_interface
exposing more information about the robot, and providing low-level control using ROS and Python API
panda_simulator
Gazebo simulator for the panda robot
Real robot
panda_robot
Python interface package providing a unified interface for controlling and handling the Franka Emika Panda robot

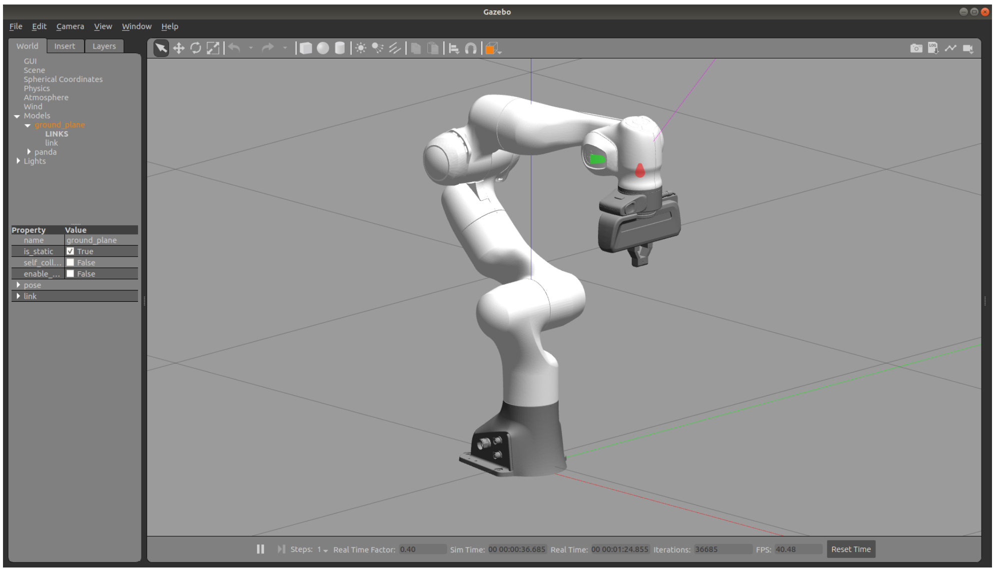
Figure 5: Illustration of dependencies and extensions among the most essential libraries

panda_ simulator

panda_ simulator is a Gazebo-based simulator for the robot, using franka_ros_interface to provide exposed controllers and real-time robot state feedback similar to the real robot when using FCI's *franka_ ros* package [24]. The package also has support for MoveIt! planning.

panda_ robot

panda_ robot is a Python interface package built over the *franka_ ros_ interface* package. It is combining its different classes to provide a unified interface for controlling and handling the Franka Emika Panda robot [25]. The interface also provide a direct *sim-to-real* code transfer, making it possible to run the same script both in the simulator and on the real robot.



Property	Value
name	ground_plane
is_static	☑ True
self_coll…	☐ False
enable_…	☐ False

Steps: 1 Real Time Factor: 0.40 Sim Time: 00 00:00:36.685 Real Time: 00 00:01:24.855 Iterations: 36685 FPS: 40.48

Figure 6: Screenshot of the running Gazebo simulator

4 Implementation

In this section, the implementation of the impedance controller is explained along with the details of executing the controller both in simulation and on the real robot.

4.1 Impedance controller implementation

As a part of this thesis, two different impedance controller approaches have been implemented. The first one is based on the inertia shaping control law in (27) and the second one on the simplified form in (32), both described in Section 2.5. In both cases, the joint actuators, under torque control mode, are fed with the control input, $\boldsymbol{\tau} = \boldsymbol{J}^T \boldsymbol{h}_c$, with $\boldsymbol{h}_c$ calculated as in (27) and (32) respectively. The desired motion, consisting of $\boldsymbol{v}_d$, $\dot{\boldsymbol{v}}_d$, $\boldsymbol{p}_d$ and $\boldsymbol{R}_d$, are chosen according to the task at hand, and serves as the input to the controller (Fig. 2). Through the python-interface described in Section 3.4, the states of the manipulator are orderly fetched, including the joint space inertia and compensation for Coriolis effects. With the exception of the desired motion, the stiffness $\boldsymbol{K}_P$, the damping $\boldsymbol{K}_D$ and also the apparent inertia $\boldsymbol{K}_M$ (in the case of inertia shaping), all parameters of the control law are purely state-dependent. This leaves $\boldsymbol{K}_P$, $\boldsymbol{K}_D$ and potentially $\boldsymbol{K}_M$ for tuning. Their values have been tuned by investigating the end-effector's dynamical properties with respect to the task at hand, tested both in simulation and in experiments.

4.2 Setting up the environment

In order to test the implemented controllers in the Gazebo simulator and with the experimental setup, the necessary software and communication framework needs to be set up. This can be done in the following steps:

1. Install ROS Melodic Morenia
2. Build *libfranka* and *franka_ros*
3. Set up the real-time kernel
4. Build *panda_simulator* and *panda_robot*

As the extended interface from Section 3.4 has support for ROS Melodic Morenia, this is the ROS distribution we are using. Accordingly, the workstation is running Ubuntu 18.04.5 LTS (Bionic Beaver) which the ROS distribution is targeting. Both the building of *libfranka* and *franka_ros*, and the setting up the real-time kernel, needed to control the robot, is described in detail in [26]. The process of building *panda_simulator* and *panda_robot* is done according to the respective README files on Github. *franka_ros_interface* and *franka_panda_description* are both installed when performing the fourth step.

5 Results

In this section, the results from testing the implementations in simulations and experiments are presented. The section is divided into two subsections, the first dealing with test scenarios performed in simulation, and the second one dealing with experiments.

5.1 Simulations

The simulations are performed in the Gazebo-based simulator described in Section 3.4, and are intended to give an initial indication of the impedance controllers' performance.

Testing the inertia shaping impedance controller

On inspecting the readings from the estimated external wrench from joint FT sensors in the simulator, a substantial offset was observed as shown in Table 1.

Table 1: Offset in estimated external wrench at neutral position in simulation

force x	force y	force z	torque x	torque y	torque z
-2.018	-0.828	-1.500	0.201	-0.184	-0.002

The offset is deceiving the controller as an external wrench acting on the robot's end-effector. This is illustrated in Fig. 7, where the manipulator is meant to stand still in its neutral position. As the measured external wrench, $\boldsymbol{h_e}$ is perceived, the deviations from the desired pose is increasing until it starts to oscillate. During the short simulation we see that the end-effector drifts off 5, 10 and 10 cm in the respective x-, y- and z-directions. In terms of orientation, the deviation is even more significant, illustrated by the vector parts of the displacement quaternion, $\begin{pmatrix}\epsilon_1 & \epsilon_2 & \epsilon_3\end{pmatrix}^T$ in the same figure.

The simulated test was repeated when removing the offset. By subtracting the offsets from Table 1 to the measurements, the external wrench, $\boldsymbol{h_e}$, was set to zero while standing perfectly still in its neutral position. However, as Fig. 8 shows, this just delayed the unstable behaviour by 0.2 seconds. The offset is found to be a function of the manipulator's position and orientation. As soon as the manipulator leaves the neutral pose, there is a mismatch between the predetermined subtracted value and the true offset. Additionally, the wrench applied from the controller itself is causing the sensor to falsely perceive external wrench. As the controller is acting on these nonfactual external forces and moments, it is amplifying the disturbances and leading to instability. The small noisy measurements is causing the end-effector to leave its stable position.

The evaluation was repeated where forces and torques with an absolute value lower than 0.05 N and 0.05 Nm respectively, were set to 0. The result of this is shown in Fig. 9. By inspecting the y-axis it can be seen that the deviation from the desired pose now is negligible. Disregarding the insignificant values from the sensor is indeed making the passive system stable. Nevertheless,

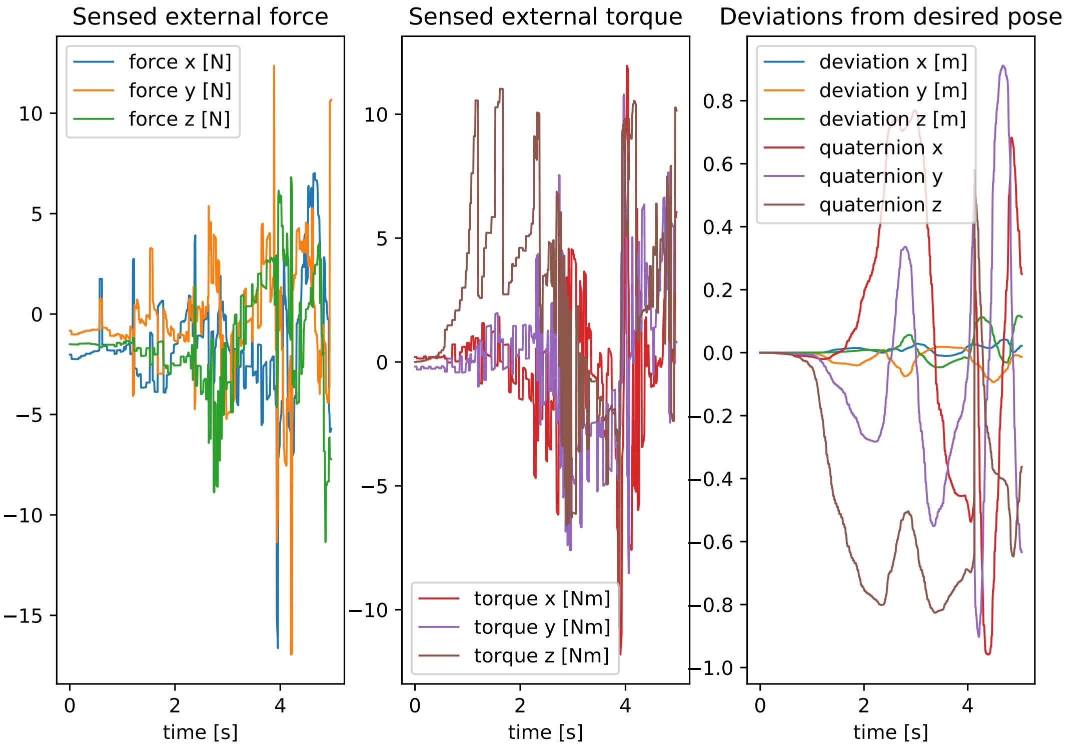


Figure 7: Falsely perceived external forces and torques leading to instability

at two occasions the simulated sensor randomly reads a significant external wrench just for an instant, causing some ripple effect in the pose the following seconds. Also, subtracting the offset at the measured wrench only fits well at the manipulator's neutral pose. Once the manipulator is manually perturbed from its neutral position, the system perceives a different environmental wrench. This still leads to instability, illustrated in Fig. 10, where an external force of 10 N in the x-direction is applied on the end-effector for approximately 1.5 second.

Testing the simplified impedance controller

The result of applying a force of 10 N in each of the three axes using the simplified impedance controller is shown in Fig. 11. Unlike the inertia-shaping controller, it is providing a stable response to the environmental forces. Additionally, it is close to maintaining the desired orientation while experiencing external forces. As the forces ceases, the end-effector pose converges back to the desired pose in a controlled manner. At the end of the sequence depicted in Fig. 11, the displacement is recorded to be 0.038 mm, 0.096 mm and 0.15 mm, in the x-, y- and z-direction respectively. However, the positional deviations in x and z appear to be rather coupled. When the end-effector is pushed 25 cm upward, it is also moving well 10 cm along the x-axis.

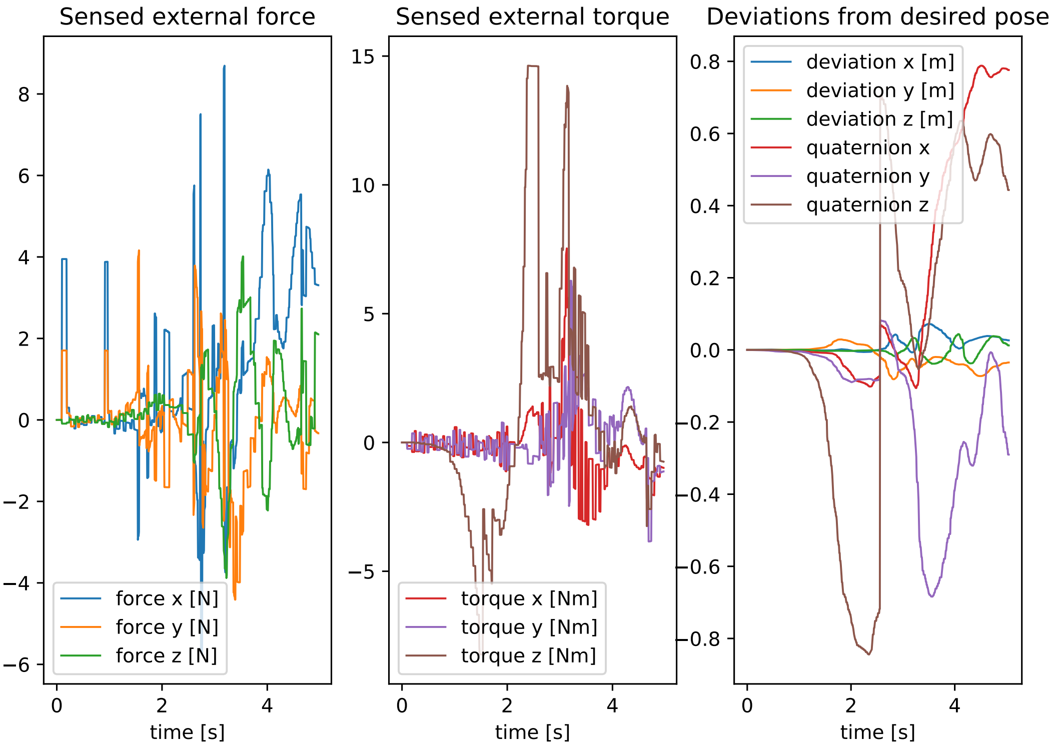


Figure 8: Falsely perceived external forces and torques leading to instability despite removing the offset at neutral position

5.2 Experiments

In order to avoid the issues of instability related to the inertia shaping controller, both the experiments in this section have been performed with the simplified impedance controller, where inertia shaping is avoided.

Applying external forces

In this experiment, the impedance controller is fed with a constant desired pose $\boldsymbol{x}_d$, while being exposed to manually applied forces, similar to the simulated test depicted in Fig. 11. The purpose of the experiment is to investigate the resulting dynamics of the compliant behavior. An excerpt of the experiment is shown in Fig. 12, and a recording of the full experiment is available at `https://www.youtube.com/watch?v=PpFfKHM4A1M&feature=youtu.be&ab_channel=akhilsAnand`. The plots in Fig. 12 specifically show the manipulator's response to four cases of external wrench, applied at regular intervals. Unlike the simulated case, the applied wrench is not a pure force along the respective axes of the end-effector's frame Σ_e, illustrated by the sensed external wrench in Fig. 12. Nevertheless, the figure shows the results of applying wrenches sequentially, moving it somewhat along the axes of x, y, z and then z again. As in simulation, the controller is providing a stable response to the external wrench, providing a compliant and

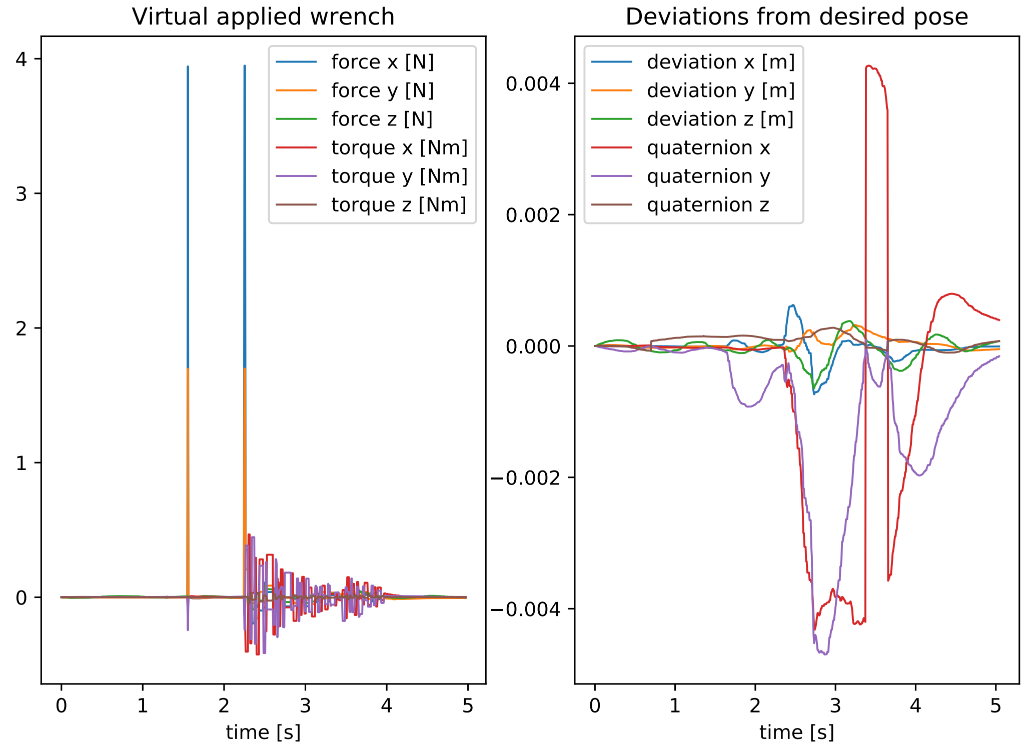


Figure 9: The measured external wrench and the resulting deviation from the desired pose of the robot when subtracting the offset at neutral position and disregarding values lower than 0.05 N and 0.05 Nm. A minor deviation occur because of random noise in the estimated external wrench

controlled behavior. However, there are significant deviations at steady state, highlighted in Fig. 13. Most notable there is a steady state error of 1.85 cm in z-direction.

Establishing compliant contact

In this experiment, the robot is expected to establish contact with a dummy of a human torso. The dummy is placed on a table in close proximity to the robot, and the desired pose of the end-effector is set to make it come in contact with its rib cage, approximately one inch away from its starting position. Also, to make it easier to maintain contact, the manipulator's gripper is equipped with a rubber-ball Fig. 14. The desired position of the end-effector is chosen to be just below the surface of the dummy, causing some positional deviation and environmental wrench at steady state, depicted in Fig. 15. The plots suggests that the impact occur after 0.3 seconds which is also the moment in time with the highest measured interaction forces. The biggest, most notable measurements being the forces in the y- and z-direction and the torque in x. However, these values does not exceed 1.3 N, 3.5 N and 1.7 Nm respectively.

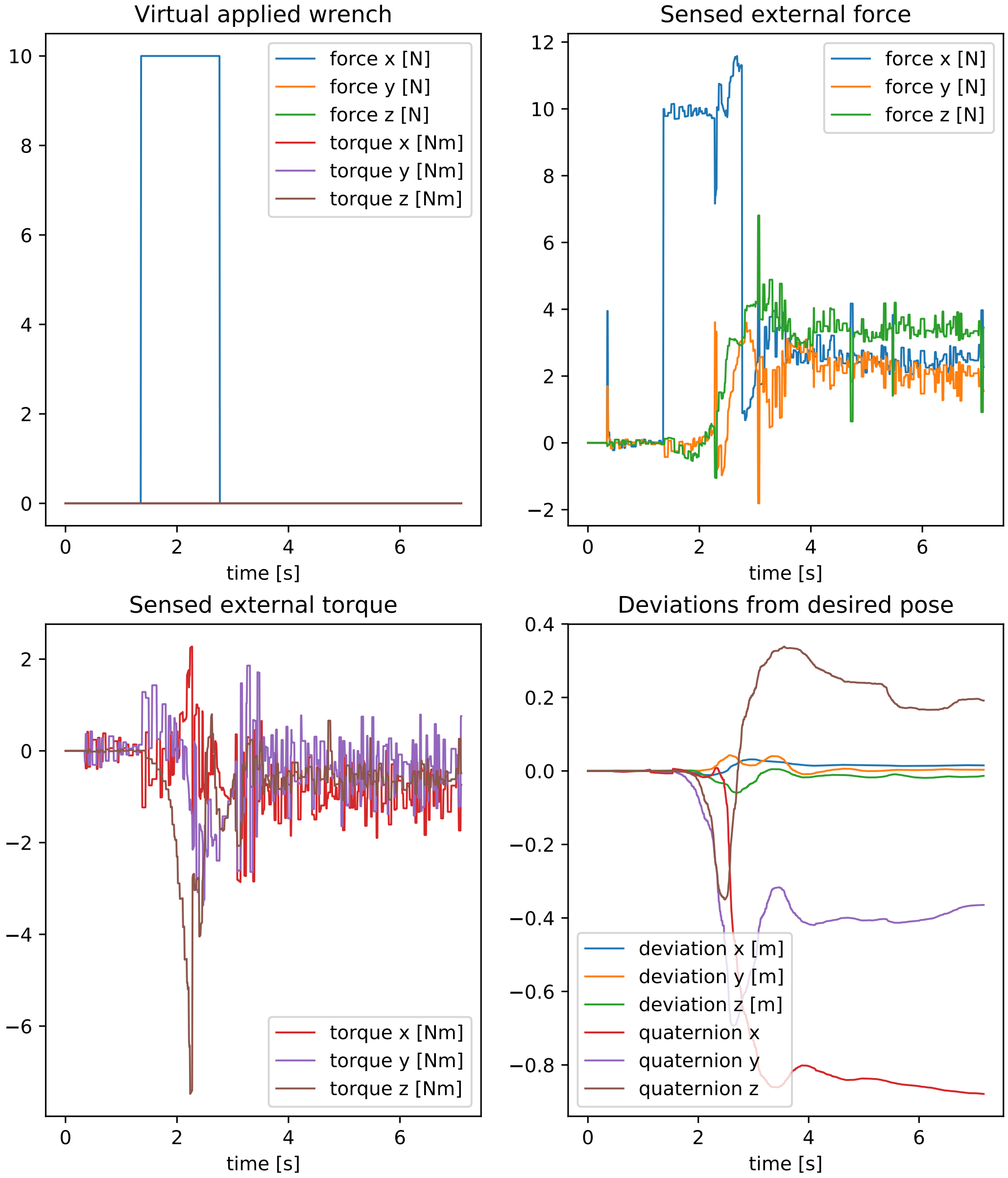


Figure 10: Plots showing the effect of perturbing the end-effector pose away from its narrow region of stability. Less than half a second after the force is applied to the manipulator, the end-effector leaves its desired orientation

The stiffness (34) and damping (35) of the controller are chosen to limit the impact-forces.

$$\boldsymbol{K}_P = \begin{pmatrix} \boldsymbol{K}_{Pt} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{Po} \end{pmatrix}, with\ \boldsymbol{K}_{Pt} = \begin{pmatrix} 55 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 22 \end{pmatrix}\ and\ \boldsymbol{K}_{Po} = \begin{pmatrix} 50 & 0 & 0 \\ 0 & 50 & 0 \\ 0 & 0 & 50 \end{pmatrix}, \tag{34}$$

$$\boldsymbol{K}_D = \boldsymbol{I} \in \mathbb{R}^{6\times6}. \tag{35}$$

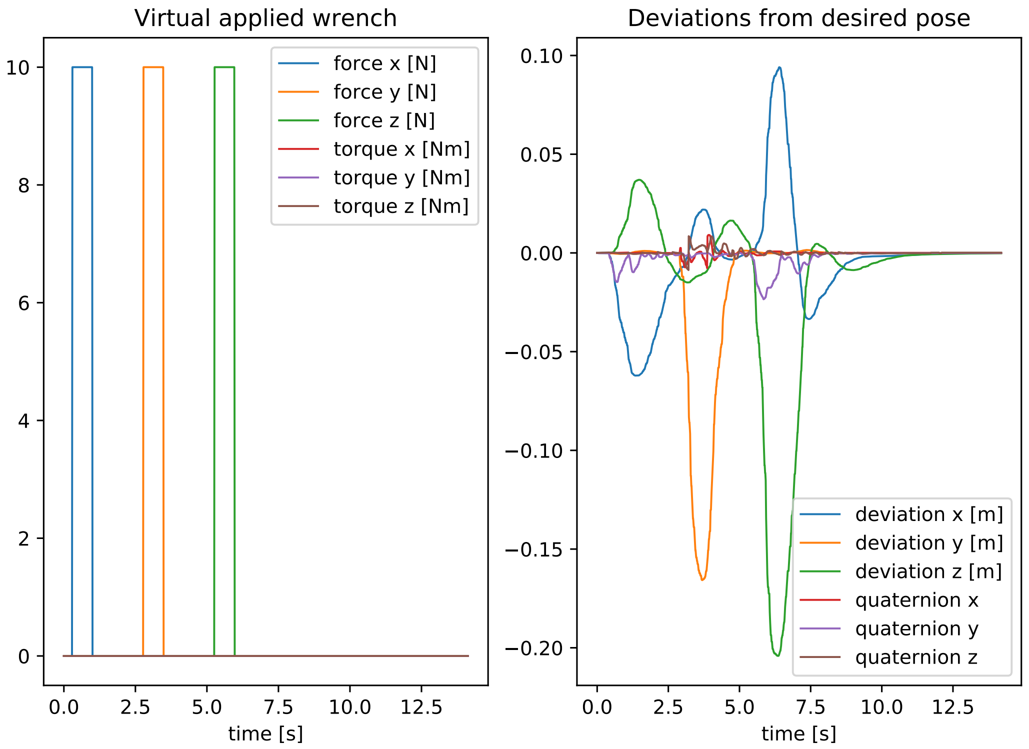


Figure 11: The applied external wrench and the resulting deviation from the desired pose of the robot using the simplified impedance controller. The force applied in the x-direction causes the end-effector to also deviate noticeable in the z-direction and vice versa

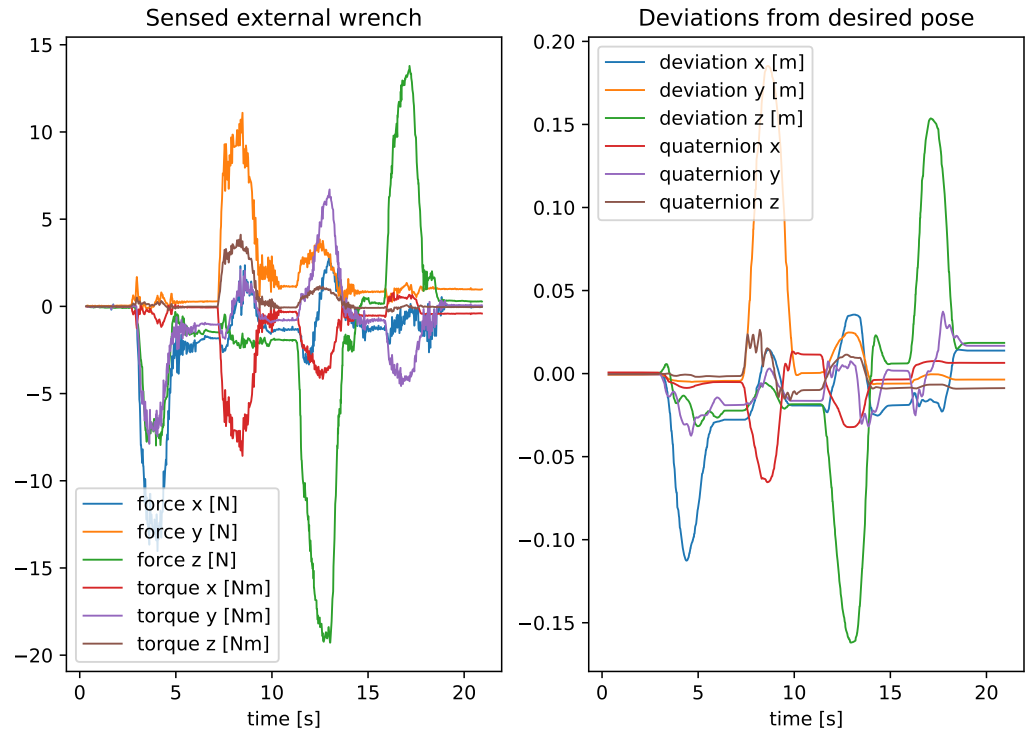


Figure 12: Sensed external wrench and deviation from desired pose, illustrating a compliant and controlled response to external forces and moments. The result also show that there are significant deviations at steady state

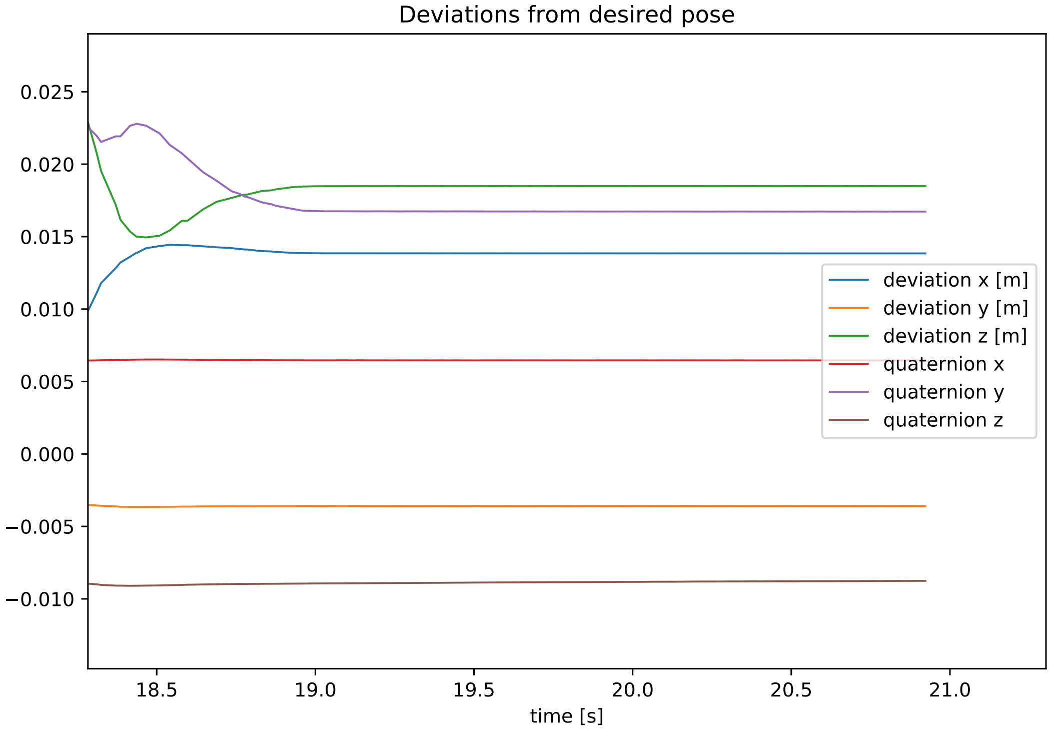


Figure 13: Steady state error at the end of the time sequence depicted in Fig. 12

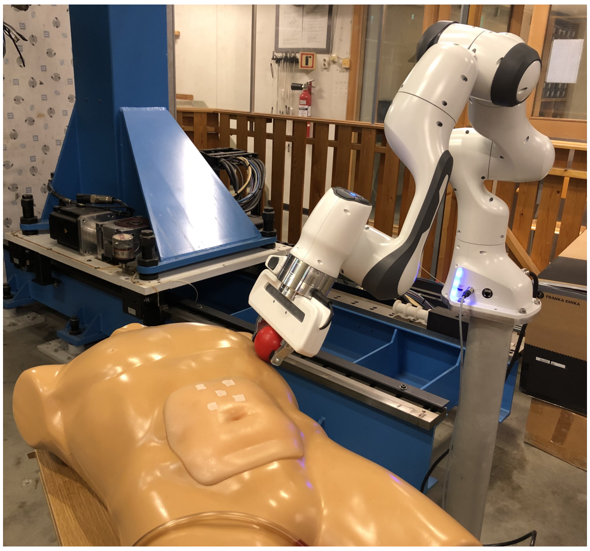


Figure 14: Setup for the experiment where the Panda robot establishes a soft contact with a dummy of a human torso

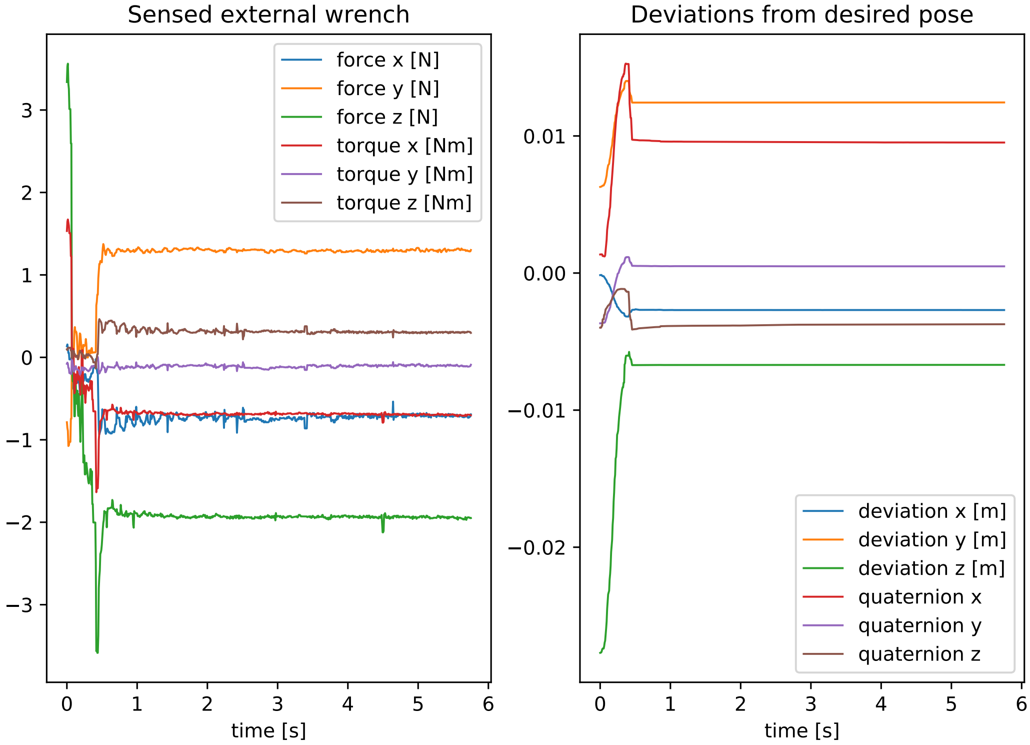


Figure 15: Measured wrench and pose-deviation as the Panda robot comes in contact with a dummy of a human torso. The stiffness (34) and damping (35) of the controller are chosen so that they ensure limited impact-forces

6 Discussion and further work

This section is dedicated to discuss the results described in the previous section, and to present the future scope of the master thesis.

6.1 Discussion

The results from the simulations is proving the inertia shaping impedance controller to be unstable when using the raw estimates of the external wrench, $\boldsymbol{h}_e$. The result of the initial test, depicted in Fig. 7, shows the effect of falsely perceiving external forces and moments. A more correct reproduction of the wrench is a prerequisite for this approach. This will have to involve zeroing out the measured offsets caused by the current state of the manipulator. For example, if the end-effector is replaced by an ultrasound probe, the change of weight and resulting wrench needs to be taken into account as well. By performing a calibration of the estimated external wrench, it is believed that we can avoid the instability-issues experienced in Fig. 7, 8 and 10.

In addition to the offset related issues, Fig. 8 and 9 highlighted the effect of having noisy measurements of $\boldsymbol{h}_e$. Fig. 8 shows that pure noise from recordings is enough to make the manipulator deviate from its desired position and orientation. Additionally, the rippling effect of the measurements in Fig. 9 also show how the estimated values are affected by the manipulator's movement. To mitigate these effects, we believe that a low-pass filter can improve performance. However, the experiment of establishing contact with the dummy of a human torso (Fig. 15) also produced spikes in the sensed external wrench, evidently correctly. If performing low-pass filtering on the estimates, one must be cautious not to suppress the true wrench.

Having a control law independent of the measured external wrench, the simplified impedance controller avoided the instability-issues related to its lack of validity. Both in simulation (Fig. 11) and in experiments (Fig. 12, 15) it is showing a stable response when interacting with the environment. As expected from an impedance controller, the deviation to its desired pose, caused by external forces, entails an applied wrench reminiscent of a spring-damper system. In Fig. 11 this is illustrated by the controlled oscillations back to the desired pose, and in Fig. 15 by the contact wrench maintained at steady state.

However, the results of applying forces along the respective axes (Fig. 11) indicated some substantial coupling in the manipulators movements along the x- and z-axis. Despite being exposed to pure forces along these respective axes, the end-effector is far from making a corresponding pure translation. In fact, the result show that displacing the manipulator 24 cm in the intended direction, also cause a displacement of 12 cm along the other axis. This is not in line with the idea of Cartesian impedance control. When having a constant desired pose, being the case for all the conducted tests and experiments, the control law of the simplified impedance controller reduces to

$$\boldsymbol{h}_c = \boldsymbol{K}_D \boldsymbol{v}_e + \boldsymbol{h}_\Delta + \boldsymbol{\eta}(\boldsymbol{q}). \tag{36}$$

The resulting closed-loop equation becoming

$$\boldsymbol{\Lambda}(\boldsymbol{q})\dot{\boldsymbol{v}}_e + (\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{K}_D)\boldsymbol{v}_e + \boldsymbol{h}_\Delta = \boldsymbol{h}_e. \tag{37}$$

For comparison the inertia-shaping impedance controller would be reduced to

$$\boldsymbol{K}_M\dot{\boldsymbol{v}}_e^e + \boldsymbol{K}_D\boldsymbol{v}_e^e + \boldsymbol{h}_\Delta^e = \boldsymbol{h}_e^e. \tag{38}$$

By inspecting (37), it seem like the unwanted behavior is caused by the inherent inertia matrix of the manipulator, $\boldsymbol{\Lambda}(\boldsymbol{q})$, and the wrench caused by centrifugal and Coriolis effects, $\boldsymbol{\Gamma}(\boldsymbol{q}, \dot{\boldsymbol{q}})$. If this assumption is correct, a properly implemented inertia-shaping impedance controller would have the manipulator behaving more in line with an ideal spring-damper system, with decoupling in x and z.

When performing the experiment of applying external forces (Fig. 12), it was also noticed that the controller requires quite a high stiffness $\boldsymbol{K}_P$ for it to be able to return back to the desired position. Increasing the ability to follow the desired trajectory accordingly leads to a reduced compliant capacity. As a result, the simplified impedance controller may not be sufficient if the task is demanding high precision in a stiff environment. It would be interesting to repeat the experiment with the inertia shaping controller, granting full design privileges of the apparent inertia of the system.

Despite the simplified impedance controllers apparent limitations, it proved to be successful in the experiment of establishing a compliant contact with a human dummy. The results in Fig. 15 show that the manipulator is made capable of following a desired trajectory while ensuring limited contact forces and moments when interacting with the environment.

The framework itself, consisting of FCI and the extensions described in Section 3.4 has been an important contributor as well. The provided Python interface makes it easier and more intuitive to write controllers for the Panda robot, using predefined functions to fetch states and to send commands to the robot. The Python API also allow writing robot software without having any particular ROS-knowledge, making it easier for beginners. The Gazebo-based simulator is also a very helpful tool when developing code for the Panda robot. Having performed the same tests in both the simulator and in the experimental setup, the simulator is proving to be very accurate, providing a very good estimate of the true performance. Because of this, and the direct *sim-to-real* transfer of code, it is a very convenient way to test the scripts prior to deploying in the robot. However, much like the estimated external wrench in the real setup, the simulated FT-sensor require calibration before it can be used directly in a control law. As the tests in Fig. 8 and 9 clearly shows, it is also quite affected by noise. In terms of compatibility, I should emphasize that the extended interface require Ubuntu 18.04. The customized interface, including the implemented controllers, is available at `https://github.com/martihmy/panda_simulator`.

6.2 Further work

In the immediate further work, we would like to investigate methods for improving the estimates of the external wrench. First by looking into potential ways of calibrating it. Then conduct tests, assessing the effect of introducing a low-pass filter, investigating if we are able to remove noise without increasing the response time or mistakenly ignore true wrenches. Subsequently, we would like to test if the inertia shaping controller is able to outperform the simplified one with respect to its limitations. The limitations being the coupled movements in x and z-direction, and the limited flexibility in terms of deciding the apparent damping and inertia of the closed-loop system (33).

The knowledge gained from this would be of great help on continuing with the planned work of the master thesis. The first objective being implementing a position based impedance controller (admittance control) with force feedback. Staying within the use case of robot-based ultrasound, the intent is to establish a desired contact force on the dummy. The second and last objective will be to implement a VIC, capable of maintaining the contact and execute a contact trajectory. VIC is considered to be a good solution because of its high level of flexibility, being able to adapt its stiffness and/or damping as it interacts with the environment. More specifically we want to implement and asses a sampling efficient learning-based VIC, having the additional ability of efficient experience based optimization.

7 Conclusion

In this thesis, the literature relevant to the field of compliant manipulation was reviewed, a framework for controlling the Franka Emika Panda Robot was presented, and so was the implementation and assessment of two fundamental approaches within impedance control. The first approach considered was an inertia shaping impedance controller, utilizing estimates of the external wrench to achieve highly specialized dynamical properties. This approach was proven to be unstable as it required high quality estimates of the external wrench, not yet available to us.

As the simplified impedance controller (avoiding inertia shaping) did not rely on the sensed external wrench, it accordingly avoided the instability-issues related to the estimate's lack of validity. Both in simulation and in experiments it maintained control when interacting with the environment. In the use case of robot-based ultrasound it also managed to establish a soft compliant contact with a dummy human torso. Despite not having a force-feedback, the inherent spring-dynamics of the controller ensured limited contact forces and moments. However, the results proved that the simplified impedance controller has limitations, having coupled movements in x- and z-direction, and requiring quite a high stiffness to achieve a satisfactory level of precision. As the inertia-shaping impedance controller in theory should overcome these limitations, it is in our interest to properly assess its performance when better wrench-estimates are available.

The development of software for the Panda robot also proved the presented framework to be very helpful. The provided Python interface makes it easier to implement controllers, and the Gazebo-based simulator is a very effective tool when developing controllers. Especially the *sim-to-real* transfer of code is a great attribute, making it a very convenient way to test scripts prior to deploying in the robot.

The framework will also be of great help when continuing the planned work of the master thesis, where we are to extend the functionality linked to the robot-based ultrasound examination use case. By sequentially developing a position-based impedance control and a VIC, the capabilities of the compliant manipulator is to be step wise expanded. The various steps include establishing contact with a desired contact force, maintaining contact despite disturbances, and executing a contact trajectory, tracking a desired contact force profile.

References

[1] Inc. The MathWorks. Urdf primer, 2020.

[2] Luigi Villani and J De Schutterm. Handbook of robotics, chapter force control, 2008.

[3] Andrea Calanca, Riccardo Muradore, and Paolo Fiorini. A review of algorithms for compliant control of stiff and fixed-compliance robots. *IEEE/ASME Transactions on Mechatronics*, 21(2):613–624, 2015.

[4] Ben Greenberg et al Revision. Franka arm, 2018.

[5] Franka Emika GmbH. Franka control interface (fci) - overview, 2017.

[6] Marialena Vagia. Romo, 2019.

[7] Kenneth J Waldron and James Schmiedeler. Kinematics. In *Springer Handbook of Robotics*, pages 11–36. Springer, 2016.

[8] Luigi Villani and Joris De Schutter. Force control. In *Springer handbook of robotics*, pages 195–220. Springer, 2016.

[9] K Lee and M Buss. Force tracking impedance control with variable target stiffness. *IFAC Proceedings Volumes*, 41(2):6751–6756, 2008.

[10] Houde Liu, Weifeng Lu, Xiaojun Zhu, Xueqian Wang, and Bin Liang. Force tracking impedance control with moving target. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1369–1374. IEEE, 2017.

[11] Fares J Abu-Dakka and Matteo Saveriano. Variable impedance control and learning–a review. *arXiv preprint arXiv:2010.06246*, 2020.

[12] Jonas Buchli, Evangelos Theodorou, Freek Stulp, and Stefan Schaal. Variable impedance control a reinforcement learning approach. *Robotics: Science and Systems VI*, pages 153–160, 2011.

[13] Freek Stulp, Jonas Buchli, Alice Ellmer, Michael Mistry, Evangelos Theodorou, and Stefan Schaal. Reinforcement learning of impedance control in stochastic force fields. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE, 2011.

[14] Fares J Abu-Dakka, Leonel Rozo, and Darwin G Caldwell. Force-based variable impedance learning for robotic manipulation. *Robotics and Autonomous Systems*, 109:156–167, 2018.

[15] Chao Li, Zhi Zhang, Guihua Xia, Xinru Xie, and Qidan Zhu. Efficient force control learning system for industrial robots based on variable impedance control. *Sensors*, 18(8):2539, 2018.

[16] WiredWorkers. Meet the franka emika panda, 2020.

[17] ROS. About ros.

[18] Amanda Dattalo. Ros / introduction, 2018.

[19] Franka Emika GmbH. libfranka, 2017.

[20] FlorianWalch. franka_ros, 2017.

[21] Franka Emika GmbH. Franka control interface documentation, 2017.

[22] Saif Sidhik. justagist, 2020.

[23] Saif Sidhik. justagist/franka_ros_interface: Controller and Interface API for Franka Emika Panda Robot Manipulator, November 2020.

[24] Saif Sidhik. panda_simulator: Gazebo simulator for Franka Emika Panda robot supporting sim-to-real code transfer, April 2020.

[25] Saif Sidhik. Panda robot, 2020.

[26] Franka Emika GmbH. Installation on linux, 2017.