ENERGY-AWARE ADAPTIVE INTERACTION CONTROL USING OFFLINE TASK-BASED OPTIMIZATION IN AN IMPEDANCE CONTROL FRAMEWORK

B. (Bart) Gerlagh

MSC ASSIGNMENT

Committee:
prof. dr. ir. S. Stramigioli
dr. ir. F. Califano
dr. ir. W. Roozing
dr. ir. W.B.J. Hakvoort

November, 2020

058RaM2020
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Impedance control is a common form of robot control where interaction is important. A key question posed in this thesis is how the impedance can be chosen in a structured manner given the robot's task and dynamics. We exploit impedance control ideas and make them optimal in time resulting in an open-loop control action which mimics time-varying Cartesian impedance control. We utilize energy tanks to recover passivity, in view of safety. In addition, we propose an iterative feed-forward adaptation law to account for model variations from the nominal plant using open-loop control. Simulation studies on a 5-DoF robot show efficacy of the methods by successfully performing an energy-aware and task-based peg-in-hole task using open-loop control with minimal feedback.

Contents

Chapter 1

Introduction

1.1 General introduction

Nowadays, when people talk about robotics they refer to the automated machines that run our industrialized world. However, robotics dates back to many centuries ago. One can find plenty of examples of ancient tales and stories in which people are intrigued by automated machines and artificial intelligence. With the catalyst called the ’industrial revolution’ , development of automated machines accelerated. Today, robotics and automated machines characterise modern society and the manufacturability of everything. Exploding research into artificial intelligence together with the exponential growth of more computing power indicate that robotics will most likely be a more integral component of our future lives.
Different notions are used to refer to a robot or automated machine. Most frequently (and also used in this thesis) are, ’robot’, ’manipulator’ or simply ’system’. In this thesis the word ’manipulator’ is often used as a reference for an industrial robot/robotic arm. A manipulator basically ’manipulates’ the world around him by performing tasks. (Industrial) examples are: picking and placing of objects, welding, grinding and performing measurements Furthermore, robots are also capable of working in environments which are dangerous to humans. In an industrial setting, those tasks are often periodic which means that the task is performed over and over again. Robots are not only found in industry. The field of so called ’service’ robots increases as well, where robots should also capable be of interacting with humans.
From a technical perspective, robotics face two interesting fields. The first concerns the question, *”How do they move?”*, this is the field of ’dynamics’. The second concerns the question, *”How can we make robots move in a desired way to perform useful tasks?”* , this is is the field of ’control’. Conventional control theory roughly distinguishes two types, motion control (position/velocity) and force/torque control. Conventional control theories do not consider control from an interaction point of view. Many modern control techniques focus on controlling interactions, often cast in the form of impedance or admittance control. This framework forms the basis for this thesis and will be decomposed in the following sections.

1.2 Previous work

1.2.1 The need for impedance control

Neville Hogan is considered to be the founding father of impedance control. His contribution was published in a threefold paper [1–3] in 1985. To this day, the work of Hogan is used as reference for impedance controlled applications. The general idea is that when a robot interacts with its environment, there is a power exchange with the environment. In that case, the environment dynamics will be part of the total system. Robust control is a type of control that deals with unknown model variations or disturbances.
Control theory classically assumes that the nominal model of the system is a good representation of the actual dynamics up to a certain frequency [4]. The 'robustness' is then a measure on how much the actual system can change from the nominal one in order to become unstable. However, when interaction takes place, also the low frequency behaviour can change [5]. An example of this would be when a mass interacts with an elastic element, it will create an additional mode to the (possibly low frequency) dynamics. Because of this, control using traditional methods will not be sufficient when interaction takes place.
In a world where robots get more integrated in daily life, the need for safe yet performance based robotics increases. Especially in situations where robots and humans are working side by side. Examples of so called 'collaborative robots' (also called cobots) can for example be found in industry where a robot collaborates with the human. In this case, a robot can hand small objects or tools to the human. For these type of situations, is not possible to put a cage around the robot. The safety has to be programmed a priori and embedded in the control architecture of the robot. Stability is not guaranteed when interacting robots are controlled with some traditional methods. An example of this is given later.
To summarize, impedance control becomes useful when interaction takes place and were traditional methods fail to maintain stability.

1.2.2 What is impedance? A port based view

Before talking about impedance 'control', it is necessary to understand what 'impedance' actually is. A mechanical impedance defines the relationship between motion (position, velocity and derivatives) and resulting force. This does not need to be linear. An example for a mechanical mass with a linear impedance is given below.

Example: Impedance of a simple mass

The equation of motion for a mass is written as,

$$F = m\dot{v} \tag{1.1}$$

where F is the force, m is the mass and v the velocity. The transfer function from force to velocity can be defined by performing the Laplace transform on the former equation . This is the impedance of the mass:

$$Z_{mass}(s) = H_{F/V} = ms\,. \tag{1.2}$$

In this case is $Z(s)$ the impedance notation and s the complex variable.

An 'impedance' does not belong only to the mechanical domain. In all other physical domains such as the electrical, magnetic or hydraulic domain, an impedance can be defined. In order to define an impedance domain independently, the power conjugated variables come in to play. The power conjugated variables are called effort and flow. They are mathematical members of dual vector spaces. It means that effort and flow take different physical aspects dependent on the domain. For all domains it holds that the duality between effort and flow equals the power:

$$P = e^{\top} f \,. \tag{1.3}$$

As power is domain independent it allows to connect different physical domains on the basis of power-ports. The main modelling approach for this is done with bond graphs. The general idea of bond graphs is that dynamic systems can be modelled with basic elements which share analogy between domains. The fundamentals of bond graphs plus a modelling example are given in appendix Chapter A. If one is familiar with the power conjugated variables, an impedance is a system with an effort-out causality [1]. The dual of the impedance is called the admittance, it has flow-out causality. If an impedance is coupled to another system, the other system is by definition an admittance. This is displayed in Fig. 1.1.

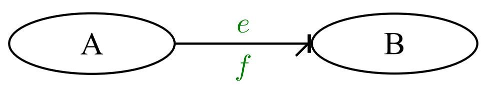


Figure 1.1: A is called the impedance, B is called the admittance. A has effort-out causality, indicated by the vertical bar in the direction away from A.

1.2.3 Basic impedance control

In classical control, two types of control can be distinguished: motion control (position/velocity) and force/torque control. For both techniques, a reference is given in the form of a (time-varying) position or force and the goal is to let the robot follow this reference as closely as possible. With impedance control it is different. With impedance control, the goal is to achieve a certain impedance. Or in other words, to achieve a certain relation between effort and flow.
The general multi-DoF equation of motion of a mechanical system is given as [6],

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{\tau} \tag{1.4}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ contains the Coriolis and centrifugal terms and $\frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}}$ represents the torques caused by gravity. $\boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}})$ denotes all other torques induced by the dynamics of the robot for example friction or stiffness. $\boldsymbol{\tau}$ represents all input torques and is sometimes split up into the control torques and the external torques:

$$\boldsymbol{\tau} = \boldsymbol{\tau_u} + \boldsymbol{\tau_e} \,. \tag{1.5}$$

In control theory, the goal is to choose the control torques $\boldsymbol{\tau_u}$ such that the robot behaves in a desired way. Choosing the control torques from an 'impedance control' point of view can be either done in joint space or in the Cartesian space. Most mathematical operators used in the sequel are explained in appendix Section D.2.

Joint space impedance control

The basic impedance controller in the joint space is obtained by shaping the effective potential of the robot. An arbitrary potential energy function $V_c(\boldsymbol{q})$ can be added to the potential of the robot by choosing the control torques such that

$$\boldsymbol{\tau_u} = -\frac{\partial V_c(\boldsymbol{q})}{\partial \boldsymbol{q}} \,. \tag{1.6}$$

By substituting this control law into the dynamics in Eq. (1.4) the following dynamics are obtained [6]:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \frac{\partial V(\boldsymbol{q}) + V_c(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{\tau_e} \,. \tag{1.7}$$

The question remaining is then how to choose $\boldsymbol{V_c}(\boldsymbol{q})$. In most cases one wants to compensate for the gravity. Furthermore if a desired joint position is given by $\boldsymbol{q}^*$ the shaped potential should have a minimum in $\boldsymbol{q}^*$. This is normally done by choosing a quadratic term for the potential. Together with the gravity compensation, the chosen potential denotes

$$V_c(\boldsymbol{q}) = -V(\boldsymbol{q}) + \frac{1}{2}(\boldsymbol{q} - \boldsymbol{q}^*)^\top \boldsymbol{K_q}(\boldsymbol{q} - \boldsymbol{q}^*) \,. \tag{1.8}$$

The second term is also recognized as the potential energy of a spring-like element with spring constant $\boldsymbol{K_q}$. This spring will induce convergence of $\boldsymbol{q}$ to $\boldsymbol{q}^*$. In addition, the control law can be extended with a dissipative term which acts on the joint velocities:

$$\boldsymbol{\tau_u} = -\frac{\partial V_c(\boldsymbol{q})}{\partial \boldsymbol{q}} - \boldsymbol{B_q}(\boldsymbol{q})\dot{\boldsymbol{q}} \,. \tag{1.9}$$

With this control law, a virtual spring and damper are added to the joints of the robot. The spring has effort-out causality which means it behaves like an impedance. The goal of this impedance controller is to let the system behave like an impedance as well. $\boldsymbol{K_q}$ and $\boldsymbol{B_q}$ will induce a certain stiffness and damping at the interaction port [5].

Cartesian space impedance control

In most cases, the task is defined in the workspace. Therefore, it is convenient to do impedance control in the workspace. In addition, the realisation of joint impedance at the end-effector depends on the robots configuration which is not constant. In contrast, with Cartesian impedance control, the end-effector impedance can be defined, which is where the task takes place. [7] describes the implementation of such a Cartesian impedance controller. It makes use of screw theory which is based on the Lie algebra of rigid body motion in $SE(3)$ [8]. The fundamentals of screw theory can be found in appendix Section B.1.

Cartesian impedance control is achieved by a virtual multi-DoF spring between the end-effector frame and a second virtual frame [7]. This is shown graphically in Fig. 1.2. Consider a frame (Ψ_n) on the end effector. Its configuration is given as the homogeneous transformation matrix,

$$\boldsymbol{H}_n^0 = \begin{pmatrix} \boldsymbol{R}_n^0 & \boldsymbol{p}_n^{0,0} \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{1.10}$$

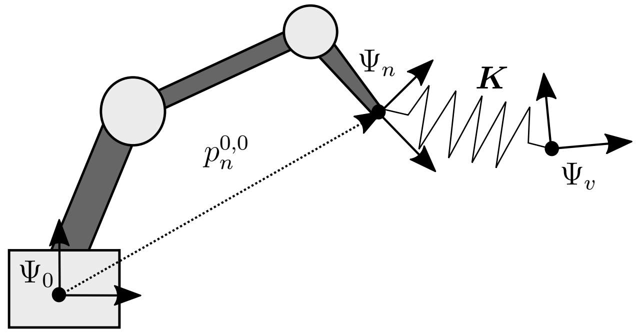


Figure 1.2: A manipulator controlled with Cartesian impedance control. $p_n^{0,0}$ represents the position vector from Ψ_0 to Ψ_n expressed in Ψ_0

and represents a point on the configuration manifold $SE(3)$, i.e. $\boldsymbol{H}_n^0 \in SE(3)$. $\boldsymbol{R}_n^0 \in SO(3)$ and $\boldsymbol{p}_n^{0,0}$ describe the orientation and position of the frame expressed in the inertial frame Ψ_0. Consider a frame Ψ_v which denotes a virtual frame corresponding to the configuration $\boldsymbol{H}_v^0$. The homogeneous matrix between the virtual and the end-effector frame is

$$\boldsymbol{H}_n^v = \boldsymbol{H}_0^v \boldsymbol{H}_n^0 = \left(\boldsymbol{H}_v^0\right)^{-1} \boldsymbol{H}_n^0 = \begin{pmatrix} \boldsymbol{R}_n^v & \boldsymbol{p}_n^{v,v} \\ \mathbf{0} & 1 \end{pmatrix}. \tag{1.11}$$

The 6-dimensional spring $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ which is connected between the end-effector and the virtual frame can be split up into the following components,

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}_o & \boldsymbol{K}_c \\ \boldsymbol{K}_c^\top & \boldsymbol{K}_t \end{pmatrix} \tag{1.12}$$

where $\boldsymbol{K}_o \in \mathbb{R}^{3\times 3}$ refers to the rotational spring; $\boldsymbol{K}_t \in \mathbb{R}^{3\times 3}$ to the translational spring and $\boldsymbol{K}_c \in \mathbb{R}^{3\times 3}$ to the coupled spring (causing both rotation and translation). Then it is shown in [9, p. 168] that the wrench exerted by the spring on the end effector is defined as,

$$\tilde{\boldsymbol{\tau}}^n = -2\,\mathrm{as}\left(\boldsymbol{G}_o \boldsymbol{R}_n^v\right) - \mathrm{as}\left(\boldsymbol{G}_n \boldsymbol{R}_v^n \boldsymbol{p}_n^{v,v} \tilde{\boldsymbol{p}}_n^{v,v} \boldsymbol{R}_n^v\right) - 2\,\mathrm{as}\left(\boldsymbol{G}_c \tilde{\boldsymbol{p}}_n^v \boldsymbol{R}_n^v\right) \tag{1.13}$$

$$\tilde{\boldsymbol{f}}^n = -\boldsymbol{R}_v^n\,\mathrm{as}\left(\boldsymbol{G}_n \tilde{\boldsymbol{p}}_n^v\right) \boldsymbol{R}_n^v - \mathrm{as}\left(\boldsymbol{G}_n \boldsymbol{R}_v^n \tilde{\boldsymbol{p}}_n^{v,v} \boldsymbol{R}_n^v\right) - 2\,\mathrm{as}\left(\boldsymbol{G}_c \boldsymbol{R}_n^v\right) \tag{1.14}$$

where the actual wrench from the spring exterted on the end-effector is the composition from the two components:

$$\boldsymbol{W}^n = \begin{bmatrix} \boldsymbol{\tau}^n & \boldsymbol{f}^n \end{bmatrix}^\top. \tag{1.15}$$

$\boldsymbol{G}_\gamma$ with $\gamma \in \{\mathrm{c}, \mathrm{o}, \mathrm{t}\}$ is the co-stiffness and can be calculated as,

$$\boldsymbol{G}_\gamma = \frac{1}{2}\,\mathrm{tr}(\boldsymbol{K}_\gamma)\boldsymbol{I}_3 - \boldsymbol{K}_\gamma \tag{1.16}$$

where $\boldsymbol{I_3}$ is the 3×3 identity matrix. Before the control law can be expressed as torques, the wrench in Eq. (1.15) has to be transformed to the inertial frame Ψ_0. As the wrench is a co-vector, it transforms with the transpose of the Adjoint. Therefore, the Wrench expressed in the inertial frame is defined as

$$(\boldsymbol{W}^0)^\top = \mathrm{Ad}^\top_{\boldsymbol{H}_0^n}(\boldsymbol{W}^n)^\top . \tag{1.17}$$

The final step is to express the resulting wrench in the joint space as the final control law are torques. To this extent, one can use the geometric Jacobian. The Jacobian maps between the end-effector space and the joint space. The torques can therefore be computed as

$$\boldsymbol{\tau}_c = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W}^0 . \tag{1.18}$$

Eq. (1.18) represents the torques caused by the 6-dimensional spring. Applying this control law to the equation of motion from Eq. (1.4) the following dynamics are obtained:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau}_e + \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W}^{\mathbf{0}} . \tag{1.19}$$

Often, gravity compensation is applied as a good estimation of the torques caused by gravity is known and to maintain the structure of a multi-link robot:

$$\boldsymbol{\tau}_c = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W}^0 + \hat{\boldsymbol{G}}(\boldsymbol{q}) . \tag{1.20}$$

Sometimes, even an estimate of the intrinsic dynamics is embedded in the control law in order to cancel them [5]. However, for low frequent trajectories and when the system is at rest the intrinsic dynamics cancel and only the gravity has to be compensated. Then, the system will behave like a spring which guarantees a stable interaction [10].
The choice of this Cartesian controller which is characterised by the multi-DOF spring can be extended with a damping term in the joint space as was done in Eq. (1.9). Damping can also be injected through a 6-dimensional Cartesian damper which is connected between the end-effector and the virtual frame (or an inertial frame) which will be explained in Section II in the paper (Chapter 3).

The desired impedance

As outlined above, the dynamics of the system can be changed to a mass-spring-damper system either in the joint space or in the work space. Essentially it becomes non-linear as the reflected inertia changes with the configuration. This type of system normally yields the 'desired impedance'. The values of the spring and damper either in the joint space or in the workspace then characterises the desired impedance. The spring and damper values do not need to be fixed. They can also vary, either in time or in space. In the sequel, a closer look is taken on how literature deals with the choice of the impedance parameters.

1.2.4 How is the impedance chosen?

In previous section it is shown how the basic impedance controller is implemented. In literature one will find a comprehensive list of applications in which impedance control is used. It ranges from applications with human-robot interaction, industrial applications such as peg-in-hole, aerial robotics and much more. The way impedance control is implemented is in most cases very similar to the procedures described before. A question one might ask is how the impedance parameters are chosen. 'Impedance parameters' refers to the $\boldsymbol{K}$ and $\boldsymbol{B}$ either in the joint space or in the workspace, representing stiffness and damping respectively. In the following, the procedure for selection of impedance parameters in literature is investigated. To this purpose, an overview of is given in Table Table 1.1. The given task is displayed together with how the impedance is chosen. 'VI' is short for variable impedance. In addition, it is indicated if the chosen impedance controller makes use of a variable impedance or a fixed one.

Source	Task	VI	How is the impedance chosen?	How does the VI vary?
[11]	Co-manipulation for peg-in-hole.	Yes.	Task is split up into sub-tasks: carrying and positioning. For each sub-task a different impedance is chosen.	In space
[12]	Trajectory following for peg-in-hole task.	No	N/A	N/A
	Trajectory following with obstacles.			
[13]	Piston insertion by teaching guidance.	Yes	High impedance used for rotational components, translational stiffness is set by human.	In space
	Wiping the table (constant force on table).		Task is split into a carrying phase and impedance phase. Overall compliant behaviour to be more safe in interaction with humans.	
	Opening a door.		Potential energy shaping with a minimum along a desired path. The rotational stiffness was set to zero	
[14]	Gear assembly	No	N/A	N/A
[15]	Trajectory following	Yes	Minimize metabolic cost and trajectory error.	In time
[16]	Telerobotic peg-in-hole	Yes	Manually	In space
	Telerobotic throwing switch			
[17]	Telerobotic force minimization	Yes	Manually	In space
	Telerobotic trajectory following			

[18]	Peg-in-hole assembly	No	Minimizing contact forces while inserting.	N/A
[19]	Aerial trajectory following and force application.	Yes	stiffness is bounded by Lyapunov stability criterion. The desired impedance is not treated.	In time
[20]	Trajectory following in stochastic force field.	Yes	Human-like reinforcement learning based on minimizing high impedance gains, trajectory error and accelerations.	Parametrized with a finite number of parameters. These parameters vary in time.
[21]	Human following trajectory	Yes	The impedance is adjusted to a variable impedance of the human. The variable impedance of the human is estimated using neural networks.	In space
[22]	Socket plugging	Yes	By human (using EMG signals)	In space
[23]	Pendulum balancing	Yes	Minimizes trajectory error while maintaining stability to adjust feed-forward and impedance.	In time

Table 1.1: Table gives an overview of a literature study to how the impedance is chosen on the basis of a task. VI means variable impedance. N/A means that it is not applicable or that it is not treated

What can be seen from table Table 1.1 is that a variety of tasks can be considered when talking about impedance control. Some of the applications, [11, 13, 15–17, 19–23], are using a variable impedance. On top of that, what is noticeable from the fourth column in Table 1.1 is that the choice of impedance whether it is variable or non-variable is different, it depends on the given task. Some designers, [12, 18], do not treat the choice of the impedance at all. In case a telerobotic application is considered like in [16, 17, 22], the impedance is chosen manually (by human). The human is connected to the 'master' device which is controls the 'slave' device. The 'slave' is the device which actually interacts with the environment and provides (haptic) feedback to the human. In this case, it is clear that the impedance is chosen by the human but it is not clear 'how' the human chooses their impedance.

Others, [11,13], use different impedances on different stages of the task, so called sub-tasks. Finally, some applications, [15, 20], try to mimic the human decision making on choosing the impedance. Their method is inspired by the CNS (central nervous system) of human which is able to make a trade-off between energetic cost and performance.

In Table 1.1, variable impedance refers to the change in impedance parameters either in space or in time. If the impedance varies only in time, it means that the stiffness is controlled 'blindly' without considering the configuration of the robot. This is similar to open-loop control. If the impedance is parametrized in space instead of time, then the impedance will depend on the configuration of the robot. This requires a feedback loop hence, closed-loop control. Recall that in the telerobotic cases, [16,17,22], the impedance is chosen manually (by human). In this case, it is assumed that the human chooses their impedance based on (haptic) feedback and therefore the impedance changes in 'space'. However, the latter statement can be bit strong as there are many more variables like confidence and fatigue.

Human-like analogy

When discussing impedance modulation many authors make the analogy with the human body [15, 20]. This is for a good reason; humans change their impedance all the time. Normally this happens in the joints and is achieved by co-contraction of the muscles [24]. Humans benefit from a modulated impedance for non-trivial task execution such as picking up objects or insertion tasks (like putting a key in a key-hole). According to [24] humans are capable of doing this as they have trained themselves thousands of times. For example, picking up an object such as a mug or inserting a key is an every-day task. However, from a robotics point of view, this is highly non-trivial due to the complex dynamics. As reported by [15] such actions are performed by an open-loop controller as the task is 'learned' in the past. Then, a feedback mechanism exists to account for unforeseen events like when the object picked up is heavier than expected. However, in terms of speed and stability, the open-loop action is beneficial over the feedback controller. A high-gain feedback controller is less safe and more prone to stability issues. This is because the feedback sensing faces delay. Following their exceptional performance, it is clear that humans are a perfect example of varying their impedance on the basis of a task. Therefore when designing a (varying) impedance controller, it might be beneficial to try to mimic the human decision making. Controller design inspired by nature in which only some aspects are tried to be copied is also called bioinspired controller design.

1.2.5 Stability and passivity

For isolated robots which do not interact with their environment, stability can be proven using classical techniques such as the small gain theorem, Hurwitz or Bode stability [4]. However, for robots which interact with their environment these classical stability criteria are not be sufficient. As described by [25], passivity is a sufficient condition for stability, not a necessary one. Hence passivity guarantees stability, but stability does not require passivity. Moreover, it was shown in [25] that a non-passive system can become unstable even when interacting with passive environments. It emphasized that passivity therefore says nothing about the performance of a system, only about stability. Passivity basically means that a system cannot store more energy than what is provided to it and was initially present. In mathematical terms, a system is said to be passive if there is a storage function $\mathcal{V} : \mathcal{X} \mapsto \mathbb{R}^{+}$, where $\mathcal{X}$ is the state manifold such that [26],

$$\int_{t_0}^{t_1} \dot{\mathcal{V}}\, dt = \mathcal{V}(x_1) - \mathcal{V}(x_0) \leq \int_{t_0}^{t_1} y^{\top} u \, dt \tag{1.21}$$

with state x, input u, output y and $t \in [t_0, t_1]$. As described before, conventional robot control is not sufficient when a robot is interacting with the environment. In the sequel, a simple example of a mass subjected to friction and controlled with a traditional PI controller is given. It is shown that because the controller contains an integrator and because interaction is assumed to be a 'disturbance' the system can become unstable while interacting.

Example: PI control

This example is taken from [5]. Consider a simple mass subjected to (viscous) friction with positive friction coefficient b like in Fig. 2.1. The transfer function from force to position can be written as,

$$H_{X/F} = \frac{1}{ms^2 + bs} \tag{1.22}$$

where x is the position of the position of the mass and F are the actuator forces plus the external forces, $F = F_a + F_e$. A traditional PI controller is applied,

$$H_{F/\tilde{X}} = \frac{K_p s + K_i}{s} \tag{1.23}$$

where $\tilde{x}$ is the desired position minus the real position $\tilde{x} = x^{*} - x$. The Routh-Hurwitz stability creation can be applied to choose an appropriate K_i value. The characteristic closed loop polynomial of this system will be

$$\chi_{cl}(s) = ms^3 + bs^2 + K_p s + K_i \,. \tag{1.24}$$

From this, the Routh table is constructed:

$$\left| \begin{array}{c|c} m & K_p \\ b & K_i \\ K_p - \frac{m}{b} K_i & \end{array} \right| .$$

The feedback system is stable if the entries of the first column have the same sign and are nonzero. It is easy to see that this is true if and only if

$$K_i < \frac{bK_p}{m} \,. \tag{1.25}$$

Now consider that the mass is interacting with another mass. This would be similar to a robot picking up an object. The the 'new' mass of the system will not equal m be something bigger. This can cause the stability condition in Eq. (1.25) to fail hence, instability of the total system. Robust control methods will suggest to work with a safety factor for K_i to allow for some variations in m. However, the system can always become unstable when the mass becomes sufficiently high [5].

In earlier section, it was shown that some of the impedance controlled applications in literature make use of a variable impedance. Regarding passivity, when a time-varying impedance is chosen, the system is not passive by itself. This is shown in the next example for a simple 1D-mass. The example is meant to give an impression on what a variable impedance implies, it is not a generalized proof on passivity with different types of impedances. The same example also showcases how the passivity is destroyed if a moving reference is used.

Example: Passivity analysis with stiffness modulation and moving reference

The equation of motion for a controlled mass is given as,

$$m\ddot{x} = F_a + F_{ext} \tag{1.26}$$

where F_a is the controlled actuator force and F_{ext} the external forces. Choosing an impedance law in which the impedance parameters k and b can be time-varying functions, the equation of motion results in a mass spring damper system,

$$m\ddot{x} - k(t)\tilde{x} + b(t)\dot{x} = F_{ext} \tag{1.27}$$

with $\tilde{x} = x^* - x$. Where x^* is the desired state. The storage function can be defined as

$$\mathcal{V} = \frac{1}{2}m\dot{x}^2 + \frac{1}{2}k(t)\tilde{x}^2 \,. \tag{1.28}$$

To see how the energy changes over time, the time derivative is taken:

$$\dot{\mathcal{V}} = m\dot{x}\ddot{x} + k(t)\tilde{x}\dot{\tilde{x}} + \frac{1}{2}\dot{k}(t)\tilde{x}^2 \,. \tag{1.29}$$

From Eq. (1.27) we isolate $\ddot{x}$ and insert it in $\dot{\mathcal{V}}$:

$$\dot{\mathcal{V}} = \dot{x}F_{ext} + k(t)\tilde{x}\dot{x} - b(t)\dot{x}^2 + k(t)\tilde{x}\dot{\tilde{x}} + \frac{1}{2}\dot{k}(t)\tilde{x}^2 \,. \tag{1.30}$$

Assuming that the reference does not change in time (called regulation) : $\dot{x}^* = 0$ and hence $\dot{\tilde{x}} = -\dot{x}$ provides

$$\dot{\mathcal{V}} = \dot{x}F_{ext} - b(t)\dot{x}^2 + \frac{1}{2}\dot{k}(t)\tilde{x}^2 \,. \tag{1.31}$$

For a fixed impedance $\dot{k}(t) = 0$. Furthermore, as b will always be greater than zero and hence $b\dot{x}^2 > 0$ the following holds:

$$\dot{\mathcal{V}} = \dot{x}F_{ext} - b\dot{x}^2 \leq \dot{x}F_{ext} \,. \tag{1.32}$$

Integrating both sides:

$$\int_0^T \dot{\mathcal{V}} dt = \mathcal{V}(x_T) - \mathcal{V}(x_0) \leq \int_0^T \dot{x}F_{ext} dt \,. \tag{1.33}$$

This holds the passivity criterion in (1.21), hence the system fixed impedance is passive. For a variable impedance, obviously $\dot{k} \neq 0$ and therefore $\dot{\mathcal{V}}$ can also be written as

$$\dot{\mathcal{V}} = \dot{x}F_{ext} + \left[\frac{1}{2}\dot{k}(t)\tilde{x}^2 - b(t)\dot{x}^2\right] \,. \tag{1.34}$$

The sign between the square brackets is not defined, it can be both positive and negative, hence the system is not passive any more. The varying stiffness produces another power port that can destroy the passivity [27]. In a final situation we can analyse the case when the desired state varies in time while using a fixed impedance: $\dot{x}^* \neq 0$, $\dot{k}(t) = 0$. Starting from Eq. (1.30), we obtain

$$\dot{\mathcal{V}} = \dot{x}F_{ext} + [k(t)\dot{x}^*] - b(t)\dot{x}^2 \,. \tag{1.35}$$

Again, the sign between square brackets is not defined. In this case it is the the varying reference which creates another power port. Again, it may result in failure of the passivity condition.

The problem of passivity has been solved in an elegant and simple way using a so called energy tank. The main idea is that the energy which the controller can use is bounded by means of a limited supply contained in a tank. If the energy is tank is empty, no more energy can be spent by the controller. This guarantees passivity and has been shown for a 1-DoF case in [28] and the extension to the multi-DoF has been made in [7]. The benefit of using the energy-tank concept is that it is easy to implement and it allows to think in a 'free' way on how to tune the impedance parameters without worrying about passivity. The energy tank will be used and explained further in the next chapter.

1.3 Problem description

1.3.1 General problem

Impedance control has been studied in detail and it is clear that different tasks require different impedance values (refer to Section 1.2.4). For some non-trivial tasks even a variable impedance is beneficial. A question which remains unanswered is how to structurally choose the impedance profile on the basis of a task. In other words, there exists no unified task-based framework to define impedance. This issue was already noticed by [29]:

"To what degree can performance metrics in interactive systems be cast into a unified framework?"

Moreover, even the choice and technique used to define the variable impedance differs across literature. It seems (refer to Table 1.1) that most recent publications working with variable impedance are data-driven. That means that the varying impedance parameters are determined with the use of data. Often, those data-driven techniques assume the dynamics of the system to be a 'black-box' and learn impedance profiles directly from sensor feedback. These techniques have the benefit that they can cope with advanced and unstructured environments (such as a rover on Mars). They learn from the feedback and how to adjust the impedance according to the task. However, many impedance controlled robots operate in more structured environments (such as industrial environments), where the characteristics of the environment are known. Therefore, the core idea put forward in this thesis is,

The knowledge about the structure of the environment, robot dynamics and task definition can be utilised to define 'good' initial values for (time-varying) impedance that allow completion of the task in nominal conditions.

1.3.2 Project scope

The scope of this thesis is to see how the impedance parameters should be changed according to one task only. The task in this thesis will be a peg-in-hole task as it is considered to be a benchmark for manipulative robotics. We examine a framework in which the task is optimized in nominal conditions to perform successful task execution and to account for model variations with the aid of feedback. Energy tanks will be used to achieve passivity of the controlled system. However, as highly debated aspect of passivity in robotics, the energy tanks do not guarantee safety as the amount of energy stored in the tanks can be very large, even if theoretical passivity properties are satisfied. That is why we will also look at how to use the energy tanks for both passivity and safety. In line with the project scope, the next research questions are formulated:

1. What is the desired time-varying impedance for a peg-in-hole task in a (semi)-structured environment based on the task definition and a task-based metric?

2. How does the desired time-varying impedance relates to a control architecture comprising feed-forward and feedback?

3. What can you say about the robustness of the optimized impedance profiles under model variations?
4. How to utilize the energy tanks in view of passivity and safety?
5. Can the results from the optimization somehow be improved with the aid of data?
6. How do the fixed impedance, task-driven impedance and data-driven impedance compare?

1.4 Contribution

This work contains three contributions:

1. The main contribution is the presentation of a control strategy comprising a task-based control action which exploits time-varying Cartesian impedance control in an optimization. The task-based control action is supplemented by a task-free impedance controller to handle environmental uncertainties and external perturbations. A major difference with existing literature is that the time-varying impedance results in an open-loop control action rather than a closed-loop strategy. In terms of speed and stability, the open-loop action is beneficial over a feedback controller. A high-gain feedback controller is less safe and more prone to stability issues due to delays in the feedback mechanism.
2. A second contribution relates to safety using the energy tanks. As being said in Section 1.3.2, energy tanks only guarantee passivity of the controlled system, not safety. Making the tanks more task-based by only storing the necessary amount of energy to complete a task will increase safety. This contribution is in line with the conclusion from [7]:

   "For this reason, in future work, it will be investigated how to make an energy-tank system which is aware of the energy amount needed to perform a specific task, resulting in a more efficient energy-aware controller."

   In addition, we exhibit a dynamic way of injecting energy into the energy tanks. The dynamic energy injection is task-based and results in lower energy content in the tanks compared to conventional input strategies. This increases safety of the controlled system.
3. A final contribution is associated with an iterative feed-forward adaptation scheme. We will present a simple adaptation law to adapt the task-based control action to such a degree in which it accounts for model variations from the nominal plant. The final result is successful task execution using an energy-aware open-loop control action which accounts for variations with minimal feedback.

1.5 Approach

Control architecture: task-based versus task-free

We shall exploit impedance control ideas and make them optimal in time resulting in an open-loop control action which mimics a time-varying Cartesian impedance controller on the basis of a task performed on a nominal robot model. The final product should be a manipulator performing a non-trivial task such as peg-in-hole. The hypothesis is that the time-varying impedance will be beneficial over a fixed impedance on the basis of a task-based metric. The resulting task-based control action is supplemented by a task-free Cartesian impedance controller to handle environmental uncertainties and external perturbations.

Optimization

In order to find the task-based control action we try to come up with a metric/objective on the basis of a task and to formulate a non-linear problem (NLP) which minimizes this objective. The objective will always be chosen by the designer and not considered to be the only option. However, following the exceptional performance of humans, (explained in Section 1.2.4) we approach the objective definition from a biomimetic perspective. Since the optimization problem is of continuous-time nature (i.e. would require gradient descent in a functional space to be solved), we seek the solution in a finite dimensional space. We will do this by parametrizing the stiffness with a finite set of parameters while maintaining the continuous nature of the problem as a whole.

Energy tanks

Energy tanks will be used to recover passivity of the controlled system. The energy-tank concept allows to freely think about varying the impedance without having to worry about passivity or stability. However, the energy-tanks will not guarantee safety as the amount of energy to be stored in the tanks can be very large. Therefore, we will also use the energy-tanks to recover safety. We shall do this by using the knowledge about the task-based optimization since it will provide information on the energy usage of the robot as a function of time in nominal conditions. The latter can be used to store the tanks only with a limited amount of task-based energy. In addition, we use the knowledge from the optimization to exploit a dynamic way of injecting the task-based energy into the tanks. This will make the tank based approach more *energy-aware* and safe.

Model variations

After optimizing the impedance in nominal conditions, also disturbances enter the problem. There will always be a mismatch between the nominal model used in the optimization and practical implementation. As the task-based time-varying impedance will enter as an open-loop control action, the task-free closed-loop controller will account for these modelling errors, as well as external perturbations. We shall study the trade-off between the necessary gains to perform a successful task execution with model variations and energetic cost given the bounds of uncertainty. This information will also be used to store an additional amount (on top of the task-based energy) of energy in the tanks to account for model variations.

Iterative feed-forward adaptation

The tasks considered in this work will be appropriate for an industrial setting. In an industrial environment, task are often carried out repeatedly. Therefore, the interplay between the task-free and task-based controller can be utilized to adapt the task-based control action in such away that it accounts for model variations. The result of this data-driven adaptation law results is that the unstructured environment also is encapsulated into the task-based feed-forward. Subsequently, the amount of energy in the tanks can be updated corresponding to the knowledge of the controlled action.

Procedure and validation

The methods will be validated though simulation studies. It also allows to be more 'flexible' with the type of application without being constrained to a real-life environment. First, a proof of concept will be conducted in which a rather simple system is used to get the main ideas for this project. Herein, a mass subjected to nonlinear friction has to perform a positioning task. After the proof of concept, the ideas are studied on a multi DoF manipulator performing a peg-in-hole task. Cartesian impedance control as described in section Section 1.2.3 will be used to define the impedance control law. Bond graphs will be used as a modelling language as it fits well with the Cartesian impedance controller and screw theory. Furthermore, these graphs are very suitable for energy-based modelling and control which are concepts used a lot in this thesis. Regarding software: 20Sim 4.8 will be used to make the bond graph models. MATLAB 2020A together with Simulink will be used to perform the optimization problems and do the actual control.

1.6 Structure of thesis

The structure of this thesis will be as follows. Chapter 1 is the introducing chapter. It introduces impedance control and how it is established in existing literature. Furthermore, it contains a reflection on literature and which questions arise due to this reflection. From this, global problem formulation is presented together with the scope for this project. Chapter 2 will present a proof of concept and will be a more technical chapter. Herein, the main ideas on how to tackle the questions posed in Chapter 1 are exploited on a simple system performing a task. Chapter 3 builds upon the work in Chapter 2 and extends to Cartesian tasks on a multi-DoF manipulator. This chapter will also contain most of the contribution. It is written in IEEE paper format as it is meant to be published on the Humanoids conference in Munich 2021. Chapter 4 will be a final chapter. It reflects on the overall work and makes suggestions for future research. The appendices provide the reader with the necessary background in bond graphs, screw theory, modelling and mathematics. However, these appendices will be rather minimal in their explanation. It serves in order to remind the reader on the essentials. Therefore, it is advised to study more elaborated literature first if one starts completely blank.

Chapter 2

Proof of concept

2.1 Problem formulation

In this chapter, a proof of concept will be presented. This proof of concept has 1-DoF such that complexity in terms a Jacobian and multi-body dynamics can be avoided. It is expected that the main features of this thesis can be exploited in 1D as well. The goals of this proof of concept are to see:

1. Can a simple system performing a task benefit from a variable impedance?
2. What is the task and what is the subsequent metric?
3. What is the control scheme regarding feed-forward and feedback?
4. Technically: how do you implement such a problem?
5. How to ensure passivity of the controlled system using the energy tank concept?

The system considered here is a (rigid) mass subjected to a nonlinear friction model, displayed in Figure 2.1.

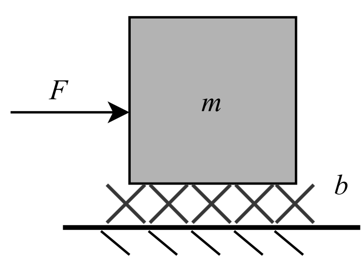


Figure 2.1: Mass m subjected to friction b with external force F

The continuous time dynamics of the mass are written as,

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} F \tag{2.1}$$

where x_1 is the position and x_2 the velocity of the mass. F is the total force exerted on the mass consisting of controller force F_c and friction force F_r:

$$F = F_c + F_r \,. \tag{2.2}$$

An impedance controller characterized by a spring with stiffness k and damper with damping $\bar{b}$ will be used to control the system:

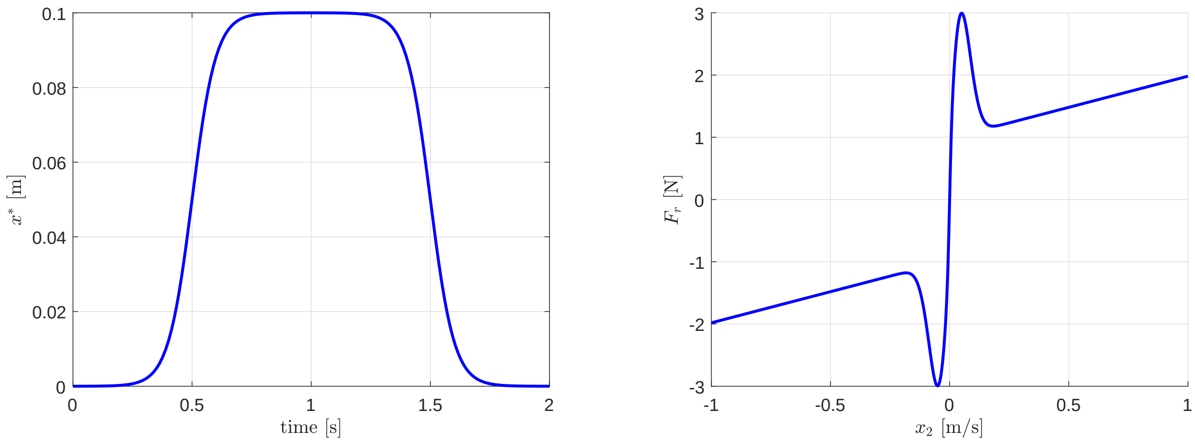
$$F_c = k\tilde{x} - \bar{b}x_2 \quad \bar{b} = 2\sqrt{km} \tag{2.3}$$

where $\tilde{x} = x^* - x_1$ defines the position error with desired trajectory x^*. The damping parameter, $\bar{b}$, is chosen such that the system will be critically damped. This is a common choice for positioning tasks. The characteristic of the nonlinear friction profile are given in Fig. 2.2b. It consists of a Stribeck friction profile together with some coulomb friction [30]. The friction force is therefore a nonlinear function b of the velocity:

$$F_r = b(x_2) \,. \tag{2.4}$$

2.1.1 Task and metric

The task is to follow a desired trajectory profile as in Fig. 2.2a. The performance objective is therefore a tracking error. On top of the performance objective the mass should be aware of how much energy it has spent to perform the task. The hypothesis is that the mass can take the friction profile to its advantage. When the mass is accelerating, the friction profile will act in opposite direction of the actuator force. However, when the mass wants to de-accelerate the friction will contribute to the task. In that situation, the energy spent by the controller might be less when de-accelerating compared to the accelerating phase. Therefore the metric chosen in this case is to achieve a certain tracking error while minimizing the metabolic cost. The metabolic cost can be defined as the energy spent to perform the task. This choice of metric is human-like as already described in Section 1.2.4 . It says that the human central nervous system (CNS) tries to minimize the energy spent on a task while taking into account the performance. This biomimetic choice of metric also fits well within the bond graph paradigm and energy tank concept as it gives immediate information on how much energy should be stored in the tanks, more on this in Section 2.4. As a final note, this choice of task and metric is reasoned but certainly not the only option. Another option could be: given a fixed amount of energy, optimize the tracking error.



(a) Desired position profile for one period.

(b) Nonlinear friction profile as a function of the velocity.

Figure 2.2: Positioning task and friction characteristics.

2.2 Optimization

After the task, robot and environment dynamics are defined, it is time to define an optimal impedance strategy on the basis of the task.

Stiffness parametrization

Regarding the practical implementation, a smooth stiffness curve in time is preferred. However, optimizing a function in continuous time would require gradient descent in a functional space. For a practical implementation, we seek the solution in a finite dimensional space defined by N parameters using cubic splines. We write the set of N parameters (also called 'knots' of the stiffness curve) as

$$\boldsymbol{k}_{knots} = \begin{bmatrix} k^1 & k^2 & \dots & k^N \end{bmatrix} \tag{2.5}$$

with equal time spacing between adjacent knots. Between the knots, a cubic interpolation will be done also a known as cubic splines [31]. In this way, the stiffness curve becomes a function of the knots:

$$k = f(\boldsymbol{k}_{knots}, t) \quad \dot{k}(k_{knots}, 0) = \dot{k}(k_{knots}, T) \tag{2.6}$$

where $t \in [0, T]$ with period time T. We choose to equate the derivative of the stiffness curve at $t = 0$ and $t = T$ to obtain a smooth stiffness curves when multiple periods are relevant (more on this in Section 2.2.1).

From metric to objective

In the following, it is explained how the biomimetic metric translates to a mathematical objective. It is considered that the mass is driven by an electrical motor. To simplify, it is assumed that the power flow to the self-inductance is negligible so that the motor only consists of a electrical resistor with resistance R and a gyrator with torque constant k_t. The metabolic cost is considered to be the power loss in the electrical resistance plus the power spent to the mechanical domain. k_t and R are assumed to be constants so that the power lost in the electrical resistance is proportional to F_c:

$$P_r = i^2 R = \frac{F_c^2}{k_t^2} R \propto F_c^2 \tag{2.7}$$

where i is the current flow in the motor. The power spent to the mechanical domain is simply

$$P_m = F_c x_2 \tag{2.8}$$

The motor does not contain physical storage elements nor is the motor driver assumed to be a four-quadrant capable driver. Therefore in order to express the metabolic cost, only the positive powers are assumed as this is the power spent to to accomplish the task:

$$E_{metabolic} = \int_0^T (P_r + P_m)^+ dt \tag{2.9}$$

where $(\cdot)^+$ denotes only the positive part of the input argument. This translates to a mathematical objective which is the weighted metabolic cost:

$$J_{metabolic} = \int_0^T (\mu_1 P_r + \mu_2 P_m)^+ dt \, . \tag{2.10}$$

where $\mu_1, \mu_2 \in \mathbb{R}$ are weighting constants used to scale the individual power terms. Given the task, the performance objective is to minimize a certain tracking error ε. The corresponding performance objective is specified as the root mean square of the tracking error over the entire interval $t \in [0, T]$:

$$J_{erms} = rms(\tilde{x}) \, . \tag{2.11}$$

In addition, another performance objective is to minimize the maximum (absolute) position error, e_{max}. A maximum tracking error upto ε is allowed but will be penalized if it surpasses ε. This translates to the following objective and is even graphically displayed in Fig. 2.3:

$$\begin{cases} J_{emax} = (e_{max} - \varepsilon)^2 & e_{max} \geq \varepsilon \\ J_{emax} = 0 & e_{max} < \varepsilon \end{cases} \tag{2.12}$$

where

$$e_{max} = |\tilde{x}|_{max} \, . \tag{2.13}$$

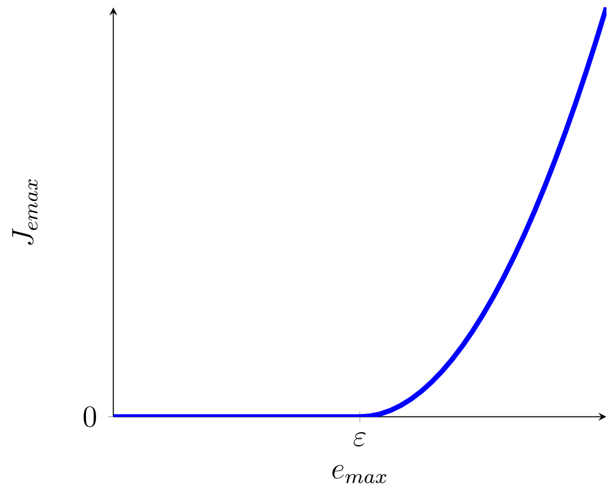


Figure 2.3: J_{emax}

The total cost function is then defined as,

$$J(\boldsymbol{k}_{knots}) = w_1 \cdot J_{metabolic} + \underbrace{w_2 \cdot J_{erms} + w_3 \cdot J_{emax}}_{\text{performance}} . \tag{2.14}$$

where $w_1, w_2, w_3 \in \mathbb{R}$ are the weighting constants used to scale the objective terms. Subsequently, the optimization problem can be formulated as

$$\boxed{\begin{aligned} \boldsymbol{k}_{knots} = \underset{\boldsymbol{k}_{knots}}{\text{argmin}}\, J(\boldsymbol{k}_{knots}) \\ \text{s.t,} \\ k^+ \geq k(\boldsymbol{k}_{knots}, t) \geq k^- \quad \forall t \\ f(\boldsymbol{k}_{knots}, 0) = f(\boldsymbol{k}_{knots}, T) \end{aligned}} . \tag{2.15}$$

The latter constraint together with the equality of the derivatives in Eq. (2.6) makes a repeating k curve continuous if a repetitive task is desired. k^- and k^+ denote the lower and upper stiffness bound.

Algorithm

The optimization problem in posed in Eq. (2.15) is solved using MATLAB 2020a. `fmincon` with the `'interior-point'` algorithm will be used as the problem is posed as a constrained and non-linear minimization problem. The dynamics (given by Eq. (2.1)) are solved using the `ode45` solver. In every objective evaluation the dynamics are simulated which ensures that the dynamics and minimization problem are solved sequentially. The main advantage of this implementation is that the problem enters as a continuous time problem and time discretization is done by the `ode45` solver itself. To clarify, the pseudo code has been outlined in Algorithm 1.

Algorithm 1: Pseudo code implementation for problem in equation 2.15

Result: $\boldsymbol{k}_{knots}$

initialization;

$k = spline(\boldsymbol{k}_{knots}, t)$;

$\boldsymbol{k}_{knots} =$ `fmincon(J(`k`),nlc(`k`),'interior-point')` ;

Function `J(`k`)`**:**

  $F(k, \boldsymbol{x}) = F_c(k, \boldsymbol{x}) - F_r(\boldsymbol{x})$;

  $[\boldsymbol{T}_{sol}, \boldsymbol{X}_{sol}] =$ `ode45(`$\dot{\boldsymbol{x}}(\boldsymbol{x}, F, t)$`)`;

  $J(k, \boldsymbol{T}_{sol}, \boldsymbol{X}_{sol}) = w_1 J_{metabolic} + w_2 J_{erms} + w_3 J_{emax}$;

  return J;

;

Function `nlc(`k`)`**:**

  $nlc(1) = k^+ \geq k \geq k^-$;

  $nlc(2) = k(0) == k(T)$;

  return nlc;

;

2.2.1 Optimization results

The parameters which are used to solve this problem can be found in the following table,

m	μ_1	μ_2	w_1	w_2	w_3	ε	N	k^+	k^-	$\boldsymbol{k}_0$	R	k_t	T
[kg]	$[-]$	$[-]$	[1/J]	[1/m]	[1/m]	[m]	$[-]$	[N/m]	[N/m]	[N/m]	[Ω]	[Nm/A]	[s]
1	1	1	3	10^3	$2 \cdot 10^4$	$12 \cdot 10^{-4}$	14	$2 \cdot 10^3$	10^3	10^3	1	1	2

where $\boldsymbol{k}_0$ is the initial guess for the spline knots in the optimization. The desired position profile, $x^*(t)$ contains frequency components up to 5Hz. Therefore, the bounds for the spline stiffness curve are chosen such that the bandwidth of the system is always bigger than 5Hz. If the bandwidth is lower than the frequency of the desired position profile, proper tracking cannot be achieved.

The results from the optimization for two periods can be seen in Fig. 2.4. This displays all forces on the mass, the resulting stiffness curve and the state evolution. In the accelerating phase (upto 0.5s), a larger impedance is used compared to the de-accelerating phase (from 0.5s to 1s). This is expected as the the friction profile will help the mass to de-accelerate. Therefore, in order to minimize the metabolic cost the controller makes use of a lower impedance when de-accelerating is desired. The continuity constraint for multiple periods allows to 'copy' the stiffness profile and obtain a smooth stiffness profile for multiple cycles (in this case two). Obviously, the advantage of minimizing the metabolic cost becomes a benefit if multiple cycles are considered as the energy 'gain' will scale linearly per cycle.

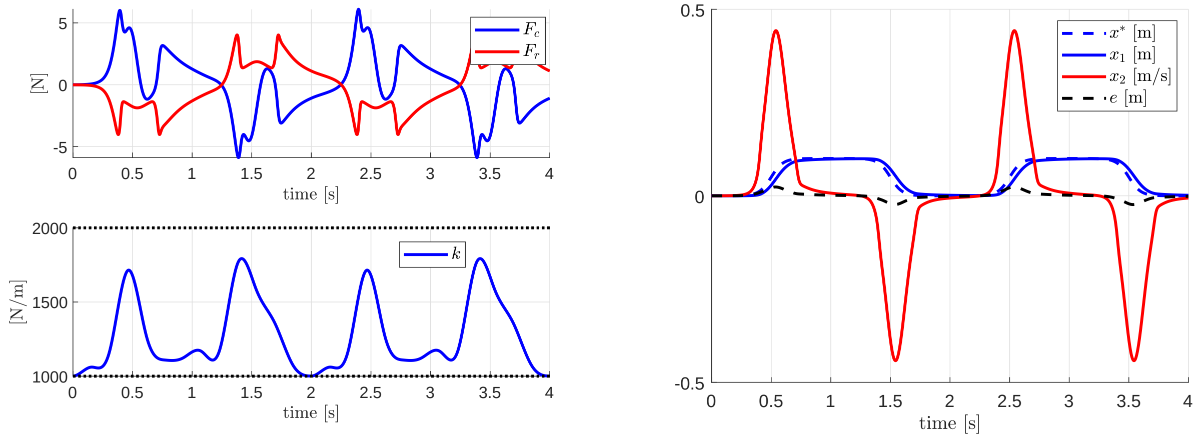


Figure 2.4: Results from optimization for 2 periods. Left plot up represents the control force together with the friction force. Left plot down represents the resulting spline curve for the stiffness. The right plot displays the state evolution.

In order to quantitatively compare the variable impedance result to a fixed impedance, ten experiments are run with fixed impedances. The range of fixed impedances is the same as the bounds given in the optimization for $\boldsymbol{k}_{knots}$. In Fig. 2.5 the variable and the fixed impedance are compared on the basis of the chosen metric.

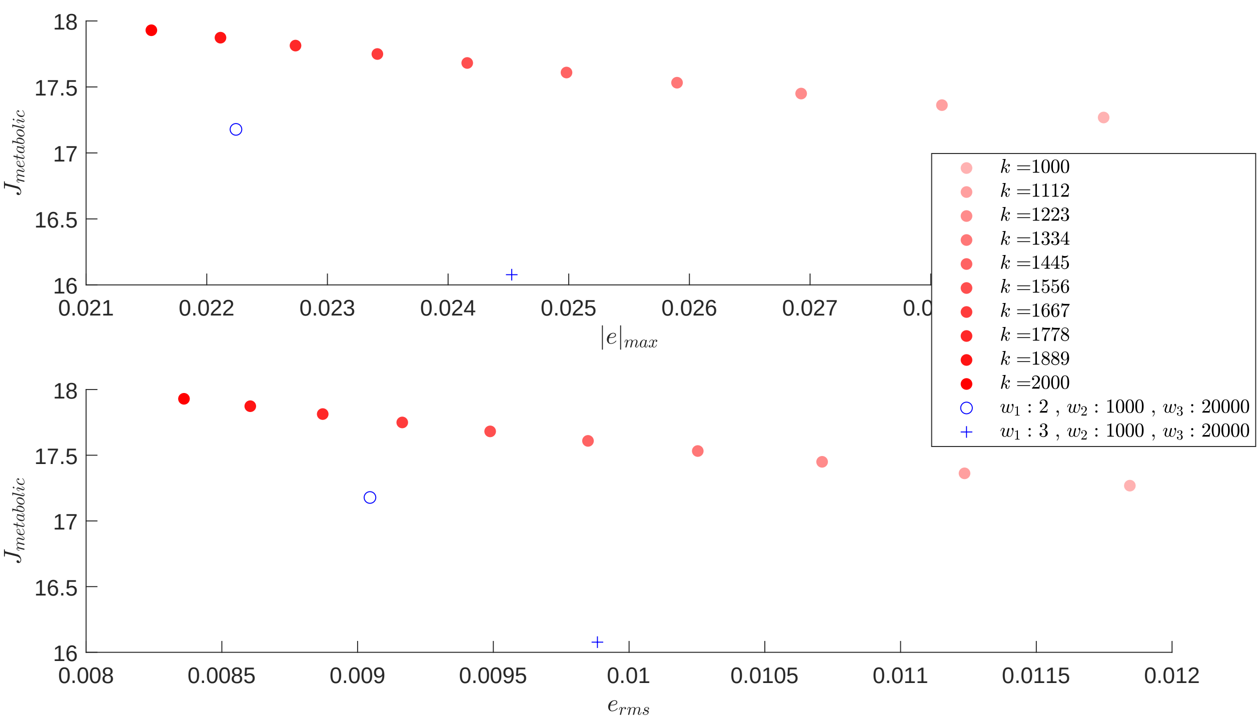
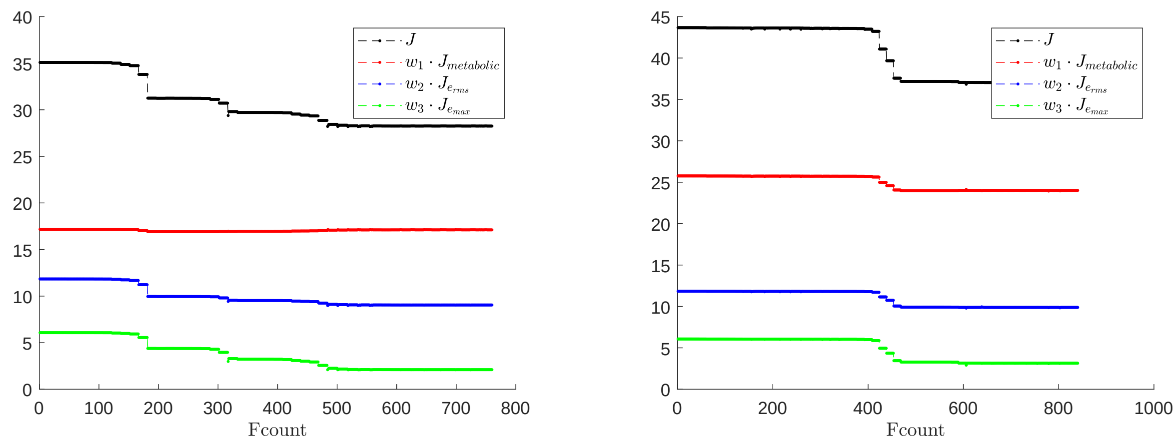


Figure 2.5: Red filled dots represent the results for the fixed impedances with stiffness value as is indicated in the legend. The variable impedance results are indicated with the blue markers. The legend for the variable impedance indicates the values for w_1,w_2 and w_3 respectively.

As can be seen from Fig. 2.5, when comparing the performance objectives versus the metabolic cost, the variable impedance is beneficial over the fixed impedances. The result and shape from the upper plot and the lower plot in Figure 2.5 are similar. The solver is able to make a trade-off between metabolic cost and positioning error (both maximum and root mean square). To also indicate the importance of weighting, Fig. 2.5 also displays two different results for the optimization based on two different sets of weights (w_1,w_2,w_3). As can be seen, increasing w_1 puts more weight of the metabolic cost within the overall objective (see also Eq. (2.14)). This leads to a trade-off more directed towards minimizing the metabolic cost. This however, is at the expense of the performance. Fig. 2.6 displays the objective evaluations for every function evaluation from `fmincon`. This is done for both set of weights which corresponds to Fig. 2.5. Clearly, how the problem is minimized depends on the weighting. It should be emphasized that by putting more weight on the performance objectives, the variable impedance result converges to a fixed impedance. This is also observed from Fig. 2.5 where the variable impedance result with the lowest weight on the metabolic cost (thus more weight on the performance objectives) term is the closest to the fixed impedance results.



(a) $w_1 : 2$, $w_2 : 1000$, $w_3 : 20000$

(b) $w_1 : 3$, $w_2 : 1000$, $w_3 : 20000$

Figure 2.6: Results from optimization for two sets of weights. Left plot corresponds to the 'circle' in Fig. 2.5 whereas the right plot corresponds to the 'plus' in Fig. 2.5

2.3 Control strategy

The results of this concept model illustrate that a simple system can benefit from a variable impedance. Furthermore, the resulting stiffness curve is only task-based, not data-driven. The error term $\tilde{x}$ in Eq. (2.3) is part of the optimization. Therefore, 'real' data is not considered in order to learn the desired impedance. The resulting stiffness curve is a function of time only. In order words, the stiffness is blindly modulated independent on the configuration of the robot. Because of that, it makes sense to let the resulting force from the optimization enter as an open-loop action to the robot. In the sequel, this open-loop part is labelled as 'task-based' control action. If the model of the robot and environment correspond to the model used in the optimization, the task-based open-loop control action will mimic as if a time-varying impedance was implemented. The state variables are (implicitly) part of the optimization, they define the optimal trajectories in nominal conditions, denoted by x_1^* and x_2^*. Everything else, which is unforeseen by the task formulation should be covered by a so-called 'task-free' controller. This 'task-free' controller is a feedback controller and can be implemented as a simple PD. The control action which enters the robot is a combination from a task-based (TB) part and task-free (TF) part:

$$F_c = F_{TB} + F_{TF} = F_{TB} + k_0 e + b_0 \dot{e} \tag{2.16}$$

where $e = (x^* - x)$ and $\dot{e} = (\dot{x}^* - \dot{x})$. k_0 and b_0 are the task-free gains and can be time-invariant. The control scheme comprising the task-based optimization and task-free compensation is given in Fig. 2.7.

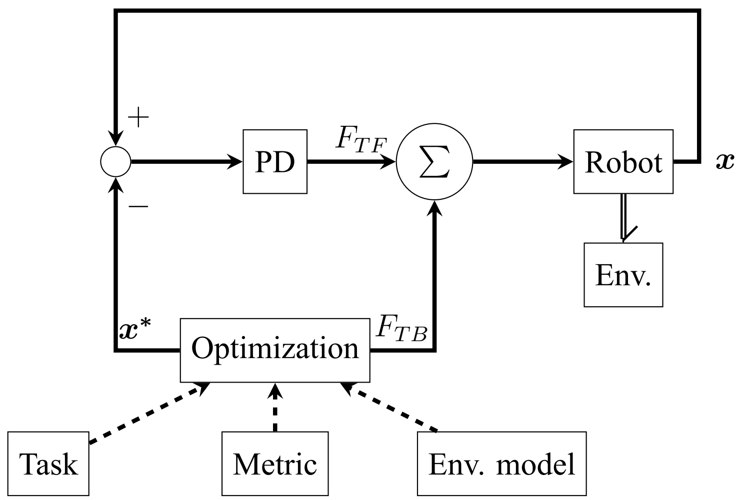


Figure 2.7: 'Env.' is short for 'environment'. Control scheme with task-based and task-free controller. The optimization receives a task, metric and characteristics of the environment in order to do an offline optimization. The dashed lines represent offline signals.

2.4 Passivity by energy tank

As indicated in Section 1.2.5, passivity can be destroyed by the creation of additional power ports due to a varying reference or a time-varying impedance. To ensure passivity the energy tank concept is used. The key idea is that the energy which the controller can use is bounded by means of an energy tank. The motor draws the energy from the tank by means of a modulated transformer. If the energy in the tank is empty, no more energy can be spent by the controller. This guarantees passivity and has been shown for a 1-DoF case in [28] and the extension to the multi-DoF has been made in [7]. Since for this model an electrical motor is considered, the current is set instead of the force. Unlike [7] the tank will be represented by an electrical inductance with $L = 1$H. Subsequently, the energy stored in the tank is given as $H(p_t) = \frac{1}{2}p_t^2$ where p_t is the state of the tank. The bond graph of the mass subjected to friction actuated with an electrical motor, connected to the tank will be represented in Fig. 2.8.

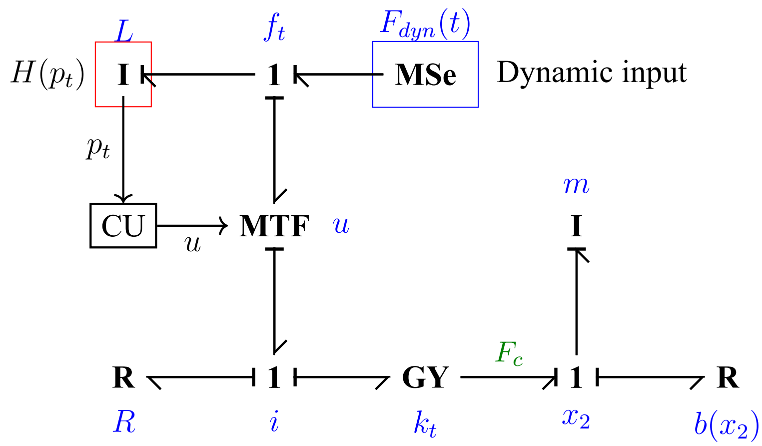


Figure 2.8: Bond graph representation of equation 2.1 driven by a motor connected to the energy tank. f_t the current through the tank.

Next, the rate of transformation between the tank and the actual system is determined in order to ensure a certain F_c going into the mechanical part. The port-Hamiltonian equations for system has to be derived. The system got two storage elements with integral causality which makes the total system second order. The derivation of these equations can be in appendix D.1 resulting in:

$$\begin{bmatrix} \dot{p}_t \\ F_c \end{bmatrix} = \begin{bmatrix} -u^2 R & -uk_t \\ uk_t & 0 \end{bmatrix} \begin{bmatrix} p_t \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} F_{in} \,. \tag{2.17}$$

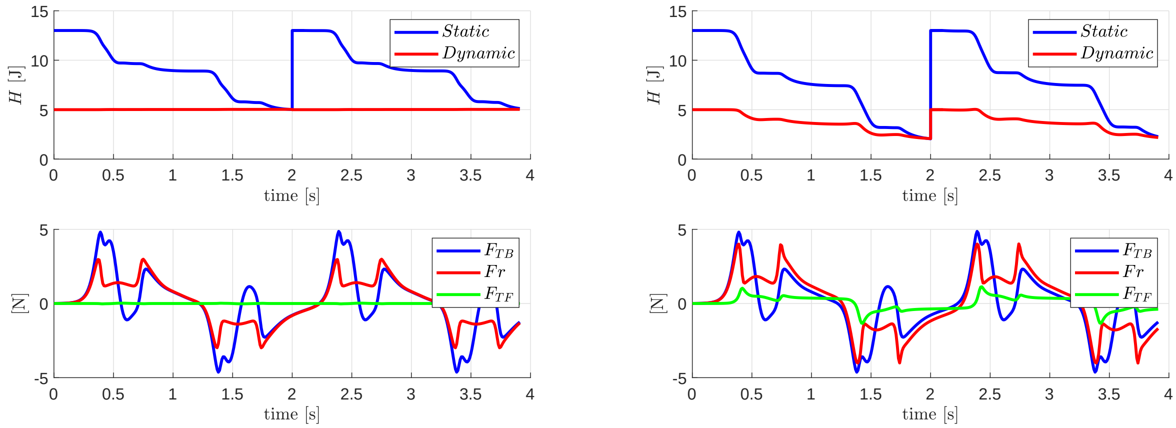
The second line in the latter equation can be utilized to express the transmission ration as a function of the controlled torque:

$$u = \begin{cases} \frac{F_c}{k_t p_t} & \text{if } H(p_t) \geq \epsilon \\ 0 & \text{otherwise} \end{cases} \tag{2.18}$$

where ϵ is a threshold close to zero. If the energy in the tank is empty, the transformation rate has to be 0 to decouple the controller from the system [7]. The energy spent to complete the task is the power lost in the resistance plus the power spent to the mechanical part, see equation (2.9). In nominal conditions (there are no perturbations) this energy is known a priori as a result from the optimization. Therefore, one can store this amount in the tank initially by setting an initial state to the tank. This type of energy storage is referred to as *static* energy injection. For every cycle, the energy can be re-injected into the tanks (by resetting the integral of the tank). The energy in the tank together with the forces on the mass for two nominal cycles are depicted in Fig. 2.9a.

2.4.1 Dynamic energy injection

The result from the optimization is a function of time. Therefore, it is exactly known a priori how much energy the task-based controller will spent over time. In face of initially storing the energy at each cycle, the energy used by the task-based controller can be dynamically added over time. This requires another power port at the side of the tank, see Fig. 2.8. The power used in a nominal cycle



(a) Nominal case (no perturbations).

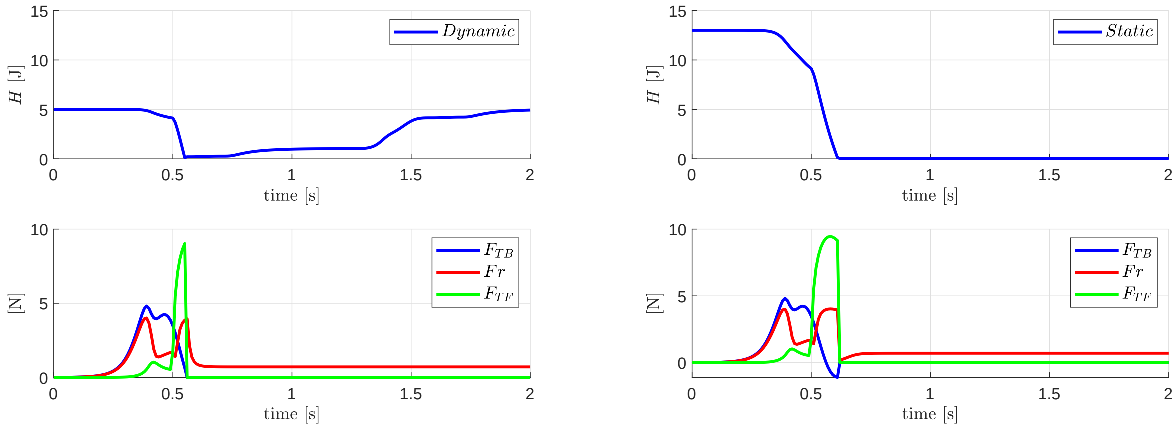
(b) Perturbed cycles.

Figure 2.9: Nominal case and perturbed case. Upper plot is the energy in the tank, lower plot are the control and friction forces acting on the mass.

($P_{cycle}(t)$) is calculated utilizing the integral argument of Eq. (2.9). To achieve this power by means of an effort-source,

$$F_{dyn}(t) = \frac{P_{cycle}(t)}{f_t} \tag{2.19}$$

where $F_{dyn}(t)$ is the source of effort connected to the tank. The power injected in the system will directly be used to perform the task. For a nominal cycle, the energy in the tank remains constant, see Fig. 2.9a. To account for any uncertainties which are compensated by the task-free controller, a small amount of additional energy is stored in the tank. Regarding the uncertainties. In real life, the nominal model used for the optimization will probably mismatch with the actual model. This is not included in the task-based controller so it will be compensated by the task-free controller. To see how this works, the 'real' friction model is perturbed with respect to the nominal friction model, see Fig. 2.11. For two cycles, the perturbed friction case for both the static and dynamic energy injection is depicted in Fig. 2.9b. Notice the energy the task-free controller consumes to compensate for the modelling error.
Another case is simulated in which the energy tank depletes completely due to a resistive element trying to stop the mass after 0.5s. Obviously, the dynamic energy injection continuous to add energy to the tank. To increase safety, it is chosen to completely decouple the tank from the rest if the tank is empty for the first time. Otherwise the control stops and continuous after a while, which is assumed to be undesired (and not safe). To illustrate what happens to the energy content in the tank and the control forces, see Fig. 2.10.



(a) Dynamic energy injection.

(b) Static energy injection.

Figure 2.10: After 0.5s an additional resistive element acts on the mass which empties the tank. Done for both dynamic (left) and static (right) energy injection types. Upper plot is the energy content in the tank, lower plot are the control and friction forces acting on the mass.

A final remark is on the difference between the dynamic energy injection and the static energy injection. Although the dynamic energy injection adds a bounded amount of energy over the cycle time, it is not theoretically passive. However, anything could be made passive with a tank so adding another tank on top which provides the dynamic energy injection could be a solution. Regarding safety, the dynamic energy injection might be safer compared to the static injection type. This is because the task-free controller can use less energy when the because there is simply less energy present in the tank. Especially when the tank is initialized (at $t = 0$s), the amount of energy in the tank is much higher compared to the dynamic energy injection, see again Figure 2.9.

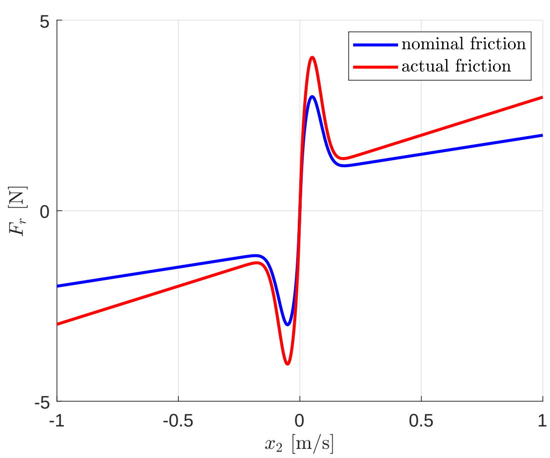


Figure 2.11: Perturbed friction

2.5 Conclusion

This section will conclude about the proof of concept. In Section 2.1 five goals were set for this proof of concept. To conclude about the proof of concept, a reflection is made based on these goals.

1. *Can a simple system benefit from a variable impedance?*
   Yes. The variable impedance is compared to an array of fixed impedances in Fig. 2.5. It was noticeable that based on the metric chosen for this positioning task, the variable impedance benefits over the fixed one.

2. What is the control scheme regarding feed-forward and feedback?
   The optimized stiffness curve is only parametrized over time. Therefore, the resulting torques of the optimization were set as an open-loop control action labelled as task-based. The rest, which was unforeseen by the task was covered by a task-free feedback controller. This was implemented a simple PD controller. One can argue that optimized task-based control action could have been parametrized as a single variable of time, which is true. However in the current implementation (parametrized like an impedance law) the task-based control action will mimic a time-varying impedance in nominal conditions. In addition, if a variable impedance actuator (VIA) was used, one has direct way to tune the physical spring and damper.

3. *What is the task and what is the subsequent metric?*
   The mass had to follow a desired position profile as closely as possible. Following the expeptional performance of humans as decision makers for varying their impedance on the basis of a task, a biomimetic metric was chosen. The corresponding metric was to minimize the tracking error while minimizing the metabolic cost as well. Clearly, the choice of metric is optional. Another option would have been: given a fixed amount of energy, what is the best performance possible?

4. *Technically: how do you implement such a problem?*
   By simulating the dynamics and solving the optimization sequentially like presented in Algorithm 1. This implementation is not the only option. Another option could have been to include the states as decision variables in the optimization and put the dynamics as a constraint. In that case, time discretization has to be done before the problem can be solved. The main advantage of the current implementation is that everything enters as a continuous time problem and the time discretization is done by `ode45` itself.

5. *How to ensure passivity of the controlled system using the energy tank concept?*
   The energy tank concept was used to recover passivity. However, the energy tanks do not guarantee safety as the amount of energy stored in the tanks can be very large, even if theoretical passivity properties are satisfied . If a very large amount of energy is stored in the tank the controller can use a lot of energy. The question is: is this very safe? If something goes wrong (for example collision with a person), it takes considerable amount of time before the controller switches off. As the metabolic cost was enclosed in the optimization, it provided a direct way to initialize the tanks. This avoided excessive energy levels in the tank. The task-based information was used to add the task-based energy in a dynamic manner. Although the

dynamic energy injection is not theoretically passive, compared to the static strategy the overall energy content in the tank is lower. That is why the dynamic energy injection increased safety.

Chapter 3

Paper

Energy-aware adaptive impedance control using offline task-based optimization

Bart Gerlagh

***Abstract*—Impedance control is a common form of robot control where interaction is important. A key question posed in this paper is how the impedance can be chosen in a structured manner given the robot's task and dynamics. We exploit impedance control ideas and make them optimal in time resulting in an open-loop control action which mimics time-varying Cartesian impedance control. We utilize energy tanks to recover passivity, in view of safety. In addition, we propose an iterative feed-forward adaptation law to account for model variations from the nominal plant using open-loop control. Simulation studies on a 5-DoF robot show efficacy of the methods by successfully performing an energy-aware and task-based peg-in-hole task using open-loop control with minimal feedback.**

***Index Terms*—Impedance Control, Time-varying, Passivity Based Control, Optimization, Robot Safety**

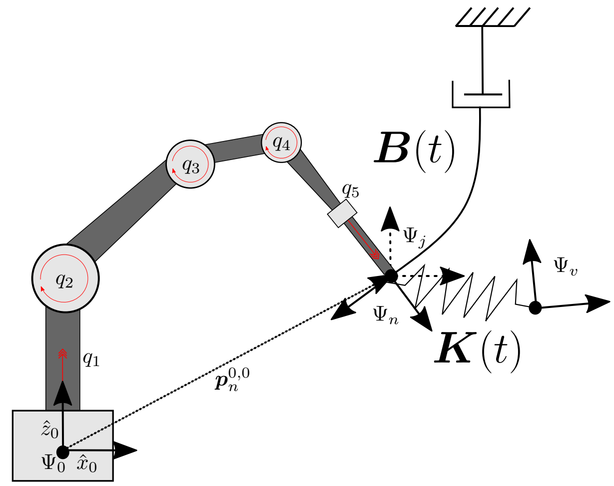


Fig. 1: A manipulator controlled with Cartesian impedance control. $\boldsymbol{p}_n^{0,0}$ represents the position vector from Ψ_0 to Ψ_n expressed in Ψ_0.

I. Introduction

Motion control classically focuses on minimizing a tracking error. However, motion control is structurally unable to deal with interactions of physical systems. When interaction is important, the problem is often cast into an impedance or admittance control framework which focuses on a relation between motion and force rather than the minimization of a signal.

A bibliography research provides a comprehensive list of applications in which impedance control is used. It ranges from human-robot interaction, industrial applications such as peg-in-hole, aerial robotics and much more. All the applications use different impedance values (characterized by a spring) on the basis of a specific task [1]–[13]. For some non-trivial tasks a variable impedance is beneficial over a fixed impedance [1]–[10]. An example of such a non-trivial task is an assembly task such as peg in hole where both following a position reference and exerting an assembly force are of importance [1]. Some author make the analogy with the human body in an attempt to copy their decision-making [3], [7]. This is for good reason; humans change their impedance all the time, they benefit from a modulated impedance for non-trivial task tasks using co-contraction of muscles [3].

A question which remains unanswered is how to structurally choose the impedance profile on the basis of a task. In other words, there exists no unified task-based framework to define impedance [14]. Moreover, even the techniques used to define variable impedance differ across literature. Many approaches used to define variable impedance are data-driven [7], [8]; treating the system as a black box and learning impedance profiles directly from sensor feedback. This is particularly the case for robots in unstructured environments, where little information about the task and dynamics is available. However, many robots operate in more structured environments (like industrial environments), where the characteristics of the environment and robot's dynamics are known. Therefore, the core idea put forward in this work is that the knowledge about the task, robot dynamics and its environment can be utilised to define 'good' initial values for varying impedance that allow completion of the task in nominal conditions.

If the task is defined in the workspace, it is convenient and effective to do impedance control in the workspace as well. In addition, the realisation of joint impedance at the end-effector depends on the robots configuration which is not constant. In contrast, with Cartesian impedance control, the end-effector impedance can be defined, which is where the task takes place. In this work we use the geometrical form of Cartesian impedance control based on the language of Lie group theory to yield a coordinate free approach [15].

In light of the missing framework to define the impedance on the basis of a given task, we exploit impedance control ideas and make them optimal in time resulting in an open-loop control action which mimics a time-varying Cartesian impedance controller on the basis of a task. The resulting task-based control action is supplemented by a task-free Cartesian impedance controller to handle environmental uncertainties and external perturbations.

The three contributions of this paper are presented is the following list:

1) The main contribution of this work is the presentation of a control strategy comprising a task-based control action which exploits time-varying Cartesian impedance control in an optimization. A major difference with existing literature is that the time-varying impedance results in an open-loop control action rather than a closed-loop strategy. In terms of speed and stability, the open-loop action is beneficial over a feedback controller. A high-gain feedback controller is less safe and more prone to stability issues due to delays in the feedback mechanism.
2) A second contribution relates to safety. Energy tanks will be used to achieve passivity of the controlled system. However, as highly debated aspect of passivity in robotic, the energy tanks do not guarantee safety as the amount of energy stored in the tanks can be very large, even if theoretical passivity properties are satisfied [16]. Therefore, we will present an *energy-aware* strategy on how to store the energy in the tanks in order to achieve passivity and safety. We validate efficacy of the methods by a simulation study of a 5-DoF manipulator along a peg-in-hole task. In addition, we exhibit a dynamic way of injecting energy into the energy tanks. The dynamic energy injection is taskbased and results in lower energy content in the tanks compared to conventional input strategies. This increases safety of the controlled system
3) A final contribution is associated with an iterative feed-forward adaptation scheme. We will present a simple adaptation law to adapt the task-based control action to such a degree in which it accounts for model variations from the nominal plant. The final result is successful task execution using an energy-aware open-loop control action which accounts for variations with minimal feedback.

In summary, we show that an optimized impedance profile is reached by minimizing a task-based metric over the impedance values. The impedance profiles resulting from this optimization defines the open-loop strategy. The task-free Cartesian impedance controller is used to account for model variations from the nominal plant (used in the optimization). We propose an iterative feed-forward adaptation law which ensures that feedback compensation for modelling errors are enclosed in the task-based feed-forward.

The structure of this paper will be as follows. Section II contains technical background for geometric Cartesian impedance control on $SE(3)$. In Section III the control strategy is proposed. Section IV presents a case study. The results of this case study are presented in Section V. The work will be concluded in Section VI.

II. Background

A. Mathematical preliminaries

This paper uses screw theory as a backbone for the proposed control strategy. It is based on the Lie algebra of rigid body motion in $SE(3)$ and will be used to define the dynamics of the robot and forward kinematic map [17]. This section presents the main mathematical notations used throughout the paper.

- Ψ_i: indicates a Cartesian frame i.
- Twist: twist of body i with respect to body j expressed in Ψ_j is written as $\boldsymbol{T}_i^{j,j} = \left[\begin{array}{cc} \boldsymbol{\omega}_i^{j,j} & \boldsymbol{v}_i^{j,j} \end{array}\right]$. $\boldsymbol{\omega}_i^{j,j}$ represents the rotational velocity and $\boldsymbol{v}_i^{j,j}$ represents the velocity of an imaginary point passing through the origin of Ψ_j [17].
- Wrench: in this work, we only consider wrenches exerted on the end-effector. $\boldsymbol{W}^n$ represent the wrench exerted on the end-effector expressed in Ψ_n. A subscript contains a generic label to indicate which component exerts the wrench, for example, we use $\boldsymbol{W}_s$ to indicate a wrench from a Cartesian spring, $\boldsymbol{W}_d$ to indicate the wrench from a Cartesian damper and $\boldsymbol{W}_e$ indicates the wrench from an (unknown) environment. A wrench written as, $\boldsymbol{W}_s^n = \left[\begin{array}{cc} \boldsymbol{\tau}_s^n & \boldsymbol{f}_s^n \end{array}\right]^\top$ defines the wrench on the manipulator from a Cartesian spring expressed in Ψ_n where $\boldsymbol{\tau}_s^n$ are the rotational forces and $\boldsymbol{f}_s^n$ are the linear forces.
- Ad(): the adjoint operator of the configuration matrix is used to change coordinates of twists and wrenches between frames like $\boldsymbol{T}_i^{i,j} = \mathrm{Ad}_{\boldsymbol{H}_j^i} \boldsymbol{T}_i^{j,j}$ maps the twist expressed in Ψ_j to Ψ_i. As the wrench is a co-vector it changes coordinates using the adjoint transpose as $(\boldsymbol{W}^i)^\top = \mathrm{Ad}^\top_{\boldsymbol{H}_i^j} (\boldsymbol{W}^j)^\top$.
- as($\cdot$): takes a square matrix and return its anti-symmetric part.
- $\tilde{(\cdot)}$: defines the skew symmetric matrix of a vector.
- tr($\cdot$): defines the trace operator of a square matrix.
- $\|\cdot\|_{FRO}$: defines the Frobenius norm of a matrix.
- $\boldsymbol{I}_n$: defines the $n \times n$ identity matrix.

B. Robot dynamics

Consider the dynamics of a robot formulated in generalized joint coordinates as,

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{\tau}_u + \boldsymbol{\tau}_e \quad (1)$$

where $\boldsymbol{q} \in \mathcal{Q}$ are the generalized coordinates and $\mathcal{Q}$ is the configuration manifold. $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})$ contains the Coriolis and centrifugal terms and $\frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}}$ represents the torques caused by gravity. $\boldsymbol{F}(\boldsymbol{q},\dot{\boldsymbol{q}})$ denotes all other torques induced by for example friction or stiffness. $\boldsymbol{\tau}_u$ are the actuator torques and $\boldsymbol{\tau}_e$ are the external torques.

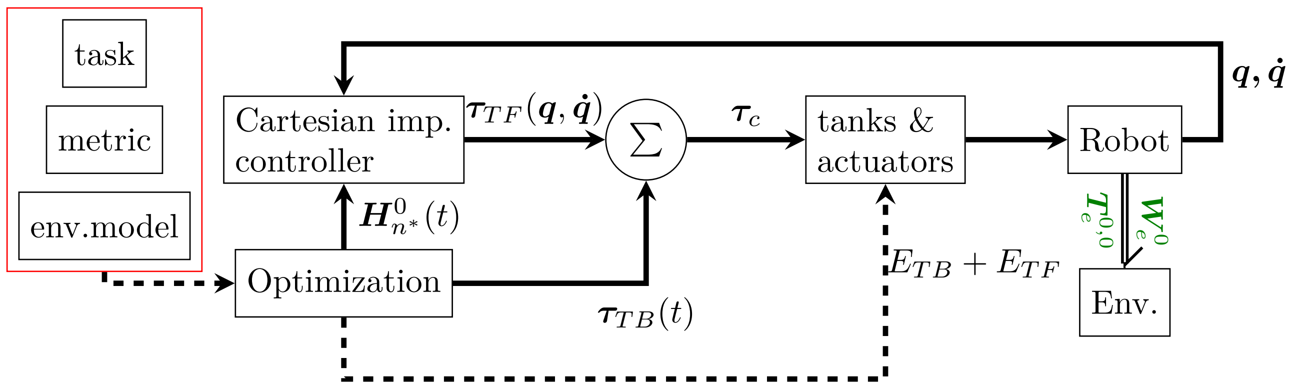


Fig. 2: 'Env.' is short for environment and 'imp.' is short for impedance. Control scheme with task-based and task-free controller. The dashed lines represent off-line signals. The robot exchanges power with the environment through power ports denoted by the multi-bond.

C. Cartesian impedance control on $SE(3)$

Cartesian impedance control is achieved by a virtual multi-DoF spring between the end-effector frame and a second virtual frame [16]. This is shown graphically in Fig. 1. Consider a frame (Ψ_n) on the end effector. Its configuration is given as the homogeneous transformation matrix,

$$\boldsymbol{H}_n^0 = \begin{pmatrix} \boldsymbol{R}_n^0 & \boldsymbol{p}_n^{0,0} \\ \boldsymbol{0} & 1 \end{pmatrix} \tag{2}$$

and represents a point on the configuration manifold $SE(3)$, i.e. $\boldsymbol{H}_n^0 \in SE(3)$. $\boldsymbol{R}_n^0 \in SO(3)$ and $\boldsymbol{p}_n^{0,0}$ describe the orientation and position of the frame expressed in the inertial frame Ψ_0. Consider a frame Ψ_v which denotes a virtual frame corresponding to the configuration $\boldsymbol{H}_v^0$. The homogeneous matrix between the virtual and the end-effector frame is

$$\boldsymbol{H}_n^v = \boldsymbol{H}_0^v \boldsymbol{H}_n^0 = \left(\boldsymbol{H}_v^0\right)^{-1} \boldsymbol{H}_n^0 = \begin{pmatrix} \boldsymbol{R}_n^v & \boldsymbol{p}_n^{v,v} \\ \boldsymbol{0} & 1 \end{pmatrix}. \tag{3}$$

The 6-dimensional spring $\boldsymbol{K} \in \mathbb{R}^{6\times 6}$ connected between the end-effector and the virtual frame can be split up into the following components in matrix form,

$$\boldsymbol{K} = \begin{pmatrix} \boldsymbol{K}_o & \boldsymbol{K}_c \\ \boldsymbol{K}_c^\top & \boldsymbol{K}_t \end{pmatrix} \tag{4}$$

where $\boldsymbol{K}_o \in \mathbb{R}^{3\times 3}$ refers to the rotational spring, $\boldsymbol{K}_t \in \mathbb{R}^{3\times 3}$ to the translational spring, $\boldsymbol{K}_c \in \mathbb{R}^{3\times 3}$ to the coupled spring (involving both rotational and translational stiffness). The wrench exerted by the spring on the end effector, $\boldsymbol{W}_s^n = \begin{bmatrix} \boldsymbol{\tau}_s^n & \boldsymbol{f}_s^n \end{bmatrix}^\top$ follows as [15, p. 168],

$$\begin{aligned}\tilde{\boldsymbol{\tau}}_s^n = -2\,\mathrm{as}\left(\boldsymbol{G}_o \boldsymbol{R}_n^v\right) - \mathrm{as}\left(\boldsymbol{G}_n \boldsymbol{R}_v^n \boldsymbol{p}_n^{v,v} \tilde{\boldsymbol{p}}_n^{v,v} \boldsymbol{R}_n^v\right) \\ - 2\,\mathrm{as}\left(\boldsymbol{G}_c \tilde{\boldsymbol{p}}_n^v \boldsymbol{R}_n^v\right)\end{aligned} \tag{5}$$

$$\begin{aligned}\tilde{\boldsymbol{f}}_s^n = -\boldsymbol{R}_v^n\,\mathrm{as}\left(\boldsymbol{G}_n \tilde{\boldsymbol{p}}_n^v\right) \boldsymbol{R}_n^v - \mathrm{as}\left(\boldsymbol{G}_n \boldsymbol{R}_v^n \tilde{\boldsymbol{p}}_n^{v,v} \boldsymbol{R}_n^v\right) \\ - 2\,\mathrm{as}\left(\boldsymbol{G}_c \boldsymbol{R}_n^v\right)\end{aligned} \tag{6}$$

where $\boldsymbol{G}_\gamma$ with $\gamma \in \{c, o, t\}$ denotes the co-stiffness:

$$\boldsymbol{G}_\gamma = \frac{1}{2}\,\mathrm{tr}(\boldsymbol{K}_\gamma)\boldsymbol{I}_3 - \boldsymbol{K}_\gamma\,. \tag{7}$$

Damping can be injected in the robot as follows. For slowly varying trajectories or when the virtual frame is at rest, the damper can be connected between the inertial frame and the end effector, as in Fig. 1. The wrench the damper exerts on the end effector can be written as

$$\boldsymbol{W}_d^n = \boldsymbol{B}\boldsymbol{T}_n^{n,0} \tag{8}$$

where $\boldsymbol{B} \in \mathbb{R}^{6\times 6}$ denotes the damping matrix and has the same structure as the stiffness matrix in Eq. (4). The resulting control law together with gravity compensation is then written as

$$\boldsymbol{\tau}_c = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W}^0 + \hat{\boldsymbol{G}}(\boldsymbol{q})\,. \tag{9}$$

$\boldsymbol{J}(\boldsymbol{q})$ denotes the geometric Jacobian which columns are the unit twists in inertial coordinates [15]. The wrench from the Cartesian spring and damper on the end-effector needs to be expressed in Ψ_0:

$$(\boldsymbol{W}^0)^\top = \mathrm{Ad}_{\boldsymbol{H}_0^n}^\top (\boldsymbol{W}_s^n + \boldsymbol{W}_d^n)^\top\,. \tag{10}$$

III. Control strategy

In this work we exploit the benefits of a time-varying impedance. This section presents a control strategy comprising an open-loop control term optimised using time-varying impedance, and a task-free Cartesian impedance controller to account for model variations from the nominal plant. The control architecture is given in Fig. 2 and will be explained in the following subsections. On top of that, it presents how the energy tanks are used to recover passivity of the controlled system in view of safety.

A. Control overview

As indicated in Section I, the impedance values will be parametrized over time, independent on the configuration of the robot. Since the resulting impedance varies only in time (and not in space) it belongs to an open-loop control action. It is not expedient to 'blindly' vary the stiffness in a feedback controller. On top of that, an open-loop controller results in a faster response and does not face stability issues due to feedback delays. The task-based control action will be calculated using time-varying impedance values in the Cartesian impedance control framework. The problem of finding the time-varying impedance values is cast into an optimization problem utilizing the definition of the task and a task-based objective, given the robot's dynamics and environment. Section III-B will go in detail about this optimization. The optimization will not only result in time-varying impedance profiles. The trajectory of the end-effector is also (implicitly) part of the optimization, it defines the optimal end-effector trajectory in nominal conditions as a function of time, $\boldsymbol{H}_n^0(t)^*$. The optimized torque profile will be labelled as 'task-based (TB)' control action since it depends on the definition of a task. Everything else, which is unforeseen by the task is labelled as 'task-free (TF)'. This task-free controller is a feedback controller and will be implemented as a separate Cartesian impedance controller. The control law is then the combination from feed-forward (task-based) and feedback (task-free):

$$\boldsymbol{\tau}_c(t, \boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{\tau}_{TB}(t) + \boldsymbol{\tau}_{TF}(\boldsymbol{q}, \dot{\boldsymbol{q}})\,. \tag{11}$$

The task-based control action supplemented with the task-free impedance controller is depicted in Fig. 2. How the task-based and task-free parts arise becomes clear in Sections III-B and III-C. The use of energy tanks to recover passivity in view of safety will be explained in III-D.

B. Task-based optimization

The optimal impedance is found by minimizing a task-based objective function. Humans are great examples of varying their impedances for different complex tasks by co-contraction of muscles [18]. Furthermore, humans follow the same control strategy as presented in this work: they learn optimizing their impedances on the basis of a task by performing the task a great number of times and apply the 'learned' impedance profiles in an open-loop fashion rather than in a closed-loop fashion [18]. This is because the human sensing mechanism faces delay of $\approx 100\,ms$ which is prone to stability issues using a high-gain feedback controller [18]. In addition, humans make use of feedback to account for unforeseen changes along the task which is analogous to the task-free controller presented in this work [3]. Following the exceptional performance of humans we try to find optimal impedance based on a biomimetic metric. [3], [7] suggests that the human central nervous system (CNS) tries to minimize the energy spent on a task while maintaining the performance. This biomimetic choice of metric fits well within the tank-based paradigm as it gives immediate information on how much energy should be stored in the tanks, more on this is Section III-D. We start from Eq. (9) as a control law to define the task-based control action:

$$\boldsymbol{\tau}_{TB} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W}^0 + \hat{\boldsymbol{G}}(\boldsymbol{q}) \tag{12}$$

where the applied wrench, $\boldsymbol{W}^0$ from the geometric spring and damper depend on the values of $\boldsymbol{K}$ and $\boldsymbol{B}$. The general goal of the optimization is to find an optimal time-varying $\boldsymbol{K}(t)^*$ and $\boldsymbol{B}(t)^*$ by minimizing a biomimetic cost,

$$\boldsymbol{K}(t)^*, \boldsymbol{B}(t)^* = \underset{\boldsymbol{K}(t), \boldsymbol{B}(t)}{\operatorname{argmin}}\; J_p + J_m \tag{13}$$

where J_m is an objective covering the energy spent to perform the task, and J_p is a task-based performance objective. The performance objective can be split up into a running cost (J_r) (integrated over the time domain) and a final cost (J_f) term at time $t = T$:

$$J_p = \int_0^T J_r(t)dt + J_f(T)\,. \tag{14}$$

The formulation of the cost terms will be elaborated for the case study in Section IV-D.

C. Task-free Cartesian impedance controller

The task-free torques are calculated in a similar way to the task-based torques without gravity compensation:

$$\boldsymbol{\tau}_{TF} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W}_{TF}^0\,. \tag{15}$$

It utilizes as set point, the optimized trajectory $\boldsymbol{H}_n^0(t)^*$ as the virtual frame. The wrench from the task-free Cartesian spring and damper in the inertial frame is written as

$$(\boldsymbol{W}_{TF}^0)^\top = \mathrm{Ad}^\top_{\boldsymbol{H}_0^n}(\boldsymbol{W}^n_{s_{TF}} + \boldsymbol{W}^n_{d_{TF}})^\top\,. \tag{16}$$

$\boldsymbol{W}^n_{s_{TF}}$ is the wrench on the end-effector due to the task-free spring calculated using Eqs. (5) and (6) and $\boldsymbol{K}_{TF}$ as the spring constant. The wrench induced by the task-free damper is written as,

$$\boldsymbol{W}^n_{d_{TF}} = \boldsymbol{B}_{TF}\boldsymbol{T}_n^{n,n^*} \tag{17}$$

where

$$\boldsymbol{T}_n^{n,n^*} = \mathrm{Ad}_{\boldsymbol{H}_0^n}(\boldsymbol{T}_n^{0,0} - \boldsymbol{T}_{n^*}^{0,0}) \tag{18}$$

and where $\boldsymbol{T}_{n^*}^{0,0}$ is the optimized end-effector twist:

$$\tilde{\boldsymbol{T}}_{n^*}^{0,0} = \dot{\boldsymbol{H}}_{n^*}^0 \boldsymbol{H}_0^{n^*}\,. \tag{19}$$

D. Passivity by energy tanks

It was shown in [19] that a non-passive system can become unstable even when interacting with passive environments. This is why passivity is important. A system is said to be passive if there is a storage function $\mathcal{V} : \mathcal{X} \mapsto \mathbb{R}^{+}$, where $\mathcal{X}$ is the state manifold such that [20],

$$\int_{t_0}^{t_1} \dot{\mathcal{V}}(\boldsymbol{x})\,dt = \mathcal{V}(\boldsymbol{x}_1) - \mathcal{V}(\boldsymbol{x}_0) \leq \int_{t_0}^{t_1} \boldsymbol{y}^{\top}\boldsymbol{u}\,dt \tag{20}$$

with state $\boldsymbol{x}$, input $\boldsymbol{u}$, output $\boldsymbol{y}$ and $t \in [t_0, t_1]$. When a variable impedance is used, passivity will be destroyed by the creation of additional power ports. Additionally, using a moving virtual frame (hence, $\dot{\boldsymbol{H}}_v^0 \neq \boldsymbol{0}$) will also create additional power ports which again destroys passivity, analogous to an one DoF mass[1]. To recover passivity in an elegant way, the concept of an energy-tank was introduced for multi-DoF robots in [16]. The energy-tanks are represented by physical storage elements connected to each joint. This ensures that each actuator can use a bounded amount of energy which guarantees passivity fulfilling the inequality in Eq. (20). In this work, the tanks are modelled as an inductance with inductance $L = 1\,H$. The energy stored in each tank j is given as $H_j(p_j) = \frac{1}{2}p_j^2$ where p_j is the state of tank. Energy can be added to the tanks by defining an initial state for p_j or by the creation of an energy source modelled as a voltage source with voltage U_j. Each motor can drawn energy from its corresponding tank by means of a modulated transformer. A schematic diagram of the manipulator, connected to each tank driven by the motors is depicted in Fig. 3. The motor inductance is neglected so that each motor with torque output τ_j, consists of an electrical resistor with resistance R_j and a gyrator with torque constant k_t. The port-Hamiltonian equations for each joint j, connected to a tank can be written as

$$\begin{bmatrix}\dot{p}_j \\ \tau_j\end{bmatrix} = \begin{bmatrix}-u_j^2 R_j & -u_j k_t \\ u_j k_t & 0\end{bmatrix}\begin{bmatrix}p_j \\ \dot{q}_j\end{bmatrix} + \begin{bmatrix}1 \\ 0\end{bmatrix} U_j\,. \tag{21}$$

The second line in Eq. (21) allows to set a desired transmission ratio as a function of the controlled torque:

$$u_j = \begin{cases}\frac{\tau_j}{k_t p_j} & \text{if } H_j(p_j) \geq \epsilon \\ 0 & \text{otherwise}\end{cases}\,. \tag{22}$$

To ensure passivity, we choose to decouple all tanks from their corresponding joint when energy in one of the tanks is empty. When one of the tanks empties, a large amount of damping is applied in the joint space which is by definition a passive action:

$$\boldsymbol{\tau}_c = -\bar{\boldsymbol{B}}\dot{\boldsymbol{q}} \tag{23}$$

[1] An one DoF mass controlled with impedance control with external force F_e and desired state x^*. The time derivative of the storage function is defined as $\dot{\mathcal{V}} = \dot{x}F_e + k(t)\tilde{x}\dot{x} - b(t)\dot{x}^2 + k(t)\tilde{x}\dot{\tilde{x}} + \frac{1}{2}\dot{k}(t)\tilde{x}^2$ where $\tilde{x} = (x^* - x)$. $\dot{x}^*$ and $\dot{k}$ create additional power ports which may destroy passivity.

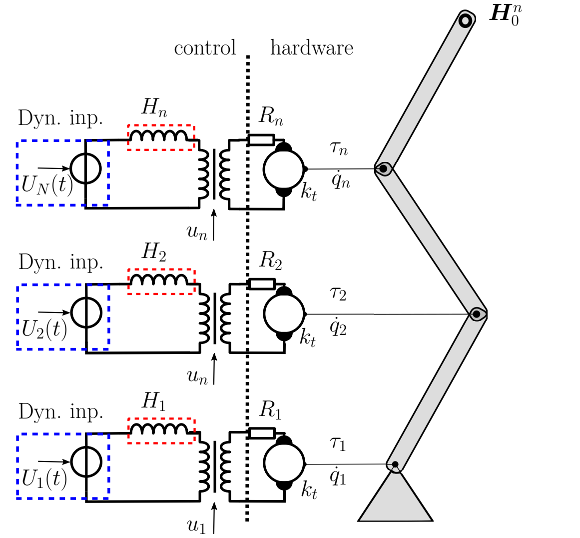


Fig. 3: Schematic multi-DoF robot with tank-subsystems

where $\bar{\boldsymbol{B}} \in \mathbb{R}^{n\times n}$ is a diagonal damping matrix. Furthermore, we do not allow recharging of the tanks when power flows back from the actuator to increase safety. This avoids the robot gaining energy from colliding with objects. The amount of energy to be stored in the tanks at $t = 0$ is then the task-based energy plus some additional task-free energy to cope with disturbances:

$$p_j(0) = 2 \cdot \sqrt{E_{TB_j} + E_{TF_j}} \tag{24}$$
$$U_j(t) = 0 \tag{25}$$

where E_{TB_j} and E_{TF_j} are the energy content in tank j allocated for the task-based and task-free part respectively. This type of energy injection is referred to as *static*. Instead of adding all the energy at $t = 0$ we can choose to dynamically add in the task-based energy over time (as it is known a priori from optimization of the nominal case). This will increase safety as the overall energy content in the tanks can be minimized which ensures that the task-free controller can use less energy than when the energy is stored in a static way. For the purpose of dynamic injection we use the voltage source at the side of the tank providing a bounded amount of task-based energy (E_{TB_j}) over time. The amount of energy to be stored at $t = 0$ is only the task-free energy:

$$p_j(0) = 2 \cdot \sqrt{E_{TF_j}} \tag{26}$$
$$U_j(t) = \frac{P_{TB_j}(t)}{p_j} \tag{27}$$

where P_{TB_j} is the task-based power usage corresponding to joint j. This type of energy injection is referred to as *dynamic*.

E. Iterative feed-forward adaptation

The task-free feedback compensation ensures that the robot can achieve successful task-execution under model variations. As mentioned earlier, a high-gain feedback controller is prone to stability issues due to time delays in the feedback mechanism. That is why we update the task-based control action in a way such that it also accounts for the model variations. Many (industrial-like) tasks are carried out sequentially. When the task is completed successfully, on the next iteration the task-based controller can be updated in such away to adapt the open-loop controller to model variations, and diminish feedback action. The adaptation law for the task-based torques becomes,

$$\boldsymbol{\tau}_{TB}^{i+1} = \boldsymbol{\tau}_{TB}^{i} + \boldsymbol{\tau}_{TF}^{i} \tag{28}$$

where $i \in \mathbb{Z}$ is the i-th iteration. As will be shown later, after a small number of iterations this ensures that the end-effector will follow $\boldsymbol{H}_{n^*}^{0}(t)$ using only open-loop control under model variations. In addition to updating the task-based torques, the amount of energy allocated in the tanks for the task-free controller can be decreased as the model-variation compensation is now enclosed in the open-loop controller. Therefore, also the energy in the tanks assigned to the task-based and task-free part can be updated in the next iteration according to,

$$E_{TF}^{i+1} = E_{task}^{i} - E_{TB}^{i} + E_{\varepsilon} \tag{29}$$

$$E_{TB}^{i+1} = E_{task}^{i} \tag{30}$$

where E_{TF} and E_{TB} are the total task-free and task-based energy levels for all joints. E_{task} is the total amount of energy to complete the task and E_{ε} is just small amount of energy which is always present to account for minor disturbances. In this way, the tanks become energy-aware as only the necessary amount of energy is stored in the tanks for successful task execution under model variations which increases safety while recovering passivity.

IV. Case study

In this section a case study is presented in which the efficacy of the proposed control method will be tested. The task considered in this paper is an insertion task, similar to peg-in-hole. The peg-in-hole task is considered to be a benchmark for industrial robotics.

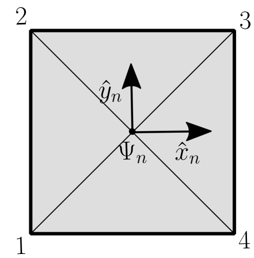
A. Modelling

L_1	L_2	L_3	L_4	L_5	m_1	m_2	m_3	m_4	m_5
0.14	0.16	0.13	0.08	0.04	1.4	1.6	1.3	0.8	0.4

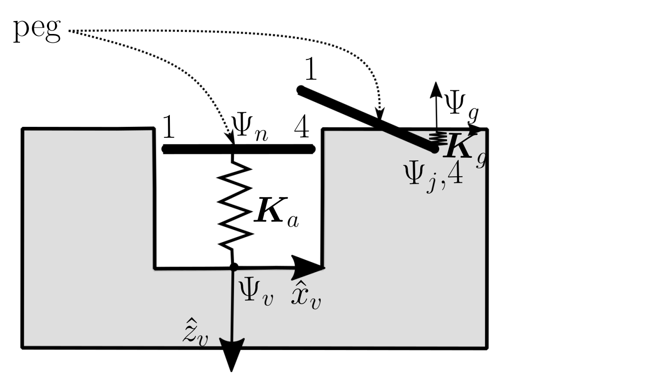
TABLE I: Lengths are in [m], masses are in [kg]. m_5 includes both the weight for the peg and the fifth arm. Dimensions and weights are based the KUKA youBot's arm.

A model of a 5-DoF robotic arm is used which corresponds to the kinematic chain in Fig. 1. The lengths and the masses for each arm are given in Table I. The nominal dimensions for the peg, joint friction, ground stiffness and assembly stiffness can found in Table II. Furthermore, we set the maximum torque output for each motor to 50 Nm. The robots dynamics are formulated in generalized joint coordinates as in Eq. (1). The generalized configuration variables are defined as

$$\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 \end{bmatrix}^\top . \tag{31}$$



(a) Square peg in the XY plane. Peg has 5 contacts points (4 corners and middle point)



(b) Hole and environment in the XZ plane. 2 situations are displayed here: 1 in which the peg enters the hole and another one in which the peg is in contact with the ground.

Fig. 4: Contact modelling

Fig. 4 demonstrates the dimensions and geometry of the peg and the hole. The peg has five points which can make contact with the environment (4 corners and middle point). They can exchange power with the environment through interaction ports (see Fig. 2). Consider a frame Ψ_j that denotes the frame aligned with the ground at the end-effector (see Fig. 1). If the peg is in contact with a position on the ground outside the hole ($\boldsymbol{p_g^{0,0}}$) corresponding to Ψ_g, then the ground will exert a translational force on the peg,

$$\boldsymbol{f}_e^j = \boldsymbol{K}_g \boldsymbol{p}_n^{j,g} \tag{32}$$

where

$$\boldsymbol{p}_n^{j,g} = \boldsymbol{R}_0^j \left(\boldsymbol{p}_n^{0,0} - \boldsymbol{p}_g^{0,0} \right) . \tag{33}$$

In Eq. (32), $\boldsymbol{K}_g \in \mathbb{R}^{3\times 3}$ is the stiffness matrix of the ground. To simulate an insertion task, a linear spring is modelled inside the hole. This ensures that the robot has to exert a certain force in order to reach Ψ_v. This linear force is calculated in the same way as in Eq. (32) using $\boldsymbol{K}_a$. The resulting external wrench acting on the manipulator is composed as $\boldsymbol{W}_e^j = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{f}_e^j \end{bmatrix}$. Then wrench is defined in the inertial frame,

$$(\boldsymbol{W}_e^0)^\top = Ad^\top_{\boldsymbol{H}_0^j}(\boldsymbol{W}_e^j)^\top \tag{34}$$

To finally express the torques caused by the interaction, the geometric Jacobian is used:

$$\boldsymbol{\tau_e} = \boldsymbol{J(q)}^\top \boldsymbol{W}_e^0 \,. \tag{35}$$

The remainder of this section will relate to the optimization problem posed in Eq. (13) applied to this case study.

B. Cartesian damping

For linear systems the damper is commonly chosen such that the closed loop system becomes critically damped. For general multi-DoF manipulators with nonlinear dynamics this is not trivial. When $\dot{\boldsymbol{q}}$ and the gravitational torques are either zero or cancelled, then the end-effector acceleration can be written as,

$$\dot{\boldsymbol{T}}_n^{0,0} = \boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}\boldsymbol{W}^0 \tag{36}$$

where $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$ is

$$\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1} = \boldsymbol{J(q)M(q)}^{-1}\boldsymbol{J(q)}^\top \,. \tag{37}$$

$L(\boldsymbol{q}) : \mathcal{Q} \mapsto \mathcal{W}$ is the forward kinematic map from the joint space to the workspace. $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$ is called the *end-point mobility tensor* [21].It describes the acceleration response of the end-effector when the system is at rest. $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$ is not positive definite by nature, it scales with $\boldsymbol{J(q)}$. The inverse of $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$ exists if and only if $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$ is non-singular. Which is true for non-singular positions and for non-underactuated robots [21]. [22] introduces an upper bound inertia matrix $\boldsymbol{I_u}$, which is diagonal (and hence positive definite). If the inverse of $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$ exists then,

$$\frac{1}{2}(\boldsymbol{T}_n^{0,0})^\top \boldsymbol{I_u}\boldsymbol{T}_n^{0,0} > \frac{1}{2}(\boldsymbol{T}_n^{0,0})^\top \boldsymbol{\Lambda}(L(\boldsymbol{q}))\boldsymbol{T}_n^{0,0} \tag{38}$$

which is based on the principle of Loewner ordering [22] where $\boldsymbol{I_u} = m \cdot \boldsymbol{I}_6$, $m \in \mathbb{R}$. m can always be chosen such that the inequality in Eq. (38) holds. Eq. (38) can be written differently to circumvent the need of taking the inverse of $\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}$. The right hand side of this equation represents the kinetic energy and can also be written in joint space [22],

$$\begin{gathered}\frac{1}{2}(\boldsymbol{T}_n^{0,0})^\top \boldsymbol{I_u}\boldsymbol{T}_n^{0,0} > \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M(q)}\dot{\boldsymbol{q}} \\ \left\{\forall \boldsymbol{q} \in \mathcal{Q} \colon rank(\boldsymbol{\Lambda}(L(\boldsymbol{q}))^{-1}) = 6\right\} \,. \end{gathered} \tag{39}$$

Now m can be calculated without the need of taking any inverse. If one replaces $\boldsymbol{\Lambda}(L(\boldsymbol{q}))$ with the diagonal $\boldsymbol{I_u}$ then the acceleration response is written as,

$$\dot{\boldsymbol{T}}_n^{0,0} = \boldsymbol{I_u}^{-1}\boldsymbol{W}^0 \tag{40}$$

where $\boldsymbol{W}^0$ is the wrench applied by the spring and damper on the end effector expressed in inertial coordinates. Consider $\boldsymbol{K} = \mathrm{diag}(\boldsymbol{K_0}, \boldsymbol{K_t})$ and $\boldsymbol{B} = \mathrm{diag}(\boldsymbol{B_0}, \boldsymbol{B_t})$ where $\boldsymbol{B}_t = b_t \cdot \boldsymbol{I_3}$, and $\boldsymbol{K}_t = k_t \cdot \boldsymbol{I}_3$. The linear force from the spring on the end effector can then be calculated using Eq. (6) and express it in Ψ_0 using the Ad() operator [22]:

$$\boldsymbol{f}^0 = \boldsymbol{K_t}\boldsymbol{p}_n^{0,v} \,. \tag{41}$$

The equation of motion in (40) is now decoupled in the translational directions. Using $\boldsymbol{I_u}$ as a 'worst-case' inertia allows to apply control techniques used for systems with only one DoF. Setting a desired end-effector inertia would require an inverse of the Jacobian or the desired inertia matrix. The translational damper is chosen such that the system becomes critically damped in translational direction:

$$b_t = 2\sqrt{k_t m} \,. \tag{42}$$

This choice of the translation damper leaves a straightforward implementation where the translational damper scales with the inertia at the end-effector and the translational stiffness of the Cartesian spring without the need of taking any inverse. For that reason, the same has been applied to the rotational damper ($b_o = 2\sqrt{k_o m}$). However, as the geometric wrench is a coordinate free approach, the term 'critically' damped is not adequate.

C. Cartesian stiffness parameterization

The structure of the Cartesian spring is chosen to be diagonal where the translational spring is split up into planar (in XY direction) and vertical (in Z direction) components as,

$$\boldsymbol{K} = \mathrm{diag}(\boldsymbol{K}_o, \boldsymbol{K}_p, k_v) \tag{43}$$

where $\boldsymbol{K_o} = k_o \cdot \boldsymbol{I_3}$ and $\boldsymbol{K_p} = k_p \cdot \boldsymbol{I_2}$. The distinction between the translational spring in planar and vertical direction is made because the end-effector has to exert a force in vertical direction within the hole (see Fig. 4). The Cartesian spring will be modulated over time. Since the optimization problem posed in Eq. (13) is of infinite-dimensional nature (i.e. would require gradient descent in a functional space to be solved), for a practical implementation we seek the solution in a finite dimensional space defined by N parameters defining the stiffness curve as a B-spline [23],

$$k_\gamma(t) = \sum_{i=1}^{N} \alpha_{\gamma,i} B_i(t) \tag{44}$$

where $\gamma \in \{o, p, v\}$. $B_i(t)$ are the basis functions and $\alpha_{\gamma,i}$ are the coefficients to be solved for with equal time spacing between adjacent coefficients. This parametrization allows for a discretized problem formulation while the final result will be smooth in time.

D. Objective

The total objective function to be minimized consists of several element. The optimization problem will be defined as,

$$\begin{aligned} \mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\operatorname{argmin}}\, J_t + J_m + J_c \\ \text{s.t,} \\ k_\gamma^+ \geq k_\gamma(\mathbf{\Theta}_\gamma, t) \geq k_\gamma^- \quad \forall t, \gamma \end{aligned} \tag{45}$$

where k_γ^-, k_γ^+ denote the lower and upper stiffness bounds. J_t comprises the task-based performance objective, J_m defines the metabolic cost and J_c is an objective term to maximize compliance. $\mathbf{\Theta} \in \mathbb{R}^m$ is the collection of B-spline coefficients for each stiffness parametrization where

$$\mathbf{\Theta}_\gamma = \begin{bmatrix} \alpha_{\gamma,1} & \dots & \alpha_{\gamma,N} \end{bmatrix}. \tag{46}$$

In the following each objective term will be dissected and motivated where $w_1, w_2, \mu_1, \mu_2, \in \mathbb{R}$ and $\boldsymbol{W_3} \in \mathbb{R}^{m \times m}$ are the weighing constants, used to scale the objective terms.

1) Performance: The main objective of assembly tasks is to insert an object into a shape which is either a 'pass' or a 'fail'. A virtual frame is placed within the hole (see Fig. 4), if the end-effector configuration is close enough to this point the task is being said to succeed. This performance is formulated as a final cost at final time T,

$$J_t = w_1 \left\| \boldsymbol{H}_n^v(T) \right\|_{FRO}^2 \tag{47}$$

where $T = \min(t)$ s.t. $\|\boldsymbol{H}_n^v(t)\|_{FRO} < \varepsilon$, otherwise T equals the simulation time. $\varepsilon \in \mathbb{R}_+$ is a properly chosen threshold.

2) Metabolic cost: The metabolic cost is defined as the energy spent to perform the task. The metabolic cost is considered to be the power loss in the electrical resistance plus the power output to the mechanical domain. The power lost to the electrical resistance in motor j with motor current i_j is proportional to τ_j:

$$P_{R,j} = i_j^2 R_j = \frac{\tau_j^2}{k_t^2} R_j \propto \tau_j^2 . \tag{48}$$

The power transferred to the mechanical domain is

$$P_{m,j} = \tau_j \dot{q}_j . \tag{49}$$

The motors do not contain physical storage elements nor is the motor driver assumed to be a four-quadrant capable driver. Therefore in order to express the metabolic cost only the positive powers are assumed as this is the power spent to accomplish the task. The metabolic cost is then

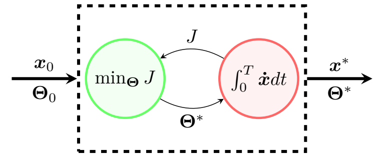


Fig. 5: Schematic algorithm to solve problem posed in equation (45) .

formulated as the sum of metabolic costs: for each motor as,

$$E_m = \sum_{j=1}^{n} \int_0^T (P_{R,j} + P_{m,j})^+ dt \tag{50}$$

where $(\cdot)^+$ denotes only the positive part of the input argument. The corresponding objective denotes the weighted metabolic cost:

$$J_m = w_2 \sum_{j=1}^{n} \int_0^T (\mu_1 P_{R,j} + \mu_2 P_{m,j})^+ dt . \tag{51}$$

3) Interaction energy: Conventionally, peg-in-hole applications consider to minimize the interaction forces [11], [13]. Instead, we choose to minimize the energy transfer as it has more physical meaning than minimizing the interaction forces as a signal. The interaction energy is formulated as the integral over the absolute interaction power at the interaction port (see Fig. 2),

$$E_e = \int_0^T |(\boldsymbol{W}_e^0)^\top \boldsymbol{T}_e^{0,0}| dt \tag{52}$$

where $\boldsymbol{W}_e^0$ and $\boldsymbol{T}_e^{0,0}$ are the effort and flow at the interaction port. Minimizing the interaction energy is equivalent to minimizing the impedance or minimizing the interaction velocity in the frequency domain [12]. The objective for this part will be based on minimizing the impedance as it is a direct optimization variable

$$J_c = \|\mathbf{\Theta}\|_{\boldsymbol{W_3}}^2 = \mathbf{\Theta}^\top \boldsymbol{W_3} \mathbf{\Theta} . \tag{53}$$

E. Algorithm

The optimization problem in equation (45) is solved using MATLAB 2020a. `fmincon` with the `'interior-point'` algorithm will be used as the problem is posed as a constrained and non-linear minimization problem. The dynamics are solved using the `ode15s` solver. In every objective evaluation the dynamics are simulated which ensures that dynamics and minimization problem are solved sequentially, as shown in Fig. 5. The state-space equations for the system can be written in general form using Eq. (1) as,

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{t}, \boldsymbol{\tau}_c(\mathbf{\Theta})) \tag{54}$$

where $\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix}$. $\boldsymbol{\tau}_c$ denotes the control law in Eq. (11). The main advantage of this implementation is that the problem enters as a continuous time problem and time discretization is done by the `ode` solver itself. In this case, the `ode` solver will use a variable time step which is useful especially because when interaction takes place, a smaller time step is used compared to when the robot moves in free space. In addition, when the peg is inserted ($\|\boldsymbol{H}_n^v(T)\|_{FRO} < \varepsilon$) the `ode` solver is stopped, which defines the completion time T in Eqs. (47) and (51). This allows the optimization to make a trade-off between completion time and energetic cost.

The task is split up into two subtasks. The first subtask is the carrying task which causes the robot to hang above the hole. It is defined by a static virtual frame corresponding to the configuration $\boldsymbol{H}_{v_c}^0$. The robot successfully reaches this frame at time $t_c = \min(\mathrm{t})$ s.t. $\|\boldsymbol{H}_n^{v_c}(t)\|_{FRO} < \varepsilon_c$. Afterwards, the virtual point is moved to a location within the hole corresponding to $\boldsymbol{H}_{v_i}^0$ (see Fig. 4), this is referred to as the insertion task. To summarize,

$$\boldsymbol{H}_v^0 = \begin{cases} \boldsymbol{H}_{v_c}^0 & \text{for carrying subtask} \\ \boldsymbol{H}_{v_i}^0 & \text{for insertion subtask} \end{cases} . \tag{55}$$

Both subtasks are solved separately. The final state and stiffness endpoints (and its derivatives) of the carrying subtask are then provided as constraint to the insertion subtask optimization. This ensures the combined result remains a smooth function of time.

V. Results

To obtain the results we use the following configurations for the virtual frame:

$$\boldsymbol{H}_{v_c}^0 = \begin{pmatrix} -1 & 0 & 0 & 0.2 \\ 0 & 1 & 0 & 0.16 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{56}$$

with $\varepsilon_c = 0.01$ and

$$\boldsymbol{H}_{v_i}^0 = \begin{pmatrix} -1 & 0 & 0 & 0.2 \\ 0 & 1 & 0 & 0.16 \\ 0 & 0 & -1 & -0.05 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{57}$$

with $\varepsilon = 0.008$. The motor constant and resistance in each motor are set to $k_t = 1\,\mathrm{Nm/A}$ and $R = 1\,\Omega$.

A. Optimization in nominal conditions

Fig. 6c displays the resulting stiffness curves from the optimization. For the carrying phase we use: $w_1 = 10^8$, $w_2 = 1$, $\boldsymbol{W}_3 = (5 \cdot 10^{-3}) \cdot \boldsymbol{I}_{12}$. For the insertion phase we use: $w_1 = 10^5$, $w_2 = 10^{-1}$, $\boldsymbol{W}_3 = (5 \cdot 10^{-3}) \cdot \boldsymbol{I}_{12}$. The weights are chosen such that the metabolic cost objective is dominant in the carrying subtask and the task related objective is dominant in the insertion subtask. For both subtasks holds $\mu_1 = \mu_2 = 1$. The initial state, $\boldsymbol{x}_0$ in the carrying task are set to: $\boldsymbol{q}_0 = \begin{bmatrix} 0 & 0 & 0 & \pi/2 & 1 \end{bmatrix}^\top$ and $\dot{\boldsymbol{q}}_0 = \boldsymbol{0}$. Figs. 6a and 6b show the objective evaluations for each subtask. The variable impedance result is compared to a fixed (stiffness) impedance. Figs. 6e and 6f show a comparison on the basis of the interaction energy and metabolic. The gains for the fixed impedance are chosen to be optimal on the basis of the same metric if only a fixed impedance is used (i.e solving the optimization for a fixed stiffness, $\boldsymbol{K}_{fixed}$. We only make a distinction between the rotational and translational component. This resulted in $\boldsymbol{K}_{fixed} = \mathrm{diag}(600 \cdot \boldsymbol{I}_3, 8000 \cdot \boldsymbol{I}_3)$.

The results are clearly interpretable:

1) Metabolic cost: within the carrying subtask, the stiffness curves are mainly optimized on metabolic cost, see Fig. 6a. As the difference between the virtual frame and the end-effector configuration is largest at time zero, excessive wrenches on the manipulator can be reduced by being more compliant at that time. This will have the effect of reducing the metabolic cost. Compared to the fixed impedance, higher compliance of the time-varying impedance in the early part of the task substantially reduces metabolic cost.
2) Compliance: in the positioning phase, vertical translational stiffness k_v will increase until $8000\,\mathrm{N/m}$ which is the expected stiffness to exert the necessary vertical force to accomplish the task[2]. This also explains the major decrease in final cost in Fig. 6b as the initial k_v is not stiff enough. Rotational stiffness k_0 and planar translational stiffness, k_p become more compliant in the insertion phase as the robot only needs to move in vertical direction. This provides a reduction in the impedance cost term J_c.
3) Interaction energy: the fixed impedance exchanges more energy with the environment than the variable impedance. This is because the fixed impedance shows some overshoot, leading to contact with the side of the peg hole. This becomes evident looking at the energetic exchange in Fig. 6e. The variable impedance will only interact with the vertical spring inside the hole itself.

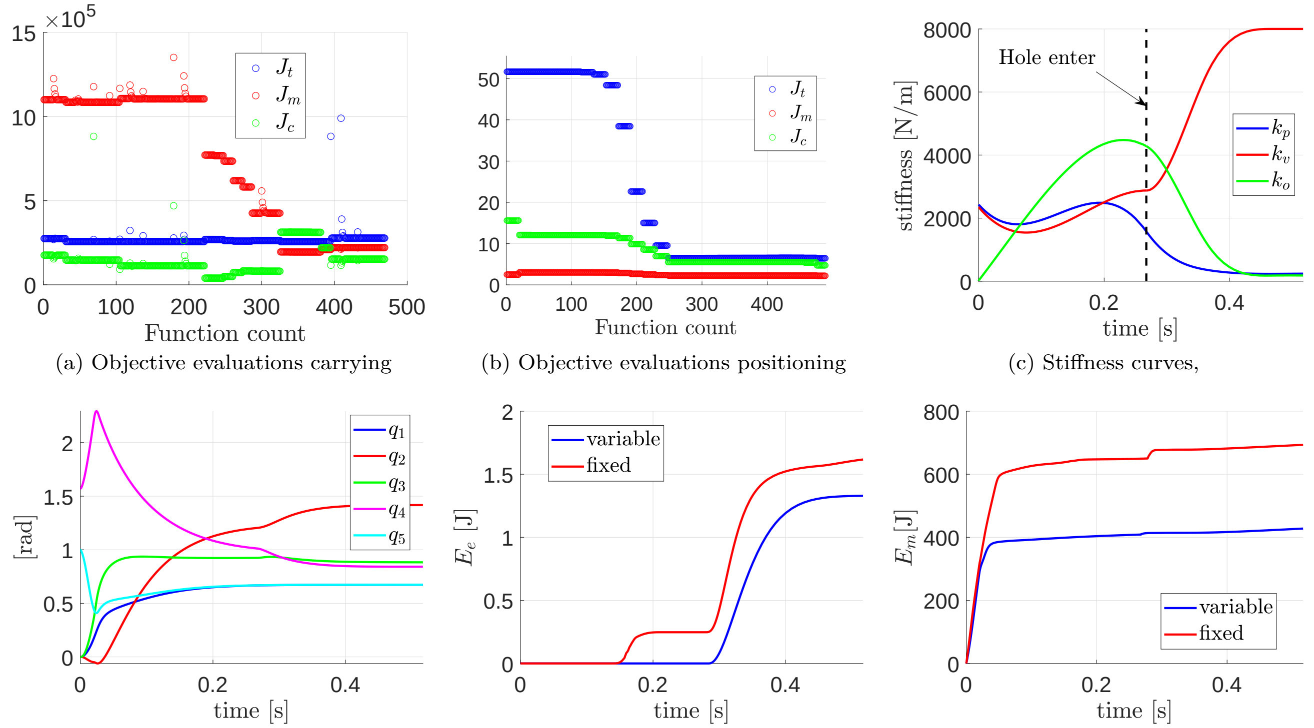
B. Robust performance under parameter variation

As introduced in Section III, the task-free feedback controller serves to reject unmodelled disturbances arising from system variations and external disturbances. Let the the nominal parameter set be denoted by $\hat{\boldsymbol{\rho}}$. The 'real' parameter set $\boldsymbol{\rho}$ is defined according to a normal distribution,

$$\boldsymbol{\rho} \sim \mathcal{N}(\hat{\boldsymbol{\rho}}, \boldsymbol{\Sigma}) \tag{58}$$

where $\boldsymbol{\Sigma}$ is a diagonal variance matrix containing the standard deviation for each parameter. The parameters which are considered to vary are the viscous friction in the joints b_j, ground stiffness $\boldsymbol{K}_g$, the stiffness within the

[2]The robot has to exert $63\,\mathrm{N}$ in vertical direction to perform a successful task exception. Using ε as the minimal distance, the required stiffness in vertical direction is then $63/\varepsilon = 7875\,\mathrm{N/m}$.



(a) Objective evaluations carrying (b) Objective evaluations positioning (c) Stiffness curves,

(d) optimized trajectories (e) Fixed vs variable: E_e (f) Fixed vs variable: E_m

Fig. 6: Optimization results.

	b_j $\left[\frac{\text{Nms}}{\text{rad}}\right]$	$\boldsymbol{K}_g$ $\left[\frac{\text{N}}{\text{m}}\right]$	$\boldsymbol{K}_a$ $\left[\frac{\text{N}}{\text{m}}\right]$	w_h [mm]	w_p [mm]
μ	1	diag$(10^3, 10^3, 10^6)$	diag$(0, 0, 300)$	50	40
σ	0.15	diag$(10^2, 10^2, 10^3)$	diag$(0, 0, 15)$	5	5

TABLE II: Parameters in $\boldsymbol{\rho}$. To ensure the possibility of successful task execution, $w_h - w_p$ is at least 10 mm.

hole $\boldsymbol{K}_a$, the width of the hole w_h, and the width of the peg w_p. These are all parameters which are likely to vary in an industrial setting. The parameters with their corresponding mean μ and standard deviation σ are given in Table II. Fig. 7 demonstrates the efficacy of the task-free controller. It can be seen that the open-loop controller achieves successful task execution in nominal conditions ($\boldsymbol{\Sigma} = \boldsymbol{0}$), corresponding to the blue dots in Fig. 7. However, task-free compensation is needed when the real dynamics starts to deviate from the nominal model ($\boldsymbol{\Sigma} \neq \boldsymbol{0}$).
To assess robust performance, we run 100 experiments each with 200 simulations implementing the control law Eq. (11). In each simulation $\boldsymbol{\rho}$ is sampled from its distribution. Over the 100 experiments, we vary the task-free controller gains $k_{TF} \in [1, 8000]$ N/m according to

$$\begin{aligned} \boldsymbol{K}_{TF} &= k_{TF} \cdot \boldsymbol{I}_6 \\ \boldsymbol{B}_{TF} &= 0.1 \cdot k_{TF} \cdot \boldsymbol{I}_6 \,. \end{aligned} \tag{59}$$

We investigate the relation between the success rate, the gains of the task-free controller and the amount of energy which should be added to. The total task-free energy to be stored on top of the nominal (task-based) energy is given as

$$E_{TF} = E_{task} - E_{TB} \tag{60}$$

where E_{task} denotes the total amount of energy to perform the task. E_{TB} denotes the total task based energy calculated using Eq. (50). Fig. 8 shows that when the task-free gains increase, the success rate increases as well. This relation flattens around $k_{TF} \approx 2000$ N/m to a 100% success rate over 200 simulations. Fig. 8b displays the task-free energy (calculated using Eq. (60)) to reach successful insertion. The red positive outer edge of Fig. 8b shows that an increase in success rate requires more task-free energy to complete the task successfully. When the success rate reach 100% then the task-free energy flattens as well. on average, the task-free energy fluctuates around zero. Fig. 8b also displays negative values which indicates that less energy is spent on the task than optimized in the nominal case.

C. Energy tanks, passivity and safety

For the remainder of this paper we use task-free controller gains according to Eq. (59) with $k_{TF} = 4000$ N/m. We use Fig. 8b to store a proper amount of task-free energy in every tank.

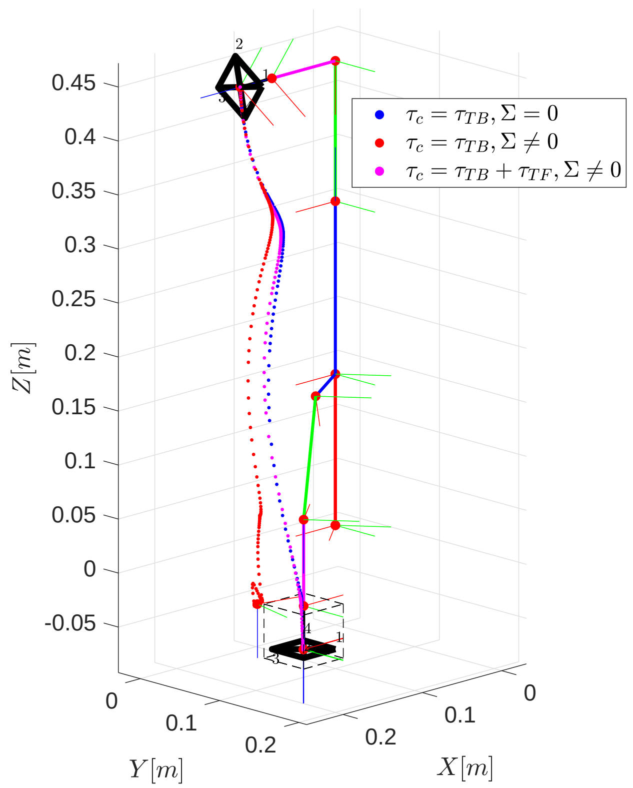
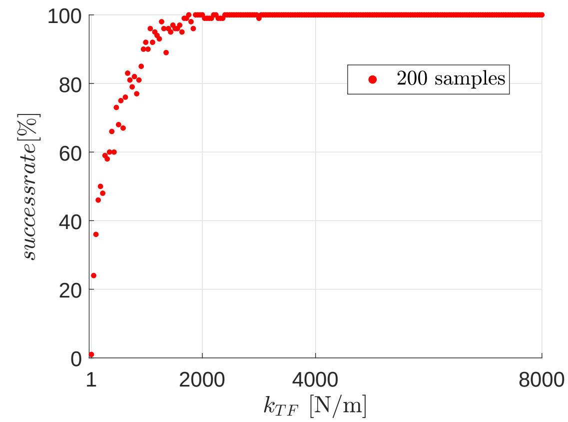
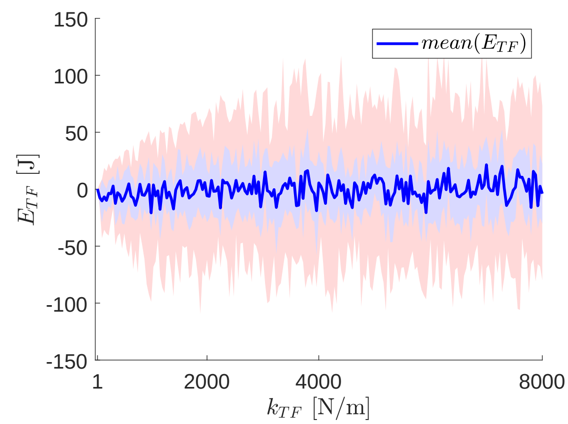


Fig. 7: End-effector trajectories



(a) Success rates as a function of the task-free gains.



(b) Total task-free energy as a function of the task-free gains. For each gain, 200 simulations are performed. The red shaded area represents the outer bounds of the distribution. The blue shaded area shows where 50% of the data is distributed.

Fig. 8: Effect of task-free control gains on success rate and task-free energy

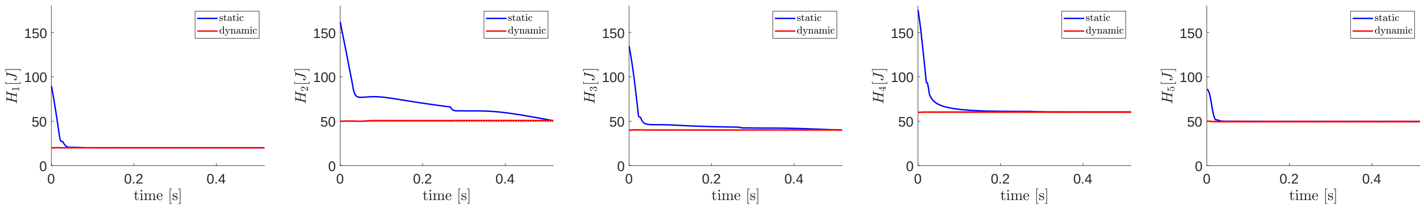
For both static and dynamic energy injections we show simulation results for the case when $\mathbf{\Sigma} = \mathbf{0}$ and $\mathbf{\Sigma} \neq \mathbf{0}$ in Fig. 9. Furthermore, a case in which the robots collides with a damper causing the tanks to empty before task completion is given in Fig. 10. In that case we look at amount of interaction energy transformed from the kinetic energy of the robot for both static and dynamic input strategies.

Some key observations are made:

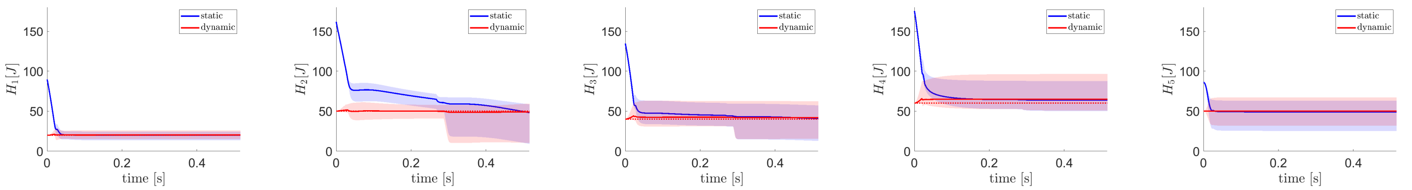
1) Figs. 9f to 9j show that the task-free controller will use some of the task-free energy in the tanks when $\mathbf{\Sigma} \neq \mathbf{0}$. When $\mathbf{\Sigma} = \mathbf{0}$, the control will only use task-based energy, as is evident from Figs. 9a to 9e.
2) Comparing the static and dynamic energy injection strategies it is clear that the energy content at $t = 0$ is smaller for the dynamic energy injection than for the static one.
3) Looking at Figs. 9f to 9j, some of the energy levels increase for $t < 0.1\,s$. This means that the energy used is less than the energy drawn from the tanks. This happens as the torques provided by the task-free controller act in opposite direction to the task-based ones.
4) Looking at the collision case in Fig. 10. When the tanks empty, damping is injected in the joint space according to Eq. (23). As can be seen from Fig. 10, when collision takes place, a part of the kinetic energy (E_k) in being transformed to the exchanged energy. As the applied damping is sufficiently large($\bar{\boldsymbol{B}} = 50 \cdot \boldsymbol{I}_5$) the robot will retain its configuration after the tanks are empty. In addition, the robot exchanges no power with the environment when when kinetic energy goes to zero. Furthermore, it is evident that the tanks with the dynamic input strategy turn off almost immediately after collision. As the energy content for the static input strategy is higher, the tanks take longer to empty resulting in a higher exchanged energy. That is why in terms of safety, the dynamic input strategy has the advantage over the static input type.

D. *Iterative feed-forward adaptation*

In this section we investigate the efficacy of the proposed iterative feedforward scheme posed in Section III-E.

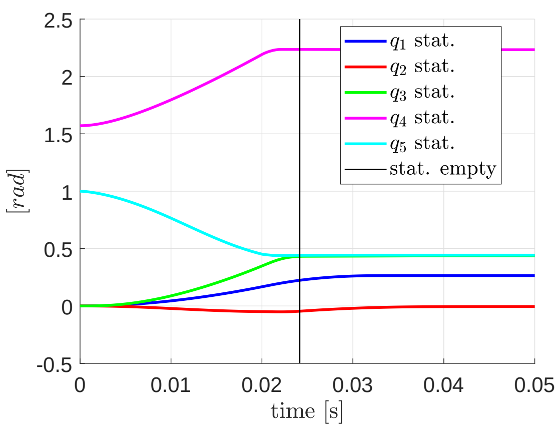


(a) H_1, $\boldsymbol{\Sigma} = \mathbf{0}$ (b) H_2, $\boldsymbol{\Sigma} = \mathbf{0}$ (c) H_3, $\boldsymbol{\Sigma} = \mathbf{0}$ (d) H_4, $\boldsymbol{\Sigma} = \mathbf{0}$ (e) H_5, $\boldsymbol{\Sigma} = \mathbf{0}$

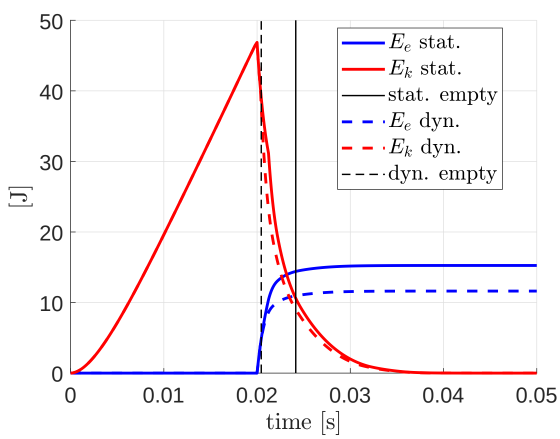


(f) H_1, $\boldsymbol{\Sigma} \neq \mathbf{0}$ (g) H_2, $\boldsymbol{\Sigma} \neq \mathbf{0}$ (h) H_3, $\boldsymbol{\Sigma} \neq \mathbf{0}$ (i) H_4, $\boldsymbol{\Sigma} \neq \mathbf{0}$ (j) H_5, $\boldsymbol{\Sigma} \neq \mathbf{0}$

Fig. 9: Energy levels in the tanks for both dynamic and static input types. Figs (a) to (e) represent the nominal case when $\boldsymbol{\Sigma} = \mathbf{0}$. Figs. (f) to (j) represent the case when $\boldsymbol{\Sigma} \neq \mathbf{0}$ for 100 different $\boldsymbol{\rho}$. The outer bounds of the shaded areas represent the minimum and maximum. The solid lines represent the mean values of all samples. The horizontal dashed line represents the amount of energy stored for the task-free controller.

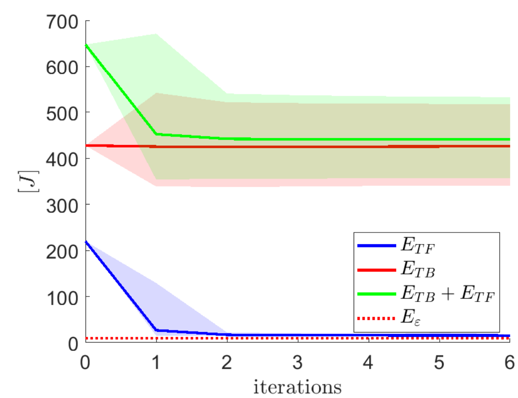


(a) Energy stored in the tanks

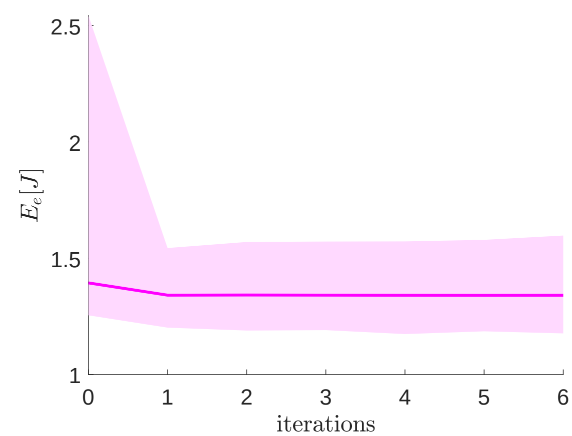


(b) Interaction energy according equation

Fig. 10: Robot collides with Cartesian damper with $\boldsymbol{B_e} = 2 \cdot \boldsymbol{I}_6$ at $t = 0.02\,s$. Stat. and dyn. are short notations for static and dynamic energy input types.

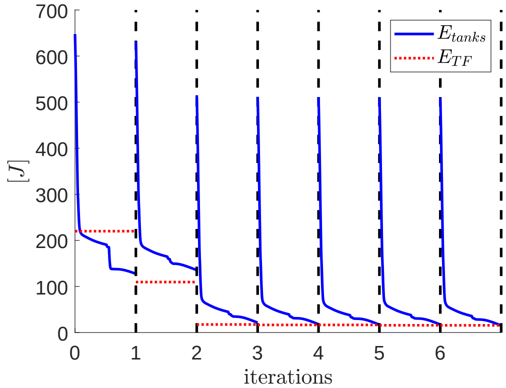


(a) Energy stored in the tanks

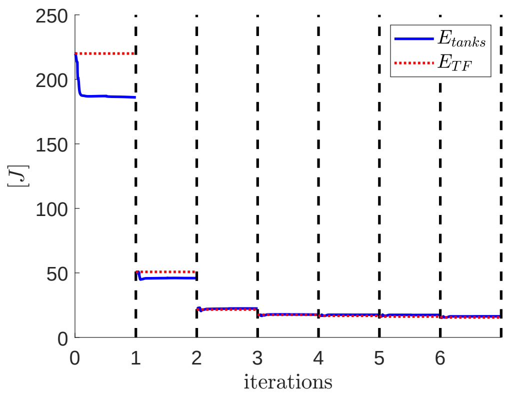


(b) Interaction energy calculated using equation (52)

Fig. 11: Iterative results for the energy in the tanks and the interaction energy. Shaded areas indicate the bounds of 100 simulations. The solid lines represent mean values.



(a) Iterative adaptation with static energy injection



(b) Iterative adaptation with dynamic energy injection

Fig. 12: Iterative feed-forward adaptation for both static and dynamic energy injection, for a single sample of $\boldsymbol{\rho}$. E_{tanks} denotes the energy content in all tanks over time.

Fig. 12 shows results for 7 iterations for 100 simulations where $\boldsymbol{\rho}$ is sampled according to Eq. (58) with $\boldsymbol{\Sigma} \neq \mathbf{0}$ in each simulation. Fig. 11a shows the energy stored in the tanks for every iteration. Noticeable is that the energy allocated to the task-free controller goes to a minimum after 4-5 iterations. The mean of the task-based energy remains relatively constant causing the total amount of energy in the tanks to go to zero. The iterative process for one simulation with a sampled $\boldsymbol{\rho}$ is given in Fig. 12a. It shows the total amount of energy in the tanks (containing both task-free energy and task-based energy) using the static energy injection explained in Section V-C. As a final result we also study the dynamic energy injection[3] given for one simulation in Fig. 12b. For both static and dynamic energy injection types we see that the energy consumption will be more and more task based after each iteration. Especially for the dynamic energy injection in Fig. 12b this is noticeable due to the 'flat' energy levels after $\approx$ 2 iterations. The result is a task-based open-loop control action with minimal feedback compensation for model variations. Combining the iterative adaptation of both task-based control action and the energy tanks with the dynamic energy injection results in a very *safe* controller. This is because the energy content in the tanks is brought to a minimum, causing the tanks to decouple and switch on heavy damping the joint space almost immediately in case of an unexpected event (like the collision in Fig. 10). Furthermore, the fact that the control is now almost completely open-loop removes dangers of feedback destabilization.

[3]Adding the task-based energy dynamically as described in Section III-D. To avoid practical issues, if the output of Eq. (29) becomes negative then $E_{TF}^{i+1} = E_{\varepsilon}$ (if the dynamic energy injection is applied).

VI. Conclusion

In this work, a control strategy is proposed based on time-varying impedance control. The task-based open-loop controller is optimized to a biomimetic metric and was shown to be beneficial over a fixed impedance based on the same metric. To account for model variations from the nominal plant, the task-based controller was supplemented with a task-free Cartesian impedance controller. The gains of this task-free controller were chosen on the basis of the amount of uncertainty, percentage of successful task executions and energetic cost. The information on the energetic cost was used to store the necessary amount of energy in the tanks to recover passivity and achieve successful task execution under model variations. A dynamic energy injection strategy to the tanks was proposed in Section V. This type of energy injection was compared to the more conventional static input type. The dynamic energy injection resulted in lower energy levels in the tanks (especially at $t = 0$). This is considered to be safer due to the fact that the task-free controller is able to use less energy compared to the static input type. As final result, the controller was supplemented with an iterative feed-forward adaptation law for repeated task execution.
In addition, the energy in the tanks was updated to store less task-free energy in every iteration. This adaptation resulted in an open-loop controller with which is task-based, energy aware and accounts for model variations with minimal feedback.

References

[1] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control with virtual stiffness for human-robot cooperative peg-in-hole task. *Proceedings of the 2002 IEEW/RSJ Intl. Conference on Intelligent Robots and Systems*, 2(October):1075–1081, 2002.

[2] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *International Journal of Robotics Research*, 26(1):23–39, 2007.

[3] Chenguang Yang, Gowrishankar Ganesh, Sami Haddadin, Sven Parusel, Alin Albu-Schäeffer, and Etienne Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics*, 27(5):918–930, 2011.

[4] Daniel S. Walker, J. Kenneth Salisbury, and Günter Niemeyer. Demonstrating the benefits of variable impedance to telerobotic task execution. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1348–1353, 2011.

[5] A. Blank, A. M. Okamura, and L. L. Whitcomb. Task-dependent impedance improves user performance with a virtual prosthetic arm. In *2011 IEEE International Conference on Robotics and Automation*, pages 2235–2242, 2011.

[6] Abeje Y. Mersha, Stefano Stramigioli, and Raffaella Carloni. Variable impedance control for aerial interaction. In *IROS 2014 Conference Digest - IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3435–3440. Institute of Electrical and Electronics Engineers Inc., 10 2014.

[7] Freek Stulp, Jonas Buchli, Alice Ellmer, Michael Mistry, Evangelos Theodorou, and Stefan Schaal. Reinforcement Learning of Impedance Control in Stochastic Force Fields. *2011 IEEE International Conference on Development and Learning (ICDL)*, 2:1–6, 2011.

[8] Xinbo Yu, Yanan Li, Shuang Zhang, Chengqian Xue, and Yu Wang. Estimation of human impedance and motion intention for constrained human-robot interaction. *Neurocomputing*, 10 2019.

[9] Etienne Burdet, Gowrishankar Ganesh, Chenguang Yang, and Alin Albu-Schäffer. *Interaction Force, Impedance and Trajectory Adaptation: By Humans, for Robots*, pages 331–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[10] Daniel Leidner, Christoph Borst, Alexander Dietrich, Michael Beetz, and Alin Albu-Schaffer. Classifying compliant manipulation tasks for automated planning in robotics. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:1769–1776, 2015.

[11] Igor Shardyko and Victor Titov. Implementation of impedance trajectory control on a 6-DoF manipulator. *13th International Scientific-Technical Conference on Electromechanics and Robotics "Zavalishin's Readings"*, 03007:4–8, 2018.

[12] Kevin Haninger and Dragoljub Surdilovic. Multimodal Environment Dynamics for Interactive Robots: Towards Fault Detection and Task Monitoring. *IEEE International Conference on Intelligent Robots and Systems*, pages 6932–6937, 2018.

[13] Rishi K. Malhan, Yash Shahapurkar, Ariyan M. Kabir, Brual Shah, and Satyandra K. Gupta. Integrating impedance control and learning based search scheme for robotic assemblies under uncertainty. *ASME 2018 13th International Manufacturing Science and Engineering Conference, MSEC 2018*, 3:1–10, 2018.

[14] Kevin Haninger, Arturo Bastidas-cruz, and Dragoljub Surdilovic. What dynamics should impedance-controlled robots render ?

[15] Stefano Stramigioli. *Lecture notes in control and information sciences*. Springer London, 1998.

[16] Gennaro Raiola, Carlos Alberto Cardenas, Tadele Shiferaw Tadele, Theo De Vries, and Stefano Stramigioli. Development of a Safety- and Energy-Aware Impedance Controller for Collaborative Robots. *IEEE Robotics and Automation Letters*, 3(2):1237–1244, 2018.

[17] Stefano Stramigioli and Herman Bruyninckx. Geometry and Screw Theory for Robotics. *Matrix*, 2001.

[18] Etienne Burdet, Rieko Osu, David W. Franklin, Theodore E. Milner, and Mitsuo Kawato. The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, 414(6862):446–449, 2001.

[19] M. K. Camlibel, L. Iannelli, and F. Vasca. Passivity and complementarity. *Mathematical Programming*, 145(1-2):531–563, 6 2014.

[20] Gerrit A Folkertsma and Stefano Stramigioli. Energy in Robotics. *Foundations and Trends in Robotics*, 6(3):140–210, 2017.

[21] Johannes Lachner, Vincenzo Schettino, Felix Allmendinger, Mario Daniele Fiore, Fanny Ficuciello, Bruno Siciliano, and Stefano Stramigioli. The influence of coordinates in robotic manipulability analysis. *Mechanism and Machine Theory*, 146(688188), 2020.

[22] Shiferaw Tadele Tadele. *Human-friendly Robotic Manipulators*. Phd, University of Twente, Enschede, 2015.

[23] Arne Lakså, Børre Bang, and Lubomir T. Dechevsky. GEOMETRIC MODELLING WITH BETA-FUNCTION B-SPLINES, I: PARAMETRIC CURVES. *Pure and Applied Mathematics*, 65(3):37–44, 2010.

Chapter 4

Reflection

This chapter reflects on the overall work. It concludes about the chosen control strategy and the results on the basis of the research questions. Conclusions are made specifically for the proof of concept in Section 2.5 and specifically for the multi DoF case in Section VI in the paper. In addition, we look at possible improvements and future work and reflect on the contribution of this work. A few points of discussion are made in the last section in this chapter.

4.1 Conclusion

Research questions

Six research questions were posed in Section 1.3.2. To tackle these questions, two cases were presented. Chapter 2 introduced a proof of concept in which the main ideas of the thesis were exploited on a 1-DoF mass subjected to a nonlinear friction profile. The ideas from the proof of concept were studied on a multi DoF situation which concerned a Cartesian peg-in-hole task. In the following, we will reflect on the individual research questions in light of the the results and the overall approach:

1. *How does the desired time-varying impedance relates to a control architecture comprising feed-forward and feedback?*
   Both cases (proof of concept and multi DoF case) comprised the proposition of a control strategy based on time-varying impedance control. The time-varying impedance resulted in torque profiles which were applied in an open-loop way labelled as task-based control action. Everything else, which was unforeseen by the task was labelled as 'task-free' and was implemented as a feedback controller.

2. *What is the desired time-varying impedance for a peg-in-hole task in a (semi)-structured environment based on the task definition and a task-based metric?*
   To find the variable impedance, a constraint optimization problem was set-up which depended on the definition of the task and a task-based metric using the knowledge of the dynamics of the robot and environment. In the optimization, the impedance values (characterized by a spring), were optimized using finite dimensional parametrization with splines. In every objective evaluation the dynamics were simulated which ensured that the dynamics and minimization problem were solved sequentially. This allowed that the problem entered as

continuous time problem as time discretization was done by the `ode` solver itself.
The objective functions used in the optimization showed to be non convex meaning that there are multiple local minima. That is why the choice of the weights of the objective function as well as the initial guess for spline parameters heavily influenced the final result. In an attempt in making the system 'critically-damped', a diagonal upper bound inertia matrix was defined in the multi DoF case. It left a straightforward implementation which was only valid in non-singular configurations for non-underactuated robots. The resulting damper scaled only with the stiffness and mass and the end-effector. Because of this choice, the damper could be left behind in the optimization which reduced the solution space in the optimization.
For both the proof of concept and the multi DoF case, the chosen objective was based on a biomimetic metric in an attempt to copy the exceptional performance of humans as decision makers for varying their impedance on the basis of a task (refer to Section 1.2.4). This metric was formulated into a mathematical objective of scalar type which made the trade-off between performance and metabolic cost.
For the proof of concept, the optimal stiffness used the friction to de-accelerate by becoming compliant resulting in lower metabolic cost. For the multi DoF case, the task was split up into a carrying subtask and insertion subtask. In the carrying subtask, minimization in terms of metabolic cost was obtained by becoming compliant. In the insertion subtask only the vertical stiffness needed to be increased which caused the planar and rotational stiffness to become compliant.

3. *What can you say about the robustness of the optimized impedance profiles under model variations?*
   For the multi DoF situation we investigated the robustness of the optimized impedance profiles under model variations. It was shown that task-free feedback compensation was needed to perform successful task execution when the parameters in the robot dynamics and environment started to vary. The gains of this task-free controller were chosen on the basis of the amount of uncertainty, percentage of successful task executions and energetic cost. The information on the energetic cost was used to store the necessary amount of energy in the tanks to recover passivity and achieve successful task execution under model variations.

4. *How to utilize the energy tanks in view of passivity and safety?*
   To ensure passivity of the controlled system, the energy tank concept was used. The information on the energetic cost was utilized to store the necessary amount of energy in the tanks to recover passivity and achieve successful task execution under model variations. In contrast to storing all the energy in the tanks at $t = 0$ (referred to as static input strategy), the information on the task-based energy consumption provided a way to dynamically store the energy in the tanks. This dynamic energy injection strategy was *safer* than the static input strategy as the overall energy content in the tanks was lower (especially at $t = 0$). Because of this lower energy content, the controller was decoupled earlier than for the static input strategy. In the multi DoF case, when one of the tanks emptied, the system was turned into a heavily damped system which is by definition a passive action.

5. *Can the results from the optimization somehow be improved with the aid of data?*
   In a final stage in the multi DoF situation, the task-based control action was updated to such a degree that it was able to adapt for model variations. This data-driven adaptation law resulted

in an open-loop control action which was task-based and energy-aware due to the optimization while accounting for model variations with minimal feedback. Since the feedback action was diminished and the control became more and more task-based, the energy in the tanks was updated accordingly. This had the effect that also the energy tank systems became more task-based. Combined with the dynamic input strategy resulted in extremely low energy contents in the tanks which increased safety (reasoned in 4.).

6. *How do the fixed impedance, task-driven impedance and data-driven impedance compare?* For both the proof of concept as well as the multi DoF case, the optimal time-varying impedance showed to be beneficial over an optimal fixed impedance on the basis of the chosen metric in nominal conditions. For the proof of concept, the variable impedance resulted in a better trade-off between metabolic cost and position error compared to an array of fixed impedances. For the multi DoF situation, the variable impedance resulted in lower exchanged energy with the environment and lower metabolic cost. The task-based control action and corresponding energy allocation in the tanks were updated using a data-driven protocol (refer to 5.). This result was a lot safer (reasoned in 4.) due to limited energy in the tanks compared to the conventional use of the tanks where all the energy is stored initially. In addition, comparing the results from the data-driven adaptation law at iteration 0 (which is just the result from the optimization) to the control action in iteration 6, it was shown that the data-driven approach after 6 iterations resulted in lower exchanged energy with the environment and completed the task with minimal feedback. In terms of speed and stability, performing the task using only open-loop control results in a faster response and is less prone to stability issues compared to a high-gain feedback controller.

Future work

Three recommendations regarding future work are made:

1. In this work, we used the Cartesian impedance control framework and optimized the geometric spring in time. The expression for the wrench exerted on the end-effector induced by the geometrical spring used to define the impedance law is distilled from the cubic potential the (linear) Cartesian spring defines. However, as the Cartesian spring is virtual, one is not restricted to the cubic form of the potential. Therefore, as a future recommendation it is advised to optimize the potential directly in the robot's configuration space. This will directly result in a feedback controller and it also known as *potential energy shaping* [26]. A major challenge is to define a new expression for the wrench exerted on the end-effector induced by the potential. Furthermore, optimizing the potential directly on a functional space requires a new type of optimization than what was used in this thesis. A possible solution may lie in the use of neural networks. They are commonly known as 'universal function approximates' and might provide a better solution than discretizing the problem using splines.

2. A second recommendation concerns the embedding of 'sensing' in the impedance choice. To clarify, an example can be given following the human analogy: if a human has to perform a task in the dark (when there is no vision), the human will be compliant to ensure he will not damage anything. This can be cast into an optimization framework to choose the impedance not only task-based but also on the basis of the amount of knowledge or structuredness of

the environment. This can be similar to a Kalman-filter[1] only then applied to impedance and interaction.

3. A final recommendation concerns safety . In this work, safety was increased by minimizing the energy content in the tanks to perform successful task execution. However, safety has not been assessed in terms maximum power or energy transfer. Therefore, to make a human-friendly extension to the control strategy, safety regarding exchanged power and energy can be embedded in the optimization. Commonly used criteria used in the automotive industry are the Head Impact Power [32] and the Head Injury Criteria [33]. These concepts have already been used as a safety layer utilizing modulated impedance in [7] and can might be utilized to make the human-friendly extension to this work.

Contribution

This work contains three contributions, already listed in Section 1.4 and are being looked at here from a more conclusive point of view:

1. The first contribution relates to the presentation of the control strategy where a distinction has been made between from what is task-based and what is task-free. The control strategy provides an overall framework to implement a task-based time-varying impedance controller and how to deal with model variations. One of the major differences with the existing literature is that the time-varying impedance is optimized a priori according to a task and that the resulting torques from the impedance framework enter as an open-loop control action rather than a closed-loop controller. The main advantage of performing open-loop control instead of high-gain feedback control is that it has a faster dynamic response and it less prone to stability issues due to time delays. The methodology has been applied to a single mass and a 5-DoF robot. However, the same approach and thinking process in defining a task-based impedance can be applied to other robots in (semi-)structured environments as well.

2. The second contribution relates to the use of energy tanks. Energy tanks were initially introduced to recover passivity. The result from the optimization was used to store the energy tanks only with the necessary amount of energy to complete the task successfully. On top of that, a dynamic energy injection was proposed which provided the necessary amount of energy over time resulting in lower energy contents in the tanks. A lower energy content increases safety since the controller will decouple sooner after which a large amount of damping in the joints was applied (resulting in lower exchanged energy in case of collision).

3. A final contribution relates to the iterative feed-forward law. The iterative adaptation of the controller and the corresponding energy allocation for the tanks resulted in successful task execution by using open-loop control with minimal feedback. As the information of the model uncertainties were enclosed in the task-based control action, the energy content in the tanks could be minimized as well resulting in an even safer task execution.

[1] Used in signal processing and data-fitting. The Kalman gain matrix describes a trade-off between the measurement and the state estimate.

4.2 Discussion

Two points of discussion are made:

1. The optimization problems posed in this work were solved using MATLAB 2020a using `fmincon` with the `interior-point` algorithm. However, the project scope was not focused on optimization alone. Therefore, the quality of the optimization has not been assessed. In addition, no account has been taken on how the optimization problems were posed with regard to convergence. Including the state-variables in the optimization as solution variables (and providing the dynamics as a constraint) might increase convergence as it increases the overall flexibility of the problem. Solving the problem with direct collocation in a symbolic framework like CasADi which implements algorithmic differentiation to calculate the gradient and IPOPT as a large scale solver, might increase converge of the problem as well [34]. However, the main problem with optimizing interactions is that the dynamics often switch (robot can move in free space and robot interacts). Switching dynamics results in nonlinearities in the objective which makes it difficult to get a proper convergence as the objective is not continuously differentiable.

2. A second point of discussion relates to the proposition of the dynamic input strategy for the energy tanks. The strategy was presented as a safety extension of the energy tanks. However, as the dynamic input strategy requires a power source at the side of the tank, the system might not be theoretically passive any more. Obviously, everything can be made passive with an energy so introducing another tank which provides the dynamic input might resolve this issue. However, the main point of discussion here is that starting from 'passivity as must' does not guarantee safety. Increasing safety resulted in destruction of the passivity criterion. If one wants to recover passivity again, safety might be decreased due to higher energy contents. This results in a circular reasoning which induces the question whether starting from 'passivity as must' needs to be relaxed.

Bibliography

[1] Neville Hogan. Impedance Control: An Approach to Manipulation: Part I—Theory. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):1–7, 03 1985.

[2] Neville Hogan. Impedance Control: An Approach to Manipulation: Part II—Implementation. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):8–16, 03 1985.

[3] Neville Hogan. Impedance Control: An Approach to Manipulation: Part III—Applications. *Journal of Dynamic Systems, Measurement, and Control*, 107(1):17–24, 03 1985.

[4] Gjerrit Meinsma. *Nine Lessons on Control Systems*. University of Twente, 2019. notes for UTwente course ”Robust Control”.

[5] Neville (Massachusetts Institute of Technology) Hogan and Stephen P. (Massachusetts Institute of Technology) Buerger. Interaction Control. In Thomas R Kurfess, editor, *Robotics and Automation Handbook*, chapter 19, pages 121–154. CRC PRESS, 2004.

[6] Stefano Stramigioli. Interative control. Lecture slides from course ’Modern Robotics’ lecture 6, 2019.

[7] Gennaro Raiola, Carlos Alberto Cardenas, Tadele Shiferaw Tadele, Theo De Vries, and Stefano Stramigioli. Development of a Safety- and Energy-Aware Impedance Controller for Collaborative Robots. *IEEE Robotics and Automation Letters*, 3(2):1237–1244, 2018.

[8] Stefano Stramigioli and Herman Bruyninckx. Geometry and Screw Theory for Robotics. *Matrix*, 2001.

[9] Stefano Stramigioli. *Lecture notes in control and information sciences*. Springer London, 1998.

[10] D. A. Lawrence. Impedance control stability properties in common implementations. In *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, pages 1185–1190 vol.2, 1988.

[11] Toru Tsumugiwa, Ryuichi Yokogawa, and Kei Hara. Variable impedance control with virtual stiffness for human-robot cooperative peg-in-hole task. *Proceedings of the 2002 IEEW/RSJ Intl. Conference on Intelligent Robots and Systems*, 2(October):1075–1081, 2002.

[12] Igor Shardyko and Victor Titov. Implementation of impedance trajectory control on a 6-DoF manipulator. *13th International Scientific-Technical Conference on Electromechanics and Robotics “Zavalishin’s Readings”*, 03007:4–8, 2018.

[13] Alin Albu-Schäffer, Christian Ott, and Gerd Hirzinger. A unified passivity-based control framework for position, torque and impedance control of flexible joint robots. *International Journal of Robotics Research*, 26(1):23–39, 2007.

[14] Kevin Haninger and Dragoljub Surdilovic. Multimodal Environment Dynamics for Interactive Robots: Towards Fault Detection and Task Monitoring. *IEEE International Conference on Intelligent Robots and Systems*, pages 6932–6937, 2018.

[15] Chenguang Yang, Gowrishankar Ganesh, Sami Haddadin, Sven Parusel, Alin Albu-Schäeffer, and Etienne Burdet. Human-like adaptation of force and impedance in stable and unstable interactions. *IEEE Transactions on Robotics*, 27(5):918–930, 2011.

[16] Daniel S. Walker, J. Kenneth Salisbury, and Günter Niemeyer. Demonstrating the benefits of variable impedance to telerobotic task execution. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1348–1353, 2011.

[17] A. Blank, A. M. Okamura, and L. L. Whitcomb. Task-dependent impedance improves user performance with a virtual prosthetic arm. In *2011 IEEE International Conference on Robotics and Automation*, pages 2235–2242, 2011.

[18] Rishi K. Malhan, Yash Shahapurkar, Ariyan M. Kabir, Brual Shah, and Satyandra K. Gupta. Integrating impedance control and learning based search scheme for robotic assemblies under uncertainty. *ASME 2018 13th International Manufacturing Science and Engineering Conference, MSEC 2018*, 3:1–10, 2018.

[19] Abeje Y. Mersha, Stefano Stramigioli, and Raffaella Carloni. Variable impedance control for aerial interaction. In *IROS 2014 Conference Digest - IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3435–3440. Institute of Electrical and Electronics Engineers Inc., 10 2014.

[20] Freek Stulp, Jonas Buchli, Alice Ellmer, Michael Mistry, Evangelos Theodorou, and Stefan Schaal. Reinforcement Learning of Impedance Control in Stochastic Force Fields. *2011 IEEE International Conference on Development and Learning (ICDL)*, 2:1–6, 2011.

[21] Xinbo Yu, Yanan Li, Shuang Zhang, Chengqian Xue, and Yu Wang. Estimation of human impedance and motion intention for constrained human-robot interaction. *Neurocomputing*, 10 2019.

[22] Etienne Burdet, Gowrishankar Ganesh, Chenguang Yang, and Alin Albu-Schäffer. *Interaction Force, Impedance and Trajectory Adaptation: By Humans, for Robots*, pages 331–345. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[23] Daniel Leidner, Christoph Borst, Alexander Dietrich, Michael Beetz, and Alin Albu-Schaffer. Classifying compliant manipulation tasks for automated planning in robotics. *IEEE International Conference on Intelligent Robots and Systems*, 2015-Decem:1769–1776, 2015.

[24] Etienne Burdet, Rieko Osu, David W. Franklin, Theodore E. Milner, and Mitsuo Kawato. The central nervous system stabilizes unstable dynamics by learning optimal impedance. *Nature*, 414(6862):446–449, 2001.

[25] M. K. Camlibel, L. Iannelli, and F. Vasca. Passivity and complementarity. *Mathematical Programming*, 145(1-2):531–563, 6 2014.

[26] Gerrit A Folkertsma and Stefano Stramigioli. Energy in Robotics. *Foundations and Trends in Robotics*, 6(3):140–210, 2017.

[27] Federica Ferraguti, Cristian Secchi, and Cesare Fantuzzi. A tank-based approach to impedance control with variable stiffness. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4948–4953, 2013.

[28] Tadele Shiferaw Tadele, Theo J.A. De Vries, and Stefano Stramigioli. Combining energy and power based safety metrics in controller design for domestic robots. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 1209–1214, 2014.

[29] Kevin Haninger, Arturo Bastidas-cruz, and Dragoljub Surdilovic. What dynamics should impedance-controlled robots render ?

[30] D Guida, F Nilvetti, and C M Pappalardo. Dry Friction of Bearings on Dynamics and Control of an Inverted Pendulum. *Journal of Achievements in Materials and Manufacturing Engineering*, 38(1):80–94, 2010.

[31] Arne Morten Kvarving. Natural cubic splines. *Stata Journal*, 4(22), 1995.

[32] J. Newman, N. Shewchenko, and E. Welbourne. A proposed new biomechanical head injury assessment function - the maximum power index. *Stapp car crash journal*, 44:215–47, 2000.

[33] Sami Haddadin, Alin Albu-Schäeffer, and Gerd Hirzinger. Safety analysis for a human-friendly manipulator. *International Journal of Social Robotics*, 2:235–252, 09 2010.

[34] Joel A E Andersson, Joris Gillis, Greg Horn, James B Rawlings, and Moritz Diehl. CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[35] Wolfgnag Borutzky. *Modeling and simulation of digital systems using bond graphs*. Springer, 2013.

[36] J B C Engelen and G A Folkertsma. *Control for UAVs*. University of Twente, 2019. notes for UTwente course "Control for UAVs".

[37] Stefano Stramigioli. Dynamics, how do robot move when we apply forces. Lecture slides from course 'Modern Robotics' lecture 4, 2019.

[38] Magnus Bjerkeng and K.Y. Pettersen. A new coriolis matrix factorization. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 4974–4979, 05 2012.

[39] Johannes Lachner, Vincenzo Schettino, Felix Allmendinger, Mario Daniele Fiore, Fanny Ficuciello, Bruno Siciliano, and Stefano Stramigioli. The influence of coordinates in robotic manipulability analysis. *Mechanism and Machine Theory*, 146(688188), 2020.

[40] Neville Hogan. Impedance control of industrial robots. *Robotics and Computer Integrated Manufacturing*, 1(1):97–113, 1984.

Appendices

Appendix A

Bond graphs

This chapter will explain the fundamentals of bond graphs. However, for a more complete guide, please refer to [35].

A.1 Bonds and power ports

The fundamentals on bond graphs rely on connecting standard elements through 'bonds'. Each bond represents a power exchange and is connected between 'power ports'. The direction of the bond represent in which way the power flow is positively defined. An example is given in the following figure where there is one bond connected between two power ports.

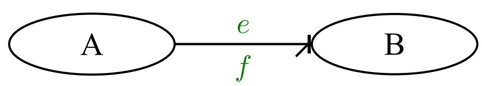


Figure A.1: System with two elements, A and B. Positive power flow is in the direction of B. This means that if the power is positive, power flows from A to B. A and B both have 1 power port. As A and B are connected, they form a new system call it C. The vertical bar indicates that A has effort-out causality.

Talking about 'power' is useful because it does not belong to one domain specifically. It means that different physical domains (like mechanical, electrical, hydraulic etc.) can be connected on the basis of a power exchange. A simple example can be to model a mechanical system driven by an electrical motor. Moreover, the idea of power can be extended to a set of basic elements from which more complex systems can be modelled. More on this in the next section.
In it's most general form, power is a duality pairing between 'effort' and 'flow'. They are mathematical members of dual vector spaces. Effort and flow take different physical aspects dependent on the domain. In the mechanical domain for example, they represent 'force' and 'velocity' whereas in the electrical domain they represent 'voltage' and 'current'. For all domains hold that the duality between effort and flow equals the power:

$$P = e^{\top} f \,. \tag{A.1}$$

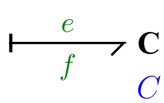
Regarding the modelling language, it is necessary to understand in which direction effort and flow are positively defined. For a mechanical mass for example, one wants to know in which direction the force is exerted independent on the direction of the power. This is solved with a vertical bar notation on the bond. An example is again given in figure A.1. The vertical bar denotes that the effort is positively defined in the direction to system B. More formally, A has ’effort-out causality’.

A.2 Standard elements

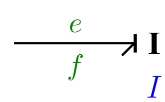
As being said in the latter section, there is an analogy on the basis of power between physical systems with different domains. This idea is extended to a list of standard elements from which more complex systems can be modelled. This list of basic elements results from different ’types’ or ’groups’ of elements. These types together with the corresponding element(s) are elaborated in the following. For each element the corresponding ’constitutive-relation’ is given. This relation tells how the element outputs an effort or flow based on the input. In addition, for each element a corresponding mechanical and electrical example is given.

Storage elements

Storage elements can store potential energy or kinetic energy. Potential energy is stored in **C**-type elements . Kinetic energy is stored **I**-type elements. The following figure represent the bond graph representation of these elements. Notice the causality for each element, this is their ’preferred/integral’-causality. Differential causality is not desired as it requires future information.



(a) Effort out causality, $e = e_0 + \frac{1}{C}\int f dt$
mechanical: spring, electrical: capacitor



(b) Effort in causality, $f = f_0 + \frac{1}{I}\int e dt$
mechanical: inertia, electrical: self inductance

Figure A.2: Storage elements

Dissipative elements

Dissipative elements turn energy into thermal energy. They do not have a prefered causaility as they cannot store energy. The bond graph representation is given below.

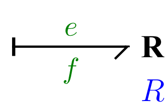
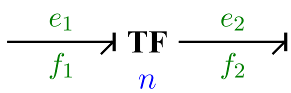


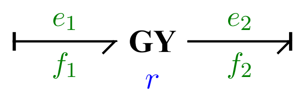
Figure A.3: $e = Rf$
mechanical: damper, electrical: resistor

Two port elements

Two ports elements have two power ports. They are mostly used to transform the energy from one domain to another. Therefore they have two power ports, one for each domain. Most regularly used two port elements are the transformer and gyrator (there are more two-port elements, but they will not be elaborated here). The **TF**-element transforms flow to flow and effort to effort whereas a **GY**-element transforms flow to effort and effort to flow.



(a) $e_2 = ne_1$, $f_1 = nf_2$
mechanical: gear box, electrical: electrical transformer

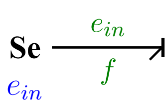


(b) $e_2 = rf_1$, $f_2 = re_1$
mechanical: gyroscope, electrical: electrical gyrator

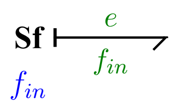
Figure A.4: Two port elements

Power supply elements

Power supply elements are able to supply energy to the system. There are two possibilities. Either with a **Se**-element which sets an effort. Or by setting a flow with a **Sf**-element.



(a) $e = e_{in}$
mechanical: force, electrical: voltage



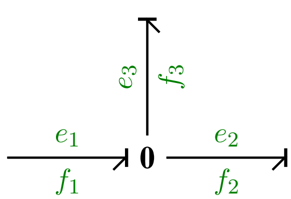
(b) $f = f_{in}$
mechanical: velocity, electrical: current

Figure A.5: Power supply elements

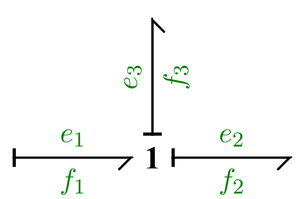
Junctions

Junctions are able to distribute energy over other elements. There are two types. The **1**-junction shares the flow between connecting bonds. Due to power continuity, the effort is than distributed over the bonds. Power continuity is that the incoming power equals outgoing power, in the following junction examples hold:

$$P_1 = P_2 + P_3 \,. \tag{A.2}$$



(a) $e_1 = e_2 = e_3$, $f_1 = f_1 + f_3$



(b) $f_1 = f_2 = f_3$, $e_1 = e_2 + e_3$

Figure A.6: Junctions

Example: Bongraph modelling of second order systems

The equation of motion for a standard mass-spring-damper system can be written as,

$$m\ddot{x} + b\dot{x} + kx = F_{in} \tag{A.3}$$

where m is the mass, b the damper coefficient and k the stiffness of the spring. x is the position of the mass and F_{in} represents all forces acting on this mass. Now consider the equation of motion for a standard RLC-circuit with resistance R_e, self inductance L and capacitance C_e. Also consider an input voltage U_{in} and electrical charge q. The equation of motion can be written as

$$L\ddot{q} + R_e\dot{q} + \frac{1}{C_e}q = U_{in}\,. \tag{A.4}$$

The similarity between equation A.3 and A.4 is obvious. Both are second order differential equations but are written in different physical domains. Using bond graphs, both equations can be represented by the same model.

The bond graph model for representing A.3 and/or A.4 is given in figure A.7. Each bond represents a power flow according to equation A.1. What can be seen is that all elements are connected to a **1**-junction. This is because all elements share the same flow which the junction represents.

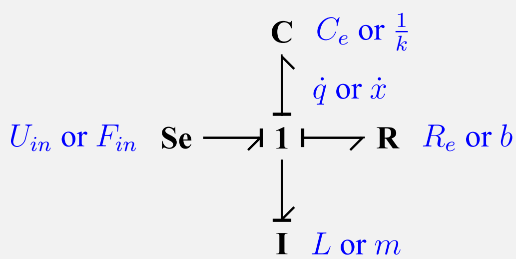


Figure A.7: Bond graph representation of equation A.4 and/or A.3

Appendix B

Screw theory

B.1 Twists and wrenches

Screw theory is based on the Lie algebra of rigid body motion in $SE(3)$ [8]. [8] is used as the main reference in this chapter. The Lie algebra is the space tangent to the identity of a Lie group $SE(3)$. The lie algebra carries interesting mathematical properties which can be used to model multi-body systems, like robots.
First consider the configuration of a body i with respect to a body j like in Fig. B.1 where Ψ_i denotes the Cartesian frame i. The configuration is a combination of an orientation and a position. The orientation is indicated by $\boldsymbol{R}_i^j \in SO(3)$ which is the rotation matrix from body i with respect to body j. The position is indicated by $\boldsymbol{p}_i^{j,j} \in \mathbb{R}^3$ which is the position from body i with respect to body j expressed in Ψ_j.

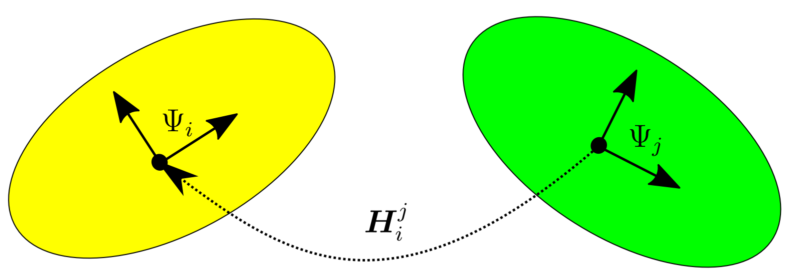


Figure B.1: Two bodies i and j together with their body-fixed frames.

The configuration matrix, $\boldsymbol{H}_i^j$ is written as

$$\boldsymbol{H}_i^j = \begin{pmatrix} \boldsymbol{R}_i^j & \boldsymbol{p}_i^{j,j} \\ \boldsymbol{0} & 1 \end{pmatrix} . \tag{B.1}$$

The configuration matrix is a continuous and smooth group in $SE(3)$. This also means that it is continously differentiable in time. The right sided and left sided transformation are members of

$se(3)$ written respectively as

$$\tilde{\boldsymbol{T}}_i^{j,j} = \dot{\boldsymbol{H}}_i^j \boldsymbol{H}_j^i \quad \text{(B.2)}$$
$$\tilde{\boldsymbol{T}}_i^{i,j} = \boldsymbol{H}_j^i \dot{\boldsymbol{H}}_i^j \quad \text{(B.3)}$$

where $\boldsymbol{T}_i^{i,j}$ is also known as the twist of body i with respect to body j expressed in Ψ_i. The convenience comes with the fact that the members of $se(3)$ are always written in a standard form,

$$\tilde{\boldsymbol{T}}_i^{j,j} = \begin{pmatrix} \tilde{\boldsymbol{\omega}}_i^{j,j} & \boldsymbol{v}_i^{j,j} \\ \boldsymbol{0} & 0 \end{pmatrix} \quad \text{(B.4)}$$

where $\tilde{\boldsymbol{\omega}}_i^{j,j} \in so(3)$ and $\boldsymbol{v}_i^{j,j} \in \mathbb{R}^3$. The $(\tilde{\bullet})$ operator is described in appendix D.2. The twist itself can now be written as,

$$\boldsymbol{T}_i^{j,j} = \left[\begin{array}{cc} \boldsymbol{\omega}_i^{j,j} & \boldsymbol{v}_i^{j,j} \end{array} \right] \quad \text{(B.5)}$$

The twist is a generalized velocity that means that $\boldsymbol{\omega}_i^{j,j}$ represents the rotational velocity and $\boldsymbol{v}_i^{j,j}$ represents the velocity of an imaginary point passing through the origin of Ψ_j [8]. Together then can be visualized like a *screw*. The dual of twists are known as wrenches. In this work, only the wrench on the end-effector is of interest. Therefore, we only write the frame in which the wrench is expressed. Therefore, consider a wrench expressed in Ψ_j to be written as $\boldsymbol{W}^j$. As they are dual of members in $se(3)$, wrenches are members of $se^*(3)$ and written as,

$$(\boldsymbol{W}^j)^\top = \left[\boldsymbol{\tau}^j, \boldsymbol{f}^j\right]^\top . \quad \text{(B.6)}$$

where $\boldsymbol{\tau}^j \in \mathbb{R}^3$ is the rotational force (or torques) on body j and $\boldsymbol{f}^j \in \mathbb{R}^3$ is the linear force on body j. The usefulness of using screw theory in terms of modelling becomes clear in the next section.

B.2 Rigid body modelling

In this work a model of a 5-DoF robotic arm is made. In order to this, screw theory and bond graphs can be used as a modelling language. Furthermore, one can also write the Euler Lagrange equations directly without the use of bond graphs. Both methods will be explained in this section.

B.2.1 Multi body dynamics bond graphs

For modelling rigid bodies using screw theory and bond graphs, we use the theory used for UAV modelling and control described in [36]. Assume an inertial frame, Ψ_0 and a frame fixed to body i, Ψ_i. The equation of motion for rigid body i can then be written as,

$$\boldsymbol{I}^i \dot{\boldsymbol{T}}_i^{i,0} = -\operatorname{ad}^T_{\boldsymbol{T}_i^{i,0}} (\boldsymbol{\mathcal{P}}^i)^\top + (\boldsymbol{W}^i)^\top \quad \text{(B.7)}$$

where $\boldsymbol{I}^i \in \mathbb{R}_+^{6\times 6}$ is the inertia matrix from body i. $\boldsymbol{\mathcal{P}}^i$ is the momentum screw is called the momentum screw equal to

$$(\boldsymbol{\mathcal{P}}^i)^\top = \boldsymbol{I}^i \boldsymbol{T}_i^{i,0} . \quad \text{(B.8)}$$

If Ψ_i is chosen in the center of gravity, the inertia matrix becomes diagonal and can be written as,

$$\boldsymbol{I}^i = \begin{pmatrix} \mathrm{diag}({J_x}^i, {J_y}^i, {J_z}^i) & \boldsymbol{0} \\ \boldsymbol{0} & m^i \boldsymbol{I}_3 \end{pmatrix} \tag{B.9}$$

where J_α, $\alpha = \{x, y, z\}$ is the moment of inertia around α. m^i is the linear mass of body i. $\boldsymbol{T}_i^{i,0}$ refers to the twist of body i with respect to Ψ_0, expressed in Ψ_i. The first right hand side term in equation (B.7) represents the gyroscopic effects. $\boldsymbol{W}^i$ is the wrench acting on body i. The general rigid body bond graph representation which corresponds to equation (B.7) can be found in figure B.2.

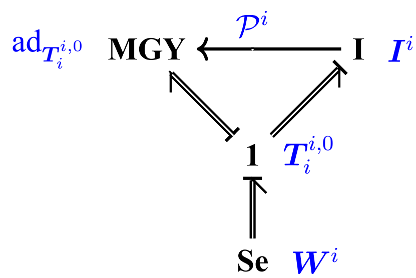


Figure B.2: Bond graph representation of a rigid body

Connecting rigid-bodies is done by means of joints. A useful property resulting from screw theory is that two twists on a rigid body are equal if they are expressed in the same frame. To clarify, assume that Ψ_i and Ψ_j are frames on the same rigid body then,

$$\boldsymbol{T}_i^{0,0} = \boldsymbol{T}_j^{0,0} \tag{B.10}$$

This allows to constraint two rigid bodies by comparing their twists in the same frame without transforming to a joint location. The joint itself is than modelled as a series of springs and damper in the constraint direction. An example: consider two rigid bodies. In order to compare the two twists, one has to make the transformation to the same frame (an inertial frame is the easiest). Then, the twist difference $\boldsymbol{T}_i^{0,j}$ is transformed to frame in the joint Ψ_l using

$$\boldsymbol{H}_0^l = \boldsymbol{H}_i^l (\boldsymbol{H}_i^0)^{-1}\,. \tag{B.11}$$

In Ψ_l a series of springs and dampers are applied in the constraint direction. The free DOF is obviously not constraint by the damper and spring. This translates to the bond graph representation in figure B.3.

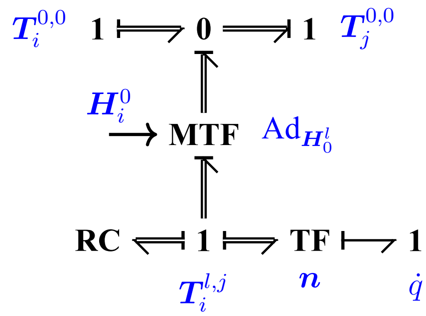


Figure B.3: bond graph representation of a joint

Figure B.4: Constraint between body i and body j. The free DOF is isolated by means of a transformer.

The free DOF can be isolated by using an unconventional 1×6 transformation matrix such that it is possible transform the 6-dimensional twist to a 1-dimensional velocity. To the free DOF, also actuators can be applied.

B.2.2 Euler Lagrange equations

Instead of making the model using bond graphs, one can also choose to write the Euler Lagrange equations directly. Recall Eq. (1.4) which represents the equation of motion for a robot with n DoF:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{F}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} = \boldsymbol{\tau} \,. \tag{B.12}$$

Using methods described in [37], it is easy to find formulations for the different terms. The potential energy of the robot can be written as the sum of the potentials of the individual links:

$$V(\boldsymbol{q}) = \sum_{i=1}^{n} V_i(\boldsymbol{q}) \,. \tag{B.13}$$

The potential of body i can be written as,

$$V_i(\boldsymbol{q}) = m_i g p_{i,z}^{0,0} \tag{B.14}$$

where m_i is the mass of link, g the free-fall acceleration and $p_{i,z}^{0,0}$ the z component of the position of body i with respect to the inertial frame. The gravity compensation is then the derivative of $V(\boldsymbol{q})$ with respect to $\boldsymbol{q}$:

$$\boldsymbol{G}(\boldsymbol{q}) = \frac{\partial V(\boldsymbol{q})}{\partial \boldsymbol{q}} \,. \tag{B.15}$$

The mass matrix can be written as sum of the individual components corresponding to each link,

$$\boldsymbol{M}(\boldsymbol{q}) = \sum_{i=1}^{n} \boldsymbol{M}_i(\boldsymbol{q}) \tag{B.16}$$

where each individual mass matrix components for body i is written as

$$\boldsymbol{M}_i(\boldsymbol{q}) = \boldsymbol{J}_i^\top \operatorname{Ad}_{\boldsymbol{H}_0^i}^\top \boldsymbol{I}^i \operatorname{Ad}_{\boldsymbol{H}_0^i} \boldsymbol{J}_i \,. \tag{B.17}$$

$\boldsymbol{J}_i$ denotes the partly filled geometric Jacobian,

$$\boldsymbol{J}_i = \left[\begin{array}{cccccc} \hat{\boldsymbol{T}}_1 & \hat{\boldsymbol{T}}_2 & \cdots & \hat{\boldsymbol{T}}_i & \mathbf{0} \end{array}\right] \tag{B.18}$$

where

$$\hat{\boldsymbol{T}}_i = \hat{\boldsymbol{T}}_i^{0,i-1} \tag{B.19}$$

are the unit twists of each joint. The Coriolis term can be written as [38],

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \dot{\boldsymbol{M}}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \frac{1}{2}\frac{\partial}{\partial \boldsymbol{q}}(\dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}) \tag{B.20}$$

Another and more usual approach to calculate the components of $\boldsymbol{C}$ is to use the Christoffels symbols [38].
A final note is on the implementation of the equations. The dynamics are found symbolically using the MATLAB symbolic toolbox. In each consecutive step, the `simplify()` command is used in order to reduce the enormous lengths of the expressions. However, MATHEMATICA might be a more suitable software environment to work with symbolic expressions like these.

B.2.3 Modelling 5-DoF robot

The two modelling techniques are evaluated on the 5-DoF robot arm used in this work. The combination of bond graphs together with screw theory is powerful as it allows to model rigid multi body dynamics in a fast and convenient way. Furthermore, the modelling language fits well within the energy paradigm used in this work. However, a major drawback is that the number of states of the model increases dramatically with the number of bodies and joints. This is because each body and joint contains at least 6 states (only counting storage elements). In the proposed optimization, the dynamics are simulated in every objective evaluation. Therefore, the speed in which the dynamics are solved becomes of importance. Instead, writing the Euler Lagrange equations directly allows to solve only for the generalized coordinates which increases the solving speed of the dynamics. That is why we use the Euler Lagrange equations as model in the optimization.

Kinematics

This section provides how the forward kinematics are formulated on the 5-DoF robot corresponding to Fig. 1 in the paper. The theory described in [8] will be used to formulate the equations. Consider the generalized coordinates to be written as

$$\boldsymbol{q} = \left[\begin{array}{ccccc} q_1 & q_2 & q_3 & q_4 & q_5 \end{array}\right]^\top . \tag{B.21}$$

Instead of writing the unit twists directly in Ψ_0, the unit-twists are first written in Ψ_{i-1} and moved to the inertial frame after that,

$$\hat{\boldsymbol{T}}_i^{0,i-1} = \operatorname{Ad}_{\boldsymbol{H}_{i-1}^0} \hat{\boldsymbol{T}}_i^{i-1,i-1} \tag{B.22}$$

Where $\hat{\boldsymbol{T}}_i^{i-1,i-1}$ can be written as,

$$\hat{\boldsymbol{T}}_i^{i-1,i-1} = \begin{bmatrix} \boldsymbol{\omega}_i^{i-1,i-1} & \boldsymbol{r}_i^{i-1,i-1} \wedge \boldsymbol{\omega}_i^{i-1,i-1} \end{bmatrix}^\top = \begin{bmatrix} \boldsymbol{\omega}_i^{i-1,i-1} & \tilde{\boldsymbol{r}}_i^{i-1,i-1}\boldsymbol{\omega}_i^{i-1,i-1} \end{bmatrix}^\top \tag{B.23}$$

where $\boldsymbol{\omega}_i^{i-1,i-1}$ and $\boldsymbol{r}_i^{i-1,i-1}$ describe the rotation and position-vector of joint i with respect to joint $i-1$. For this 5-DoF robot the unit twists are composed using,

$$\begin{aligned}
\boldsymbol{\omega}_1^{0,0} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top \\
\boldsymbol{\omega}_2^{1,1} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top \\
\boldsymbol{\omega}_3^{2,2} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top \\
\boldsymbol{\omega}_4^{3,3} &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^\top \\
\boldsymbol{\omega}_5^{4,4} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^\top
\end{aligned} \tag{B.24}$$

and

$$\begin{aligned}
\boldsymbol{r}_1^{0,0} &= \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^\top \\
\boldsymbol{r}_2^{1,1} &= \begin{bmatrix} 0 & 0 & L_1 \end{bmatrix}^\top \\
\boldsymbol{r}_3^{2,2} &= \begin{bmatrix} 0 & 0 & L_2 \end{bmatrix}^\top \\
\boldsymbol{r}_4^{3,3} &= \begin{bmatrix} 0 & 0 & L_3 \end{bmatrix}^\top \\
\boldsymbol{r}_5^{4,4} &= \begin{bmatrix} 0 & 0 & L_4 \end{bmatrix}^\top
\end{aligned} \tag{B.25}$$

To express the configuration of end effector, Brockett's formula can be used where the unit twists are evaluated at $\boldsymbol{q} = \boldsymbol{0}$,

$$\boldsymbol{H}_n^0(\boldsymbol{q}) = \prod_{i=1}^{n} \left(e^{\tilde{\hat{\boldsymbol{T}}}_i(\boldsymbol{q}=\boldsymbol{0})q_i} \right) \boldsymbol{H}_n^0(0) \tag{B.26}$$

where $\boldsymbol{H}_n^0(0)$ is the initial configuration of the end-effector. $e^{\tilde{\hat{\boldsymbol{T}}}_i(\boldsymbol{q}=\boldsymbol{0})q_i}$ is the exponential map and can be calculated using Rodriguez formula. Or using directly the matrix exponential function in MATLAB, `expm()`.

20Sim model

The model is also being made in 20Sim using the technique described in Section B.2.1. The overall structure is given in Fig. B.5.

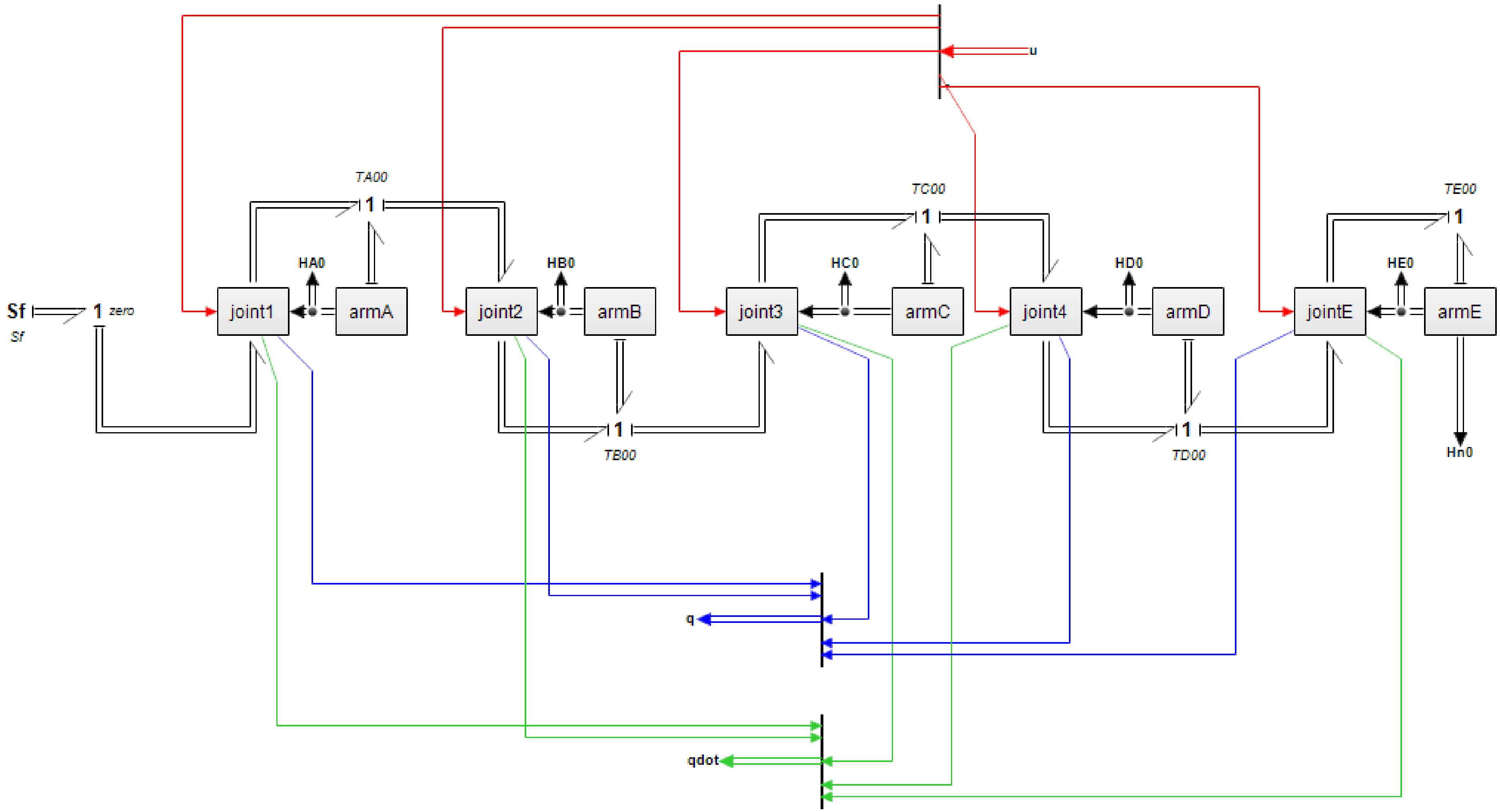


Figure B.5: 20sim model for the 5-DoF robot. The 'arm' and 'joint' subblocks corresponds to the bond graph structures in Figures B.2 and B.3 respectively.

Appendix C

Notes on geometry

C.1 The end-point mobility tensor

The following definition for the end-point mobility tensor is taken from [39]. $\boldsymbol{M}(\boldsymbol{q})$ represents the mass matrix in the joint space. It is positive definite and symmetric. To construct the kinetic energy, it takes two vectors and produces a scalar:

$$E_k = \frac{1}{2}\dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}}\,. \tag{C.1}$$

For example,

$$\dot{\boldsymbol{q}}^\top \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} = 1 \tag{C.2}$$

represents an ellipsoid at the configuration $\boldsymbol{q}$. As $\boldsymbol{M}(\boldsymbol{q})$ is positive definite the inverse of $\boldsymbol{M}(\boldsymbol{q})$ will contract the co-vector of $\dot{\boldsymbol{q}}$ to a scalar:

$$\boldsymbol{\tau}^\top \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\tau} = 1\,. \tag{C.3}$$

Is also an ellipsoid on the configuration of $\boldsymbol{\tau}$. Using the Jacobian we can map the wrench in the workspace to the torques in the joint space:

$$\boldsymbol{\tau} = \boldsymbol{J}(\boldsymbol{q})^\top \boldsymbol{W^0}\,. \tag{C.4}$$

Substituting equations will result in

$$(\boldsymbol{W^n})^\top (\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{\boldsymbol{-1}}\boldsymbol{J}(\boldsymbol{q})^\top)\boldsymbol{W^n} = 1\,. \tag{C.5}$$

The *end-point mobility tensor*[1] is then defined as the part between brackets of the latter equation:

$$\boldsymbol{\Lambda^{-1}} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{\boldsymbol{-1}}\boldsymbol{J}(\boldsymbol{q})^\top\,. \tag{C.6}$$

To see what $\boldsymbol{\Lambda}^{-1}$ implies, we derive $\boldsymbol{\Lambda}^{-1}$ in another way. At rest, the joint positions and corresponding velocities are zero (or fully compensated gravity and zero joint velocity). In this situation, the equation of motion in the joint space (Eq. (1.4)) results in

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{\tau}\,. \tag{C.7}$$

[1]This term was initially introduced by Hogan(1984) in [40].

Isolating $\ddot{\boldsymbol{q}}$ and multiply both sides of the equation with $\boldsymbol{J}(\boldsymbol{q})$ results in

$$\dot{\boldsymbol{T}}_n^{0,0} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{\tau}\,. \tag{C.8}$$

By substituting equation (C.4) the acceleration response in the end-effector is defined as

$$\dot{\boldsymbol{T}}_n^{0,0} = \underbrace{\boldsymbol{J}(\boldsymbol{q})\boldsymbol{M}^{-1}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^\top}_{\boldsymbol{\Lambda}^{-1}} \boldsymbol{W}^{\boldsymbol{0}}\,. \tag{C.9}$$

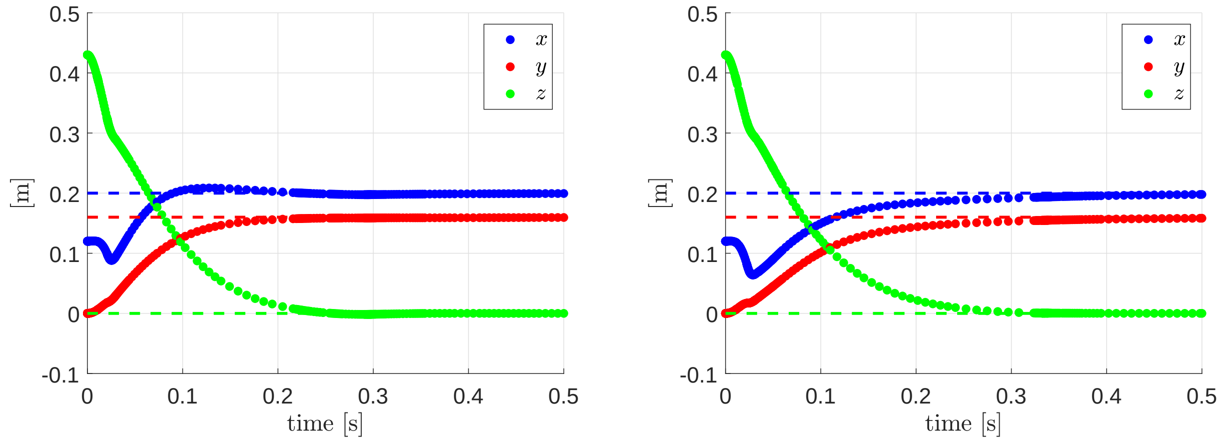
Equation (C.9) describes the acceleration response at the end-effector when the manipulator is at rest. Because $\boldsymbol{\Lambda}^{-1}$ is weighted by the Jacobian, it can have rank deficiency. For under-actuated robots or when the robot is in a singular position, a wrench at the end-effector will not provide a velocity twist at the end-effector (like in the joint space). The inverse of Lambda is symmetric but semi positive definite, unlike $\boldsymbol{M}(\boldsymbol{q})$. If $\boldsymbol{\Lambda}^{-1}$ is full rank, then the inverse exists and $\boldsymbol{\Lambda}$ can be calculated.

C.2 Cartesian damping

The 'critically' damped approach for multi-DoF systems was presented in Section IV-B in the paper. As indicated, the critically damped approach only works if $\boldsymbol{\Lambda}^{-1}$ is invertible or in other words: for non-singular configurations and non-underactuated robots. In addition, the diagonalization of the equation of motion is in theory only valid when the gravitational, Coriolis and friction forces of the robot are cancelled. In this section we study the effect of cancelling the Coriolis terms on the 'critically' damped statement. Fig. C.1 shows two graphs in which the entries of the position vector belonging the configuration of the end-effector ($\boldsymbol{p}_n^{0,0}$) are plotted. The robot is controlled with a Cartesian impedance controller using a fixed spring value and scaling the damper together with the inertia and the stiffness as explained in Section IV-B in the paper. A fixed virtual frame is used corresponding to,

$$\boldsymbol{H}_v^0 = \begin{pmatrix} -1 & 0 & 0 & 0.2 \\ 0 & 1 & 0 & 0.16 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{C.10}$$

The initial joint positions are given as $\boldsymbol{q} = \begin{bmatrix} 0 & 0 & 0 & \pi/2 & 1 \end{bmatrix}$. This initial configuration corresponds to initial robot configuration in Fig. 7 in the paper.



(a) $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is not cancelled

(b) $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is cancelled

Figure C.1: Entries of $\boldsymbol{p}_n^{0,0} = \begin{bmatrix} x & y & z \end{bmatrix}^\top$ using $\boldsymbol{K} = \mathrm{diag}(750 \cdot \boldsymbol{I}_3, 3000 \cdot \boldsymbol{I}_3,)$

Comparing both figures, we see that when $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is cancelled the position components do not overshoot their corresponding reference ('critically damped'). When $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is not cancelled, the x position will overshoot at ≈ 0.1s. Furthermore, a 'drop' in x position is noticed for both figures. When we study the initial configuration of the robot (again, see Fig. 7 in the paper). Then the 'drop' in x position can be explained due to the rotation at the end-effector. All in all, the Coriolis terms have an effect on the dynamics and thus on the position vector of the end-effector.

Appendix D

Mathematical background

D.1 Derivations

In the following, the port-Hamiltonian equation is derived as given in Eq. (2.17). The labelling of effort and flow variables correspond to the bond graph in Fig. D.1.
First derive the state space equation:

$$
\begin{aligned}
\dot{f}_2 &= \frac{1}{L} e_2 \\
&= \frac{1}{L} [e_1 - e_4] \\
&= \frac{1}{L} [e_1 - u e_5] \\
&= \frac{1}{L} [e_1 - u [e_6 + e_7]] \\
&= \frac{1}{L} [e_1 - u [R f_6 + k_t f_8]] \\
&= \frac{1}{L} [e_1 - u [R u f_2 + k_t f_8]]
\end{aligned} \tag{D.1}
$$

$$
\begin{aligned}
\dot{f}_3 &= \frac{1}{m} e_3 \\
&= \frac{1}{m} [e_8 - e_9] \\
&= \frac{1}{m} [k_t f_7 - b(x_2)] \\
&= \frac{1}{m} [k_t f_5 - b(x_2)] \\
&= \frac{1}{m} [k_t u f_4 - b(x_2)] \\
&= \frac{1}{m} [k_t u f_2 - b(x_2)]
\end{aligned} \tag{D.2}
$$

Substitution of $L = 1\text{H}$, $f_2 = f_t = p_t$, $e_1 = F_{dyn}$, $f_8 = x_2$, $F_c = e_8 = e_3 + e_9 = m\dot{f}_3 + b(x_2)$ results in

$$\dot{p}_t = F_{dyn} - u^2 R p_t - u k_t x_2 \tag{D.3}$$

$$F_c = k_t u p_t \, . \tag{D.4}$$

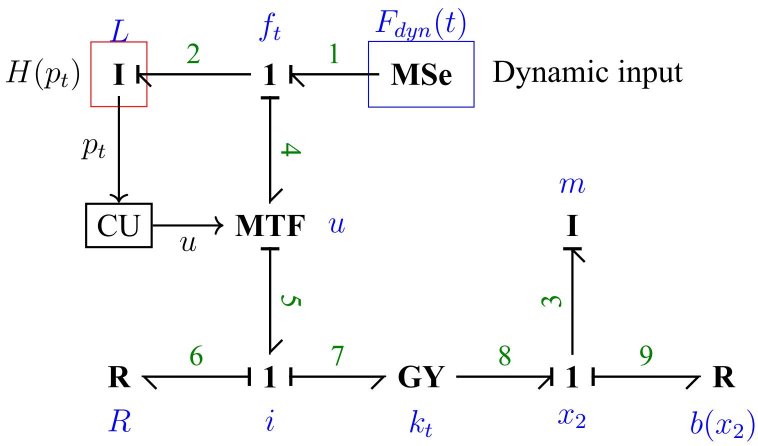


Figure D.1: Bond graph representation of equation 2.1 driven by a motor connected to the energy tank. I_t is the inertia of the tank, f_t the current through the tank.

D.2 Operators

- *The* as$(\cdot)$ *operator*,
  Every square matrix can be decomposed into a symmetric part and a anti symmetric part:

$$\boldsymbol{A} = \underbrace{\frac{1}{2}(\boldsymbol{A} + \boldsymbol{A}^\top)}_{symmetric} + \underbrace{\frac{1}{2}(\boldsymbol{A} - \boldsymbol{A}^\top)}_{anti-symmetric} \,. \tag{D.5}$$

  as$(\boldsymbol{A})$ represents the anti-symmetric part of input argument $\boldsymbol{A}$. $\boldsymbol{A}$ has to be a square matrix.

- *The* Ad$(\cdot)$ *operator,*
  The Adjoint operator is introduced to translate twists and wrenches from one frame to another. An example, to translate the twist expressed in Ψ_b to the twist expressed in Ψ_a:

$$\boldsymbol{T}_d^{a,c} = \mathrm{Ad}_{\boldsymbol{H}_b^a} \boldsymbol{T}_d^{b,c} = \begin{pmatrix} \boldsymbol{R}_b^a & \boldsymbol{0} \\ \tilde{\boldsymbol{p}}_b^{a,a} \boldsymbol{R}_b^a & \boldsymbol{R}_b^a \end{pmatrix} \boldsymbol{T}_d^{b,c} \,. \tag{D.6}$$

  As the wrench is a co-vector and the dual of the twist, it transforms with the transpose of the Adjoint. Therefore:

$$(\boldsymbol{W}^a)^\top = \mathrm{Ad}^\top_{\boldsymbol{H}_a^b} (\boldsymbol{W}^b)^\top \,. \tag{D.7}$$

- *The* ad$(\cdot)$ *operator,*
  The ad$_{\boldsymbol{T}}$ operator of input twist $\boldsymbol{T}$ is defined as,

$$ad_{\boldsymbol{T}} = \begin{pmatrix} \tilde{\boldsymbol{\omega}} & \boldsymbol{0} \\ \tilde{\boldsymbol{v}} & \tilde{\boldsymbol{\omega}} \end{pmatrix} \tag{D.8}$$

  where the twist is defined as $\boldsymbol{T} = \begin{bmatrix} \boldsymbol{\omega} & \boldsymbol{v} \end{bmatrix}$ and $\tilde{(\cdot)}$ is the skew-symmetric operator.

- *The* $\mathrm{tr}(\cdot)$ *operator,*
  The trace operator takes a square matrix and produces the sum of its diagonal elements,

$$\mathrm{tr}(\boldsymbol{A}) = \sum_{i=0}^{n} a_{ii} \tag{D.9}$$

  where $\boldsymbol{A}$ is a $n \times n$ square matrix and a_{ij} are the entries of $\boldsymbol{A}$.

- *The* $(\tilde{\cdot})$ *operator,*
  The tilde operator defines the skew symmetric matrix of input vector $\boldsymbol{x}$,

$$\tilde{\boldsymbol{x}} = \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix} \tag{D.10}$$

  where $\boldsymbol{x} = [\ x_1 \quad x_2 \quad x_3\ ]^\top$. The skew symmetric matrix has the property that

$$\tilde{\boldsymbol{x}} = -\tilde{\boldsymbol{x}}^\top . \tag{D.11}$$

- *The* $\|\cdot\|_{FRO}$ *operator,*
  The Frobenius norm is the square root of the sum of all absolute entries of a matrix,

$$\|\boldsymbol{B}\|_{FRO} = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{m} |b_{ij}|^2} \tag{D.12}$$

  where $\boldsymbol{B}$ is a $n \times m$ matrix and b_{ij} are the entries of $\boldsymbol{B}$.

- *The* $\|\cdot\|_1$ *operator,*
  The 1-norm of a vector $\boldsymbol{x}$ with length n is defined as the sum of absolute entries in $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{n} |x_i| . \tag{D.13}$$

- *The* $\|\cdot\|_2$ *operator,*
  The 2-norm of a vector $\boldsymbol{x}$ with length n is defined as the root square from the sum of the square absolute entries in $\boldsymbol{x}$:

$$\|\boldsymbol{x}\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2} . \tag{D.14}$$