MASTER THESIS

PASSIVE MASS REDUCTION IN ADMITTANCE CONTROL

Bas Koster

FACULTY OF ENGINEERING TECHNOLOGY
DEPARTMENT OF BIOMECHANICAL ENGINEERING

EXAMINATION COMMITTEE
prof. dr. ir. H. van der Kooij
dr. ir. A.Q.L. Keemink
dr. ir. G.A. Folkertsma
A. Vallinas Prieto, MSc

DOCUMENT NUMBER
BW - 764

UNIVERSITY OF TWENTE.

29/10/2020

Contents

Chapter 1

Thesis introduction

In recent years, more attention has risen for robots cooperating with a human operator. Examples of this would be industrial manipulation, rehabilitation, or medical applications, such as powered exoskeletons. Because of this close cooperation and humans sharing the same work-space with a robot, safety must always be guaranteed. Different approaches have been developed for reliable and safe human-robot interaction. One of these is admittance control, which can make existing non-backdrivable manipulators show desired behavior.

In admittance control, the forces exerted by an operator are measured and imposed on a virtual model. The positions and velocities from this virtual model are then used as setpoints for the robot to follow. Doing so ideally would make the robot behave as if it had the properties of the virtual model. This would allow for i.e. the removal of friction in the robot or cancellation of gravity.

Fig. 1 shows the schematic layout of an admittance control system. In this, Y_v is the virtual model, converting the measured force F into the desired velocity v_d. The controller C_{fb} takes the difference e between this desired velocity and the actual velocity v and uses it to compute a force F_c, which aims to make the dynamics of the robot Y_r follow the desired velocity v_d.

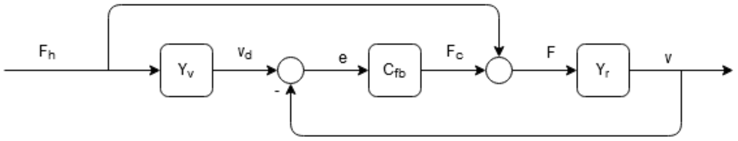

Figure 1: Schematic diagram of an admittance controlled system.

One of the examples in which admittance control is used is powered exoskeletons, which are coupled to the human body. They aim to assist the operator with moving by providing additional force or to amplify force exerted by the operator [1]. Another example is industrial robots helping operators with lifting or positioning heavy payloads [2]. However, not in all cases is the assistance of the robot required. Then the apparent admittance should be much higher than the natural admittance of the robot, making it easier to move. Ideally, the apparent admittance in that situation should be infinite.

An infinite admittance is impossible in reality, due to a division by zero, so the aim is to increase the admittance as much as possible. One of the ways to achieve this is to reduce the mass present in the virtual system, which will cause the mass of the robot to be masked.

The problem shows when an operator attempts to operate such a mass-reduced admittance-controlled device. In this situation, mechanical power is exchanged between the operator and the device. A schematic representation of this can be seen in Fig. 2. In this case, the force created by the operator causes a coupled motion. The operator has an impedance which results in a reaction force. This impedance is time-varying in nature, and often also non-linear. The reaction force is measured by the device, effectively creating a force feedback loop. This closed loop can under certain conditions negatively affect the stability of the system.

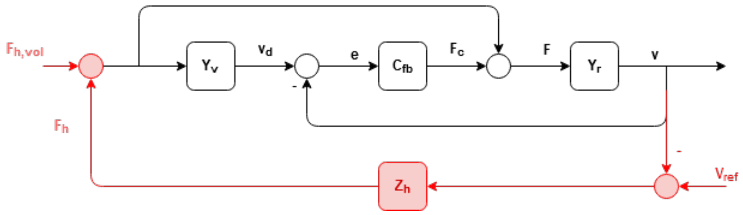

Figure 2: Schematic diagram of a coupled admittance controlled system. The red components belong to the feedback loop due to the addition of a human operator.

If the virtual model is virtual mass much lower than the robot mass and the system is coupled to an operator with a high impedance, a normal velocity PI controller will lead to unstable behavior [3]. As this is highly undesirable, extensive research has been done to prevent this. One elegant solution to this can be found in passivity theory.

Passivity is a concept often used in control design. A system is passive if the energy that can be extracted from the system is bounded by the injected and initially stored energy. In other words, no more energy can be extracted from the system than was already put into the system. For example, consider the admittance controlled system shown in Fig. 1 with the input force $\mathbf{F}_h(t)$ and the device velocity $\mathbf{v}(t)$. This combination of a force and velocity vector is also referred to as a power-port. The energy exchanged between the device and the operator can be expressed as:

$$E(t) = \int_0^t \mathbf{F}_h^T(\tau)\mathbf{v}(\tau)d\tau + E_0, \tag{1}$$

in which E_0 is the initially stored energy. In this case, positive energy means that energy was transferred to the system. Negative energy means that energy was transferred to the operator. For a system to be passive, only one condition must hold:

$$E(t) \geq E_0, \forall t. \tag{2}$$

Passive systems, as well as interactions of passive systems, are stable [4]. Humans, although considered to be active elements, are very capable to interact with passive systems. Furthermore, involuntary human forces are mostly passive. [5]

1.1 Objective

Multiple attempts were made in recent years to guarantee passivity, described in section I. In these, the virtual mass was kept equal and by utilizing modified forces or variable virtual dampers, passivity was preserved or restored. Feragutti et al. [6] introduced the idea of varying the virtual mass as a way to guarantee stability. Increasing the virtual mass would create virtual kinetic energy, which could violate the passivity of the virtual model. This problem was solved by using energy tanks to guarantee the passivity of the virtual model. The concept of varying virtual mass as a means to guarantee a passive interaction is investigated in this report. The objective is to develop a framework that can guarantee passive interaction of a mass-reduced admittance controlled robot by using variable virtual dynamics.

With this objective does not come a strict demand on the amount of mass-reduction used. From literature, it is known that an upper bound exists on the amount of passive mass-reduction possible, and in this literature, a fixed amount of virtual mass is used. However, since variable virtual mass will be used, it is not possible to fix the amount of mass-reduction beforehand. Instead the amount of mass-reduction that can be achieved will be investigated, as well as ways to reach more mass-reduction.

The presented framework will be tested on a two degree of freedom (DoF) robot in simulation, where the end-effector attempts to behave as prescribed by the virtual mass-damper model. This application is closer to the industrial robot helping with payload manipulation. The chosen task is for the operator to make the robot follow a reference trajectory that is relatively small compared to the workspace at relatively high speed.

This report is divided into three distinct parts. This chapter serves to provide background information to the thesis and to give context and information on specific elements that were focused on in this report. In section II, a novel framework will be shown which aims to guarantee passivity by exclusively modifying the virtual dynamics. An optional addition, if the robot allows for direct force input, is detailed in section III. The simulation set-up to test and show both systems can be found in section IV, the results of which are shown in section V. The discussion about these results can be found in section VI. Lastly, chapter 3 is a more elaborate discussion, mostly touching on all points not mentioned in VI. It also looks back on the apparent end-effector dynamics and the estimated energy exchanged at the end-effector.

Passive Variable Mass Reduction in Admittance Control

Bas KOSTER (s1657143)

***Abstract*—Admittance control has been used in recent years to reproduce a desired dynamic response on non-backdrivable manipulators. One of the topics of research has been to increase the apparent admittance by reducing the virtual mass. However, mass-reduced admittance controllers become unstable when the environment (e.g. the operator) the robot is interacting with has too high impedance. In this paper, a framework is proposed to guarantee passive interaction between robot and human operator, while attempting to increase admittance as much as possible. An additional adaptive feedforward component is proposed, if direct force control is possible, to increase the apparent admittance further. The proposed framework is tested in simulation and has shown to indeed reduce apparent admittance.**

I. INTRODUCTION

In recent years, more attention has risen for robots cooperating with a human operator. Examples of this would be industrial manipulation, rehabilitation or medical applications, such as powered exoskeletons. Because of this close cooperation and humans sharing the same workspace with a robot, safety must always be guaranteed. Different approaches have been developed for reliable and safe human-robot interaction. One of these is admittance control, which can make existing non-backdrivable manipulators show desired behavior. A schematic diagram is shown in Fig. 3. Admittance control is used in various fields, from exoskeleton control [1] [2] to end-effector manipulation [3] [4]. In these cases, admittance control aims to remove undesirable behavior or amplify the force exerted by an operator. However, when the assistance of the controller is not required, the robot should have high admittance, preferably higher than the natural admittance of the robot. One of the ways to do this is to use a lower mass in the virtual model than the actual robot. The problem with this is that the robot becomes unstable when interacting with stiff environments, such as stiffened human limbs. An elegant solution for this problem could be found in passivity theory. Passivity is a concept often used in control design. A system is passive if the energy that can be extracted from the system is bounded by the injected and initially stored energy. In other words, no more energy can be extracted from the system than was already put into the system. Passive systems, as well as interactions of passive systems, are stable [5]. Humans, although considered to be active elements, are very capable to interact with passive systems. Furthermore, involuntary human forces are mostly passive [6].

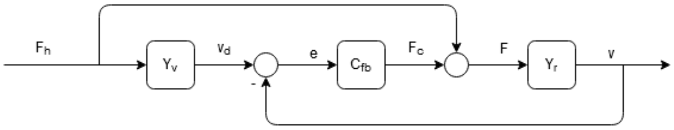

Fig. 3: Schematic of admittance controlled system.

A. Current research

Passivity can be enforced on admittance controlled devices by having certain parameters, such as controller gains or virtual model parameters, within certain ranges. One of the guidelines to find these ranges is the positive real condition [7], used for linear time-invariant (LTI) systems. In [8], Keemink performed an analysis of a single degree of freedom LTI mass-damper system, controlled by an admittance controller with an LTI virtual mass. He found that the solution provided by the positive real condition is rather conservative. The lower bound of the virtual mass was lower than the actual mass, but only barely so. The reason for the conservative nature of the positive real condition is that it checks it the power-port of the system is passive in the frequency domain, which is more strict than time-domain passivity. Several papers have been written with suggested frameworks for enforcing time-domain passivity. One of the more recent papers is by Nabeel et al. to prevent activity when mass reduction is intended [9]. It is based on a method to prevent activity in the interaction of impedance controlled systems, developed by Hannaford and Ryu [10]. They developed the so-called PO/PC method to enforce proper energy transfer between not mechanically coupled systems, such as a master-slave setup in teleoperation. The Passivity Observer (PO) observes all energy E_{obs} exchanged through power-ports that are estimated to display active behavior. When the Passivity Observer registers activity ($E_{obs} < 0$), the Passivity Controller activates and attempts to dissipate this excess energy. Nabeel et al. considered the copying of the virtual velocity as the desired motion as the source of activity. To counter this, they used a PO observing the energy going into the controller and the energy coming out of the motion-controller because of the virtual dynamics. The PC is activated when the output is greater than the input. Keemink [8] came however with a counterexample to the approach of Nabeel et al. which showed that their proposed method did not work for all systems. He used a virtual model with a mass half of the real system and a position-proportional controller. The environment was purely elastic. The resulting simulation was unstable. Keemink [8] also proposed a different approach. The Strict Passivity Inspired Stability (SPIS) applies a linear fixed damper on the virtual model only when the strict passivity condition $E_{hr} \geq H_v$ no longer holds. In this, E_{hr} is the energy exchanged between operator and robot, and H_v the energy stored in the virtual model. It has shown that it can stabilize the tested system, but it is not able to guarantee passivity. Furthermore, the discontinuous nature of the controller meant the operator could suddenly feel a jerky movement when the damper activates. The method shows that a virtual damper does help in restoring passive behavior, but that at the same time it may be a conservative approach. Recently, a paper was written by Ferraguti et al. [11] introducing a method to stabilize an admittance controlled system by varying the virtual dynamics. While not guaranteeing passivity, it does introduce a new method to potentially guarantee passivity. As such, it forms the basis for the framework presented in this paper.

II. Passivity Preserving Framework

This section will show the passivity preserving framework. First the mechanism to be able to vary the virtual mass passively derived in [11] is detailed. After this, the framework is derived.

A. Background

Ferraguti et al. [11] presented a method to restore stability of an admittance-controlled robot, by increasing the virtual mass without breaking passivity of the virtual model. Additionally, the use of a constant mass-to-damping ratio means that an increase in virtual mass also increases in virtual damping, which is beneficial to enforcing passivity with mass-reduction ([8],sec. 4.6.4). This results in the following equation of motion at sample time k:

$$\mathbf{M}(k)\dot{\mathbf{v}}_d(k) + \mathbf{M}(k)\mathbf{R}_\mathbf{d}\mathbf{v}_d(k) = \mathbf{F}_h(k). \tag{3}$$

In the method proposed by Ferraguti et al, the issue with using a variable virtual mass matrix is the loss of passivity of the virtual model. This can be shown by looking at the energy storage in the virtual model, in this case a kinetic energy model:

$$H(\mathbf{v}_d(k)) = \frac{1}{2}\mathbf{v}_d^T(k)\mathbf{M}(k)\mathbf{v}_d(k), \tag{4}$$

where $\mathbf{v}_d(k)$ is the velocity vector of the virtual model at time t and $\mathbf{M}_v$ the virtual mass matrix at time t. If the time-derivative is taken and (3) substituted, one obtains:

$$\begin{aligned}&\dot{H}(\mathbf{v}_d(k)) = \\ &\quad \mathbf{F}_h^T(k)\mathbf{v}_d(k) + \frac{1}{2}\mathbf{v}_d^T(k)(\dot{\mathbf{M}}(k) - 2\mathbf{M}(k)\mathbf{R}_d)\mathbf{v}_d(k),\end{aligned} \tag{5}$$

with $\mathbf{F}_h(k)$ the force of the operator at time t, and $\mathbf{R}_d$ the mass-to-damping matrix. As can be seen, the term between brackets can be positive, and as such, it is possible that energy is injected due to an increasing mass matrix. This means that the virtual model is no longer passive at all times. The solution for this is to use energy tanks, which allows for the use of (virtual) energy present in the system in multiple ways. Here the energy dissipated by the virtual damper is stored in the energy tank and can be reused to increase the mass in a way that preserves the passivity of the virtual model. Therefore, the dynamics are expanded as follows:

$$\begin{aligned}\mathbf{M}(k)\dot{\mathbf{v}}_d(k) + \mathbf{M}(k)\mathbf{R}_\mathbf{d}\mathbf{v}_d(k) &= \mathbf{F}_h(k)\\ \dot{z} = \frac{\phi(k)}{z(k)}P_D(k) &- \frac{\gamma(k)}{z(k)}P_M(k)\end{aligned} \tag{6}$$

where

$$\begin{aligned}P_D(k) &= \mathbf{v}_d^T(k)\mathbf{M}(k)\mathbf{R}_d\mathbf{v}_d(k)\\ P_M(k) &= \frac{1}{2}\mathbf{v}_d^T(k)\dot{\mathbf{M}}(k)\mathbf{v}_d(k)\end{aligned} \tag{7}$$

is the power dissipated by the virtual damper, and the power injected or dissipated by the mass variation, respectively, and $z(k)$ the state of the tank. $\dot{\mathbf{x}}(k)$ is the velocity vector of the virtual model. Additionally, let

$$T(z(k)) = \frac{1}{2}z(k)^2 \tag{8}$$

be the energy stored in the tank. Due to the shape of (6), it is assumed that $T_{min} \leq T(z(k)) \leq T_{max}$ for all t, with $0 < T_{min} < T_{max}$. The upper bound of the tank is guaranteed by the parameters $\phi(k)$ and $\gamma(k)$, which disable the injection of energy in the tank in case the energy limit T_{max} is reached. The available energy in the tank needs to be bound, as the energy could become very high over time. This could potentially lead to situations where, although the virtual system as a whole is still passive, practically unstable behaviors could be implemented [12]. The following equations are used to enforce the upper bound:

$$\phi(k) = \begin{cases}1 & \text{for } T(z(k)) \leq T_{max}\\ 0 & \text{otherwise}\end{cases}, \tag{9}$$

which will disable or enable the injection of dissipated energy, and

$$\gamma(k) = \begin{cases}\phi & \text{if } \dot{\mathbf{M}}(k) \text{ semi-negative definite}\\ 1 & \text{otherwise}\end{cases}, \tag{10}$$

with $\phi \in \{0, 1\}$ reducing the amount of energy injected due to mass reduction, while it always allows energy to be extracted from the tank, allowing an increase of the virtual mass. The condition $\dot{\mathbf{M}}(k)$ semi-negative definite theoretically allows for the transfer of kinetic energy between degrees of freedom. This is prevented by assumptions on $\dot{\mathbf{M}}$ and design choices made. The lower bound, required to prevent a division by zero, is guaranteed by putting limits on the amount of energy extracted to increase the mass. For this, Ferraguti et al. make two relevant assumptions, namely:

- The desired inertia matrix $\mathbf{M}(k)$ as well as the product of $\mathbf{M}(k)\mathbf{R}_d$, also expressed as $\mathbf{B}_v(k)$, are diagonal matrices, defined as
$$\begin{aligned}\mathbf{M}(k) &= \text{diag}\{m_1(k), m_2(k), ...m_N(k)\}\\ \mathbf{B}_v(k) &= \text{diag}\{b_1(k), b_2(k), ...b_N(k)\},\end{aligned} \tag{11}$$
and as $\mathbf{M}(k)$ is diagonal, $\dot{\mathbf{M}}(k)$ is also diagonal.
- Every DoF of the virtual model has a robot-configuration-dependent maximum allowable velocity expressed as $\bar{\dot{x}}_i$ (i=1,...,N).

The entire derivation can be found in [11], but the following equation expresses the maximum allowable inertia variation per element of $\mathbf{M}$:

$$m_i(k+1) - m_i(k) = \frac{2w_i(T(z(k)) - T_{min})}{\bar{\dot{x}}_i^2}. \tag{12}$$

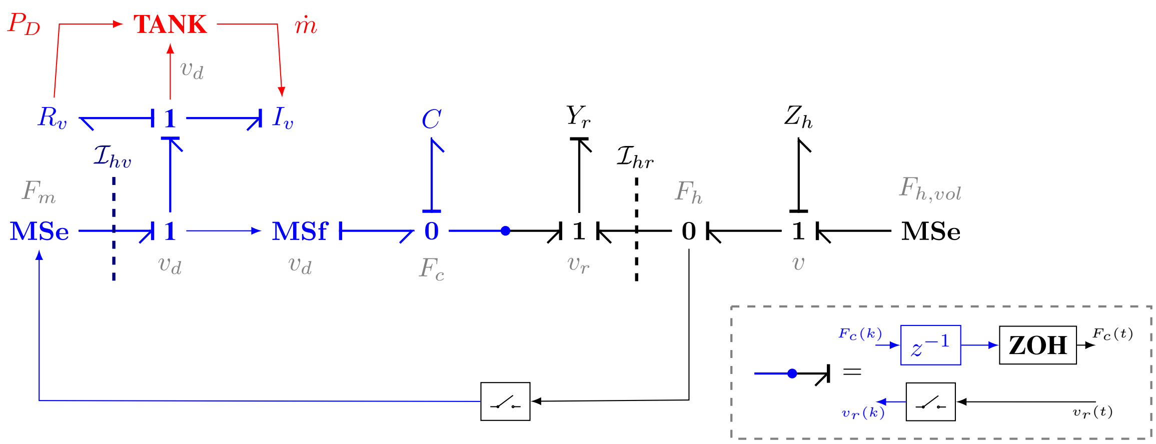

Fig. 4: Bond diagram of single DoF admittance controlled system adapted from [8] with permission. Blue components indicate virtual signals and powers, black denotes mechanical power flow and signals. Operators impedance $F_{h,vol}$ and voluntary force Z_h exert a force F_h on the robot dynamics Y_r, measured as F_m. This force is exerted via the modulated effort source (**MSe**) on the virtual mass I_v and virtual damping R_v, yielding virtual velocity v_d. This velocity is imposed on the controller C via the velocity source **MSf**, yielding a control force F_c that is put on the robot via the interface shown in the inset with dashed lines. Red indicates the energy tank and its required connections, allowing changes to the virtual mass without losing passivity of the virtual system. Having variable passive virtual mass allows for a mechanism to keep the operator-robot interaction passive.

w_i is the i'th component of a vector of weights $W = \{w_1, w_2, ..., w_N\}$, which divides the available amount of energy over the N degrees of freedom, and for which holds that

$$\sum_{i=1}^{N} w_i = 1. \tag{13}$$

Of course, it should be noted that (12) presents the maximum allowed inertia variation, energetically speaking, based on the amount of energy at that time instance. In practice, this could lead to very large mass fluctuations. As such, a second vector $\overline{M} = \{\overline{m}_1, \overline{m}_1, ..., \overline{m}_N\}$ is defined, which present a second upper bound:

$$m_i(k+1) - m_i(k) \leq \overline{m}_i, \text{ for } i = 1, ..., N. \tag{14}$$

(12) and (14) can then be used to compute the inertia variation matrix $\mathbf{S_j}$:

$$\mathbf{S_j} = \text{diag}\{s_1, s_2, ..., s_n\}, \tag{15}$$

with j denoting that this is the j'th instance of a variation of the virtual inertia. s_i is given by:

$$s_i = \min\left\{\overline{m}_i, \frac{2w_i(T(z(k)) - T_{min})}{\overline{\dot{x}}_i^2}\right\}. \tag{16}$$

The last step is the actual variation of the inertia parameters:

$$\mathbf{M}(k+1) = \mathbf{M}(0) + \sum_{p=0}^{j} \mathbf{S_p}\beta^{F_s(k-k_p))}, \tag{17}$$

with $\mathbf{M}(0)$ the virtual mass at time 0, F_s the sample frequency, and k_p the sample instance at which the p'th variation of the mass took place. $\beta(0 < \beta < 1)$ is a forgetting factor that allows for the desired interaction model, with which the robot is instantiated, to be restored over time. Lastly, the damping matrix needs to be updated:

$$\mathbf{B}_v(k+1) = \mathbf{M}(k+1)\mathbf{R}_d \tag{18}$$

B. Preserving passive interaction

While Ferraguti et al. successfully implemented a method to restore stability of an admittance controlled robot, it cannot guarantee passivity of the interaction between operator and robot. Being able to passively increase the virtual mass is very useful, but as important is knowing when to increase or decrease virtual mass to guarantee passivity in the interaction between robot and operator. In this section will be derived how the framework detects (potential) loss of passivity in the interaction and how it varies mass over time to guarantee or restore a passive interaction.

1) Detection algorithm: The new detection system consists out of two detection rules, each of which is checked at set intervals, which do not necessarily have to be the same as the intervals of the admittance controller. After checking the conditions, the mass is increased or decreased as described in section II-B2, depending on the conditions and situation. The first detection rule is simply the strict passivity condition, proposed in [8]:

$$E_{hr}(k) \geq H_v(k), \tag{19}$$

with k being the sample instance k, E_{hr} the energy exchanged between operator and robot, and H_v the energy stored in the virtual model. This is the passivity that will be enforced by this framework on the interaction between operator and robot. Simply only using this detection rule was observed in simulation to cause cyclic behavior in the virtual inertia, including, at times, large virtual masses. This is most likely caused by the delayed effect of adding virtual mass and damping, which does not instantly restore strict passivity. The solution to this problem is a second detection rule that aims to break this cyclic behavior. It is based on the following passivity condition:

$$\mathbf{F}^T\mathbf{v} \geq \frac{dE_{stored}}{dt}, \tag{20}$$

which simply states that the power supplied or withdrawn from the robot must be greater than the change in energy stored in the robot. For the second detection rule, the following intermediate equation is required:

$$P_{\mathit{diff}}(k) = \lambda P(k) + (1-\lambda)P_{\mathit{diff}}(k-1), \tag{21}$$

with $\lambda \in \{0, 1\}$ being a lowpass-filter constant, and $P(k)$ given by:

$$P(k) = \frac{E_{\mathit{hr}}(k) - E_{\mathit{hr}}(k-1)}{T_s} - \frac{H_v(k) - H_v(k-1)}{T_s}. \tag{22}$$

λ has the purpose of preventing noise, discretization errors, etc., from falsely triggering the second detection rule, which is:

$$P_{\mathit{diff}}(k) \geq 0. \tag{23}$$

Something to note here is that, simply due to the presence of the low-pass filter, the second detection rule is not sufficient to guarantee any kind of passivity by itself, nor is that its purpose. Because of that, λ does not have to be relatively large, giving the associated filter a large bandwidth. The only note here is that a negative $P(k)$ means that energy is returned to the operator in a way that violates (20). This means that while this behavior may be strictly speaking passive according to (19), it could be considered undesirable, which is an argument for not setting λ and the associated bandwidth of the lowpass-filter too low.

2) Virtual mass variation: The original mass-varying method of Ferraguti et al. did not explicitly have mass reduction in mind. As such, a new method had to be derived. This method will use a linear increase or decrease of virtual mass over time:

$$\mathbf{M}(k) = \mathbf{M}_{v,0} + \Delta\mathbf{M}(k), \tag{24}$$

with $\mathbf{M}_{v,0}$ the desired virtual mass, $\Delta\mathbf{M}(k)$ being a matrix of the shape $\Delta M(k) = \text{diag}\{\Delta m_1(k), \Delta m_2(k), ..., \Delta m_N(k)\}$, of which each element is either positive or zero. Every single element is calculated as follows:

$$\Delta m_{n,step}(k) = \Delta m_n(k-1) + \begin{cases} \psi s_i & \text{if not (19) or not (23))} \\ 0 & \text{otherwise} \end{cases}, \tag{25}$$

with s_i as described in (16). ψ is another intermediary variable, defined as:

$$\psi = \begin{cases} 1 & \text{if } (H_v(k) - E_{\mathit{hr}}(k))/P_{\mathit{diff}}(k) > T_p \\ & \text{or } (H_v(k) - E_{\mathit{hr}}(k))/P_{\mathit{diff}}(k) < 0 \\ & \text{or } H_v(k) \leq E_{\mathit{hr}}(k) \\ \psi_r & \text{otherwise} \end{cases} \tag{26}$$

where T_p and $\psi_r \in \{0, 1\}$ are constants. The reason for the first condition in (26) is to check if it is going to satisfy the first detection rule within the time specified by T_p. If it appears that the first detection rule is going to be satisfied within the specified time T_p, it is possible to reduce the rate at which mass is added to the virtual system. Adding more mass has a diminishing return on the rate at which passivity according to the first detection rule can be restored, and it may not be desirable to needlessly add more mass to the virtual system. The other two conditions are to prevent an edge case and to prevent ψ_r from affecting the mass increase rate when the second detection rule is triggered, respectively. After this, the mass reduction step is applied:

$$\Delta m_n(k) = \begin{cases} \max\{\Delta m_{n,step}(k) - M_{\mathit{loss}}, 0\} & \text{if } P_{\mathit{hr}} > P_{\mathit{min}} \\ & \text{and } E_{\mathit{hr}}(k) > H_v(k) \\ \Delta m_{n,step}(k) & \text{otherwise} \end{cases} \tag{27}$$

with M_{loss} the virtual mass decrease rate. It is chosen here as an identical constant for all degrees of freedom of the virtual system, but it could also be implemented in a component-wise manner, so with different mass decrease rates per virtual degree of freedom. This may be beneficial when working in a situation where both rotational and linear inertias are present. Reason for the change to a linear decrease is a finer control, which was found to work better with the method in section III. P_{min} is an additional condition implemented, with the purpose of preventing the virtual mass from being lowered when the operator is not providing enough power to the robot.

Added to this is another energy source for increasing the virtual mass. This source is the kinetic energy present in the virtual model. Ignoring the energy added or removed for the time being, the following could be said:

$$\mathbf{v}^T(k-1)\mathbf{M}(k-1)\mathbf{v}(k-1) = \mathbf{v}(k)^T\mathbf{M}(k)\mathbf{v}(k), \tag{28}$$

which, when the virtual mass is increased and on the assumption that the virtual mass matrix $\mathbf{M}$ is diagonal, can be reduced to:

$$v_i(k) = v_i(k-1)\sqrt{\frac{m_i(k-1)}{m_i(k-1) + \eta s_i}}, \tag{29}$$

with $\eta \in \{0, 1\}$ a constant determining what fraction of the mass increase should be compensated for by energy from the kinetic energy and subscript i the i'th component. To be absolutely clear, this is not supposed to replace the energy tank, even with $\eta = 1$. The disadvantage is that it does cause a discontinuity on the virtual velocity, which may be felt by the operator as a sudden change in velocity.

3) Energy-based safety measure: Because the interaction between operator and robot must be safe, a limit on the amount of energy stored in the robot could be used. Normally, one of the ways this can be done is by setting velocity bounds on the virtual model. However, seeing the method above employs variable virtual inertia, velocity bounds on the virtual model may not be sufficient to limit the amount of energy. Similar as (29), it is based on the assumption that only kinetic energy is stored in the virtual system, and that the matrix $\mathbf{M}$ is diagonal. Let E_{lim} be the maximum allowable energy stored in the virtual system. To satisfy $E_{\mathit{lim}} = H_v$, the energy removed from each virtual degree of freedom should be proportional to the ratio of that degree of freedom contributing to H_v. This results in the following:

$$\frac{1}{2}m_i v_{\text{i, corr}}^2 = \frac{1}{2}m_i v_i^2 - (H_v - E_{lim})\frac{\frac{1}{2}m_i v_i^2}{H_v}, \tag{30}$$

with m_i the i'th component of the mass matrix $\mathbf{M}$, v_i and $v_{i,\ corr}$ the corresponding elements of the vectors $\mathbf{v}_d$ and $\mathbf{v}_{corr}$, respectively. This can be simplified to:

$$\frac{1}{2}m_i v_{i,\ corr}^2 = \frac{E_{lim}\frac{1}{2}m_i v_i^2}{H_v}, \tag{31}$$

or even further:

$$v_{i,corr}(k) = \begin{cases} v_i(k)\sqrt{\frac{E_{lim}}{H_v(k)}} & \text{if } H_v(k) > E_{lim} \\ v_i(k) & \text{otherwise} \end{cases} \tag{32}$$

This safety measure also has the disadvantage of potentially causing a discontinuity on the virtual velocity vector, which may be noticeable by the operator.

III. Power-adaptive feed-forward

Where the previous section provides a method to guarantee a passive interaction, it does have a disadvantage. Where passivity is achieved, it is done so by increasing the virtual mass. Especially when the operator keeps grasping the device with a firm grasp, consciously or by reflex, the above system will keep the virtual mass high, to prevent the system from going active. One option to achieve more mass reduction could be adding more virtual damping, but this is not ideal. This section has the purpose of investigating a novel method to achieve mass reduction while keeping the advantages of the system described in the previous section. The addition of a perfect feed-forward component can increase the bandwidth of the motion controller and in doing so achieve passive interaction for mass-reduced admittance controllers [8]. However, this requires a correct model of the robot, which becomes difficult for multiple DoF robots, or robots with internal compliance. Adaptive feed-forward controllers have been extensively researched. These aim to reduce the tracking error of the feedback controller, which would, by extent, make the admittance controller passive. Most of these adaptive control systems require sufficient knowledge of the system, which could theoretically be derived from physics. In this section, another type of adaptive feed-forward controller is derived, based on difference in power exchanged. The schematic of this proposed controller can be found in Fig. 5.

The method itself is based on gradient descent, which is a first-order iterative optimization algorithm, originally proposed by Cauchy [13]. The idea is that if the power exchanged between the robot and operator, $P_{hr}(t)$, is the same as the power exchanged between the operator and virtual model $P_{hv}(t)$ at any time t, the systems are the same. It was established in [8] that the phase-lag of the feedback controller, caused by its finite bandwidth, is one of the causes of active behavior. This can be interpreted as power that was not injected at the correct time in the robot, leading to a power error $P_{err} = P_{hv} - P_{hr}$.

Effectively, we are attempting to minimize the following power-error function:

$$E_k(\boldsymbol{\theta}) = \frac{1}{2}P_{err,k}(\boldsymbol{\theta})^2 \tag{33}$$

in which $\boldsymbol{\theta}$ is the vector of the parameters in the feed-forward component and subscript k the sample time k. This function $P_{err,k}(\boldsymbol{\theta})$ is unknown, especially when the actual robot model is not (exactly) known. The gradient of this error function can be found by chain differentiation:

$$\nabla E_k(\boldsymbol{\theta}) = (\nabla P_{err,k}(\boldsymbol{\theta}))P_{err,k}(\boldsymbol{\theta}). \tag{34}$$

The first element is based on the measured difference in power:

$$P_{err,k}(\boldsymbol{\theta}) = P_{hv}(k) - P_{hr}(k). \tag{35}$$

The second element $P_{err,k}(\boldsymbol{\theta})$ is considered power that should have been injected by the motion controller. This error is assumed to be caused by incorrect parameters of the motion controller, specifically in the feed-forward component. On that assumption and with the knowledge that the feedforward only affects P_{hr}, it can be said that $P_{err}(\boldsymbol{\theta})$ is related to $-P_{ff}(\boldsymbol{\theta})$. More specifically, the approximation is made that $\nabla P_{err}(\boldsymbol{\theta})$ can be set equal to $-\nabla P_{ff}(\boldsymbol{\theta})$. This approximation will be reflected on later.

Let the force injected by the feed-forward component be a function of parameter vector $\boldsymbol{\theta}$, calculated desired angular acceleration $\ddot{\mathbf{q}}_d(k)$ and calculated desired angular velocity $\dot{\mathbf{q}}_d(k)$:

$$\mathbf{F}_{ff}(k) = f(\boldsymbol{\theta}, \ddot{\mathbf{q}}_d(k), \dot{\mathbf{q}}_d(k)). \tag{36}$$

Both $\ddot{\mathbf{q}}_d(k)$ and $\dot{\mathbf{q}}_d(k)$ are desired joint accelerations and velocities found from the desired end-effector motion. $\mathbf{F}_{ff}(k)$ is a row-vector with as many elements as actuators on the robot. The power injected at sample k can then be expressed as:

$$P_{ff}(k) = \mathbf{F}_{ff}^T(k)\dot{\mathbf{q}}(k). \tag{37}$$

From this, the gradient can be determined w.r.t each element of $\boldsymbol{\theta}$:

$$\frac{\partial P_{ff}(k)}{\partial \theta_i} = \sum_{j=1}^{M} \frac{\partial f_j(\boldsymbol{\theta}, \ddot{\mathbf{q}}_d(k), \dot{\mathbf{q}}_d(k))\dot{q}_j(k)}{\partial \theta_i}, \tag{38}$$

with $\dot{q}_j(k)$ the j'th component of the robots actuator velocity vector $\dot{\mathbf{q}}$ at sample instance k. Doing so for every component in the parameter vector $\boldsymbol{\theta}$ yields the following vector:

$$\nabla P_{ff}(k) = \begin{bmatrix} \frac{\partial P_{ff}(k)}{\partial \theta_1} \\ \frac{\partial P_{ff}(k)}{\partial \theta_2} \\ \dots \\ \frac{\partial P_{ff}(k)}{\partial \theta_N} \end{bmatrix} \tag{39}$$

It is possible to make a change to the parameter vector $\boldsymbol{\theta}$ every sample instance k. This is called stochastic gradient descent [14]. However, this would make the system vulnerable to noise, e.g via the readout of the robots actuator velocity vector $\dot{\mathbf{q}}$. Because of that, $\boldsymbol{\theta}$ is updated every N sample instances,

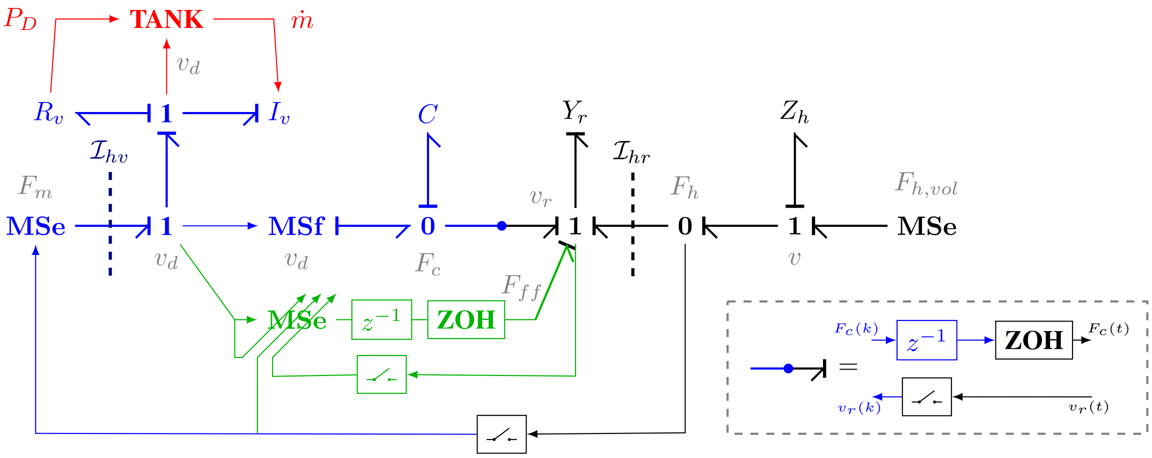

Fig. 5: Bond diagram of single DoF admittance controlled system adapted from [8] with permission. Similar to Fig. 4, with the addition of proposed adaptive feedforward shown in green. Modulated effort source (adaptive feedforward) creates a virtual force based on virtual velocity v_d, which is converted into a mechanical force F_{ff} via the delay and zero-order hold. This modulated effort source is adapted by a combination of v_d, v_r and F_h.

based on the information gathered between consecutive updates of $\boldsymbol{\theta}$. This is effectively mini-batch gradient descent. For one parameter in the vector $\boldsymbol{\theta}$, the update step is as follows:

$$\theta_i(k) = \theta_i(k-1) + \frac{1}{N}\sum_{n=0}^{N-1} \epsilon \nabla P_{\mathit{ff},i}(k-n) P_{err}(k-n), \quad (40)$$

where ϵ is the gradient descent step size and $\nabla P_{\mathit{ff},i}(k-n)$ the i'th component of ∇P_{ff} at sample time $k-n$. If both the robot and virtual system are sufficiently excited for long enough, the parameter vector $\boldsymbol{\theta}$ should have converged to a set of values which are optimized for the space in which the robot operates and the desired virtual dynamics.

A. Comparison with other method

To show the similarities between this method and an adaptive feedforward method based on tracking errors, an single DoF example will be considered. Let $C_{\mathit{ff}} = \theta \dot{v}_d$, which for the proposed new feedforward method would result in the following feedforward method, ignoring the mean over N samples:

$$\dot{\theta} = \dot{v}_d v F_h (v_d - v). \quad (41)$$

This would be rather similar to a tracking error-based adaption method proposed by Nakanishi and Schaal [15]. They assume an nth order SISO system of the form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ &\dots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= f(\mathbf{x}) + u, \end{aligned} \quad (42)$$

and that $f(\mathbf{x})$ can be represented as:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \theta + \Delta(\mathbf{x}), \quad (43)$$

with $\phi(\mathbf{x})$ a vector of basis functions, $\boldsymbol{\theta}$ the parameter vector and $\Delta(\mathbf{x})$ the approximation error. The update law proposed by Nakanishi and Schaal is:

$$\dot{\theta} = \Gamma \phi(\mathbf{x}_d)\mathbf{ce}, \quad (44)$$

with Γ is a positive adaptation gain. $\mathbf{ce}$ is given by

$$\mathbf{ce} = \Lambda_1 e + \Lambda_2 \dot{e}. \quad (45)$$

Comparing (41) to (44), and assuming that the used basis-function $\phi(\mathbf{x}_d)$ is equal to $\dot{v}_d$, it can be seen that if $\Lambda_1 = 0$, $\Lambda_2 = 1$ is used, the velocity errors are present in both equations. In that case, setting Γ equal to vF_h, yields the same equation, even though vF_h is time-varying.

B. Limits on parameter updates

Based on section III-A, as well as some experimental insights, some conditions are established that prevent (part of) $\boldsymbol{\theta}$ from being updated. These conditions are in place to prevent instabilities of the proposed feedforward system. From the tracking error method stability proof in [15] can be learned that, for the feedforward to be stable, Γ must be positive. From this follows that the product vF_h must also be positive. This yields a constraint on when the parameter vector θ is allowed to update. In other words, if vF_h is negative, then the system will not update. For a multi-body situation, this is implemented per actuator DoF. If for the i'th DoF holds $\tau_{h,i}(k)\dot{q}_i(k) < 0$, all parameters in the vector $\boldsymbol{\theta}$ that affect that actuator DoF will not consider that timestep in the update step in (40). In this, $\tau_{h,i}$ is the force/torque exerted on the i'th DoF by the operator due to applying a force on the end-effector. A deadband is implemented, which uses P_{avg}, the average power-error over the last N sample:

$$\theta_i(k) = \begin{cases} \theta_i(k-1) & \text{if } |P_{avg}| < P_{deadband} \\ \text{eq. 40} & \text{otherwise} \end{cases} \quad (46)$$

The purpose of this addition is to prevent discretization errors, sensor noise or small differences between the feedforward model and the actual robot from causing $\boldsymbol{\theta}$ to drift for very low power differences. It also prevents changes due to low power-errors that the feedforward may not be able to correct for, such as controllers with time-delay or errors. The last condition is that if either of the detection rules from section II is violated, the parameter vector will also not be updated.

IV. Experiment set-up

To test the proposed frameworks in section II and III simulations will be done. This section has the purpose of supplying the relevant information.

A. Model used

The model used in these simulations will be a two degrees of freedom robot. This model, seen in Fig. 6, is nonlinear and can be used to investigate the effects of non-linearity as well as the effect of attempting mass-reduction. Its equations of motion can be found in section A of the appendix. The equation of motion of the virtual model is given in (3), with the virtual model being attached to the end-effector. $\mathbf{F}_h$ is given by:

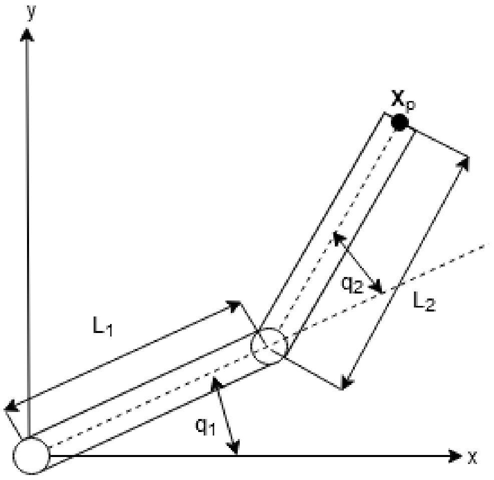

Fig. 6: Schematic representation of 2 Dof robot model

$$\mathbf{F}_h(t) = K_h(\mathbf{x}_{ref} - \mathbf{x}_p) + D_h(\mathbf{v}_{ref} - \mathbf{v}_p), \qquad (47)$$

where K_h and D_h are the gains of the model of the operator, $\mathbf{x}_p$ and $\mathbf{v}_p$ the position and velocity vectors of the end-effector and $\mathbf{x}_{ref}$ and $\dot{v}_{ref}$ the position and velocity of a reference position.

The values for the spring damper model of the operator was 1000 N/m stiffness and 1 Ns/m damping, respectively. The stiffness was based on a report by Höppner et al. [16], who determined that no subject could reach this stiffness. The damping value is not the maximum value that is possible for an operator, which is measured by Miller et al. [17] to be 0.09 Nms/rad for a wrist joint, or approximately 19 Ns/m. It can be seen in [8] that a larger amount of damping present in the human model helps with getting passive mass-reduction. The parameters used can be found in table I. It should be noted that the robot-configuration-dependent maximum virtual velocity used in section II has been simplified to v_{max} for all degrees of freedom.

B. Discrete virtual system

To reduce computation time, the virtual system is implemented as a discrete system, using forward Euler for both velocity and position calculations. Its equations are as follows:

TABLE I: Parameter values of two DoF system

Parameter	Value (unit)
M_1	5.0 (kg)
L_1	0.4 (m)
M_2	4.0 (kg)
L_2	0.3 (m)
K_p	50 (Nm s/rad)
K_i	600 (Nm / rad)
B_r	0.2 (Nm s/rad)
$\mathbf{R}_d$	diag$\{3.0, 3.0\}$ (N s/m kg)
K_h	1000 (N/m)
D_h	1.0 (Ns / m)
$\mathbf{M}_{v,0}$	0.1 (kg)
v_{max}	20 (m/s)

$$\mathbf{a}_d(k) = \mathbf{M}(k)^{-1}(\mathbf{F}_h(k) - \mathbf{B}_v(k)\mathbf{v}_d(k-1)), \qquad (48)$$

$$\mathbf{v}_d(k) = \mathbf{v}_d(k-1) + \mathbf{a}_d(k)T_s, \qquad (49)$$

$$\mathbf{x}_d(k) = \mathbf{x}_d(k-1) + \mathbf{v}_d(k)T_s + \mathbf{a}_d(k)T_s^2, \qquad (50)$$

with T_s the sample time.

C. Simulation setup

To illustrate both methods in sections II and III, both are applied to the 2 DoF manipulator model as described in section IV-A. In the first simulation, only the passivity preserving framework of section II is used. In the second simulation, the passivity preserving framework with adaptive feedforward is used. The dynamics are simulated using the ODE45 function provided by MATLAB 2020A, with the default settings. The admittance controller is run at 3 kHz, with the passivity-preserving functionality run at 150 Hz. The admittance controller is implemented with one sample time delay. The parameters of the passivity preserving method can be found in table II.

For the simulation with both methods implemented, all parameters of the passivity preserving method are the same as the ones described in table II, with the exception of the mass-loss constant, which for this simulation has been increased to $M_{loss} = 1$ g. This will result in a maximum mass loss rate of 0.15 kg / s. There are two modifications to the virtual

TABLE II: Parameter values for passivity preserving framework

Parameter	Value (unit)
M_{loss}	0.2 (g)
P_{min}	0 (J/s)
λ	0.005 ()
ϕ	0.5 ()
η	0.5 ()
$\overline{m}$	3.33 (g)
T_{max}	5.0 (J)
T_{min}	0.5 (J)
$T(0)$	2 (J)
E_{lim}	0.2 (J)
ψ_r	0 ()
T_p	10 (s)

TABLE III: Parameter values of adaptive feed-forward

Parameter	Value (unit)
ϵ	0.025 ()
N	125 ()
$P_{deadband}$	0.5 (mW)

model in this simulation, which is that $M_{v,\,0}$ is decreased to 10 g, and that $\mathbf{M}(0) = \text{diag}\{2.2\text{ kg}, 2.2\text{ kg}\}$. The reason for both changes will be discussed with the results. The following feedforward model has been chosen as:

$$\mathbf{F}_{ff}(k) = \mathbf{I}_d\ddot{\mathbf{q}}_d = \begin{bmatrix}\theta_1 & \theta_2\\ \theta_3 & \theta_4\end{bmatrix}\begin{bmatrix}\ddot{q}_{d,1}(k)\\ \ddot{q}_{d,2}(k)\end{bmatrix} \tag{51}$$

with $\ddot{q}_{d,1}(k)$ and $\ddot{q}_{d,2}(k)$ the desired accelerations of the first and second actuator, found from the desired end-effector motion. This yields the following vector ∇P_{ff}:

$$\nabla P_{ff}(k) = \begin{bmatrix}\ddot{q}_{d,1}(k)\dot{q}_1\\ \ddot{q}_{d,2}(k)\dot{q}_1\\ \ddot{q}_{d,1}(k)\dot{q}_2\\ \ddot{q}_{d,2}(k)\dot{q}_2\end{bmatrix}. \tag{52}$$

The parameters used in the feedforward model can be found in table III. In both simulations, the operator moves the end-effector around in a periodic manner:

$$\mathbf{x}_{ref} = \begin{bmatrix}-0.007 + 0.05\sin(0.5\pi t)\\ 0.361 + 0.05\sin(\pi t)\end{bmatrix}, \tag{53}$$

which creates a reference trajectory that resembles an infinity symbol. Its center was chosen such that it was far away from any singularity in the robot jacobian $J(q)$.

V. Results

The results of the simulation with only the passivity preserving framework can be seen in Fig. 7 and 8. In section B it is determined that the effective end-effector mass of the robot is between $5.5 - 6.8$ kg in one direction and $0.88 - 0.97$ kg on the other direction.

The results for the simulation with passivity preserving framework and feedforward can be seen in Fig. 9, 10 and 11. The end-effector masses for this simulation are similar to the previous one. This is to be expected, seeing as the reference trajectory did not change between simulations.

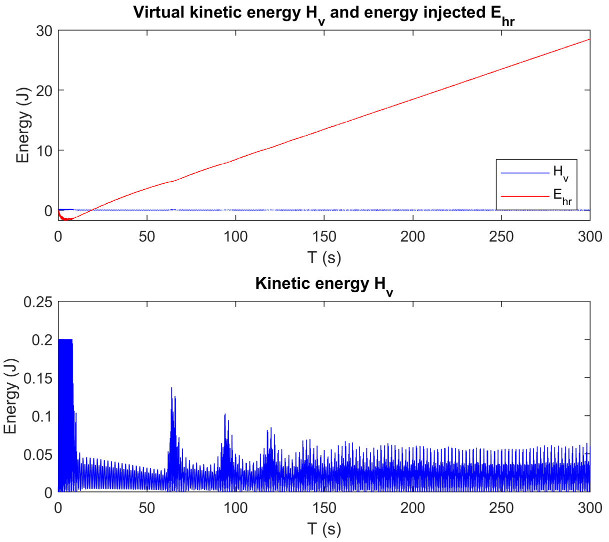

Fig. 8: Kinetic energy in virtual model H_v and energy inserted by operator E_{hr} over time. Initial active behavior is due to too low virtual mass. Kinetic energy H_v plotted separately, showing kinetic energy being limited by (32) in the first 10 seconds. Peaks in H_v after 40 seconds correspond to moments where (23) was violated.

VI. Discussion and conclusion

A few things can be noted from the simulation with only the passivity preserving framework. The first is that because the robot is instantiated with a large amount of mass-reduction, it immediately becomes active. However, passivity according to (19) is restored and maintained after twenty seconds. The second thing is that three minutes after initialization the virtual mass has more or less settled on something that can be considered an equilibrium, where the virtual mass changes at most 75 grams, which with an average mass of 2.18 kg is barely noticeable by an operator. It was in principle possible the system could have been instantiated with a mass-increment already present, e.g. with an $\Delta M(0) = \text{diag}\{2\text{kg}, 2\text{kg}\}$, to avoid the initial activity. However, it was chosen not to do so to show all features in the proposed framework. It should be noted that the initial activity could also be resolved more quickly by making $\overline{m}$ larger, but that comes at the cost of larger variations in the phase where the virtual mass has more or less settled.

Fig. 9 shows virtual mass over time for the simulation with passivity preserving framework and feedforward. In this, it can

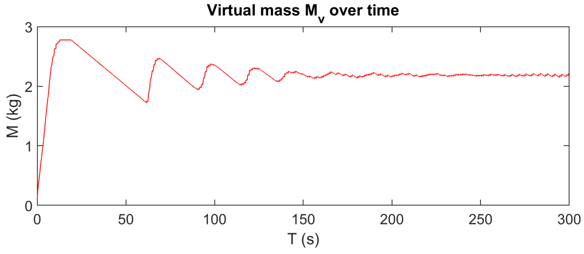

Fig. 7: Virtual mass $\mathbf{M}$ evolution over time. Initially. it increases linearly to restore passivity. After this, it approaches an equilibrium.

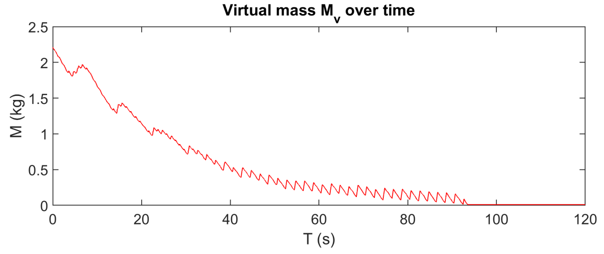

Fig. 9: virtual mass over time. Started with an offset to avoid initial active behavior, decreased back to desired value after 100 seconds.

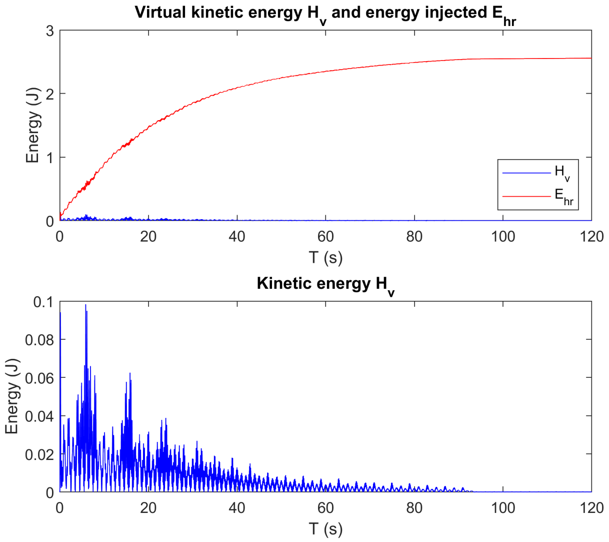

Fig. 10: Kinetic energy in virtual model H_v and energy inserted by operator E_{hr} over time. Initial active behavior is due too low virtual mass. Kinetic energy H_v plotted separately. Peaks in H_v correspond to moments where (23) was violated.

be seen that after 100 seconds, the virtual mass has reached the desired value. The passivity preserving method did have to add mass quite often, most likely because the rate at which the feedforward adapts was not as high as the rate at which virtual mass was lost. With the robots end-effector mass at most being around 6.8 kg, see section B of the appendix, it can be said that effectively a mass reduction of up to 680 is reached, at the expense of some virtual damping.

As for the changes compared to the other simulation, the reason for the increase in mass loss rate to $M_{loss} = 1$ g is to investigate the speed with which large amounts of mass reduction can be achieved. $M_{v,0}$ was reduced to explore the limits of the framework with adaptive feedforward. $\Delta\mathbf{M}(0)$ was instantiated at 2.2 kg to avoid the initial active behavior seen in Fig. 8, using the equilibrium seen in Fig. 7.

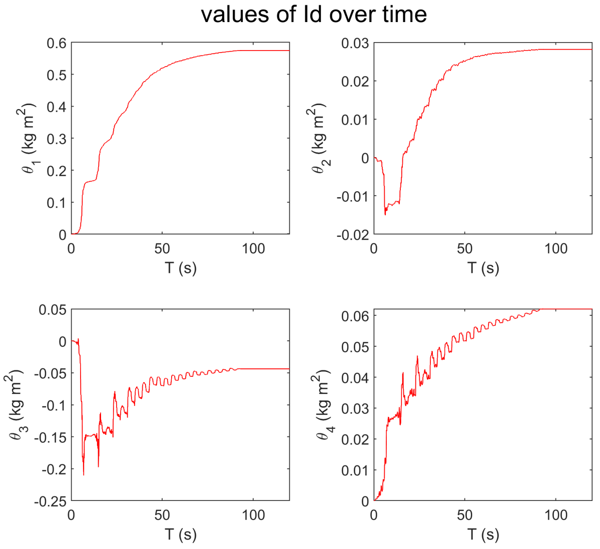

Fig. 11: Values of θ over time. Some fluctuations within the generally asymptotic approach can be observed in all four graphs.

A. *Remarks*

The arguably largest issue is that no proof of stability is given in this report for the adaptive feedforward. The connection is made to an existing method based on tracking errors [15]. However, the new proposed feedforward is different with the time-varying nature of vF_h as well as the method of implementation of the related constraint for a multi-actuator system. Because of this, the claim cannot be made that the proof of stability used for the existing method also works for the proposed feedforward.

B. *Conclusion*

The presented framework for preserving passivity has indeed shown that it can preserve passivity in simulation. In this framework, only the virtual model and by extent the virtual velocity is modified, so in principle it should work with black box robot control, where only velocity or position setpoints are used, provided that the energy exchange between operator and robot can be measured accurately. The proposed adaptive feed-forward has shown to work as well for this simulation, increasing the apparent admittance further. However, the author believes that its mathematical support should be expanded further before applying it in a real setup or having it interact with humans.

Appendix

A. *Equations of motion for 2Dof Model*

The two arms of the robot are modeled as point-masses located at half of the length of their respective arms. Doing so will ignore any effects caused by the rotational inertia, however, the model will still have non-linearities. The model of the robot is:

$$\boldsymbol{\tau}_{mc}(t) + J(\mathbf{q})^T\mathbf{F}_h(t) = \mathbf{I}_r(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}\dot{\mathbf{q}}, \quad (54)$$

where $\boldsymbol{\tau}_{mc}(t)$ is the torque exerted by the motion controller, $J(\mathbf{q})^T\mathbf{F}_h(t)$ is the torque exerted on the joints by the operator. $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the angular position, velocity and acceleration vectors, respectively. Before defining $\mathbf{I}(\mathbf{q})$, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ and $\mathbf{D}$, some intermediary variables must be declared:

$$\alpha = M_1(L_1/2)^2 + M_2(L_1^2 + (L_2/2)^2), \quad (55)$$
$$\beta = M_2L_1L_2/2, \quad (56)$$
$$\delta = M_2(L_2/2)^2, \quad (57)$$

with M_1, M_2 the mass of the first and second arm, respectively. L_1 and L_2 are the lengths of the first and second arm, respectively. Now, $\mathbf{I}(\mathbf{q})$ can be defined:

$$\mathbf{I}_r(\mathbf{q}) = \begin{bmatrix} \alpha + 2\beta\cos(q_2) & \delta + \beta\cos(q_2) \\ \delta + \beta\cos(q_2) & \delta \end{bmatrix}. \quad (58)$$

Also $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ can be defined:

$$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -\beta \sin(q_2)\dot{q}_2 & -\beta \sin(q_2)(\dot{q}_1 + \dot{q}_2) \\ \beta \sin(q_2)\dot{q}_2 & 0 \end{bmatrix}. \quad (59)$$

Lastly, $\mathbf{D}$ is the matrix representing the damping in the system, chosen to be modeled as viscous damping on the joints:

$$\mathbf{D} = \begin{bmatrix} 2B_r, -Br \\ -Br, Br \end{bmatrix} \quad (60)$$

The controller is a PI velocity controller:

$$\boldsymbol{\tau}_{mc}(t) = K_p(\dot{\mathbf{q}}_d - \dot{\mathbf{q}}) + K_i(\mathbf{q}_d - \mathbf{q}), \quad (61)$$

with K_p, K_i the proportional and integral gains. $\mathbf{q}$ and $\dot{\mathbf{q}}$ are the angular position and velocity vectors of the robot. $\mathbf{q}_d$ and $\dot{\mathbf{q}}_d$ are the desired angle and rotational speed vectors, which are derived from the virtual model velocity $\mathbf{v}_d$ and the jacobian $J(\mathbf{q})$.

B. End effector operational mass

It is useful to know how much weight is felt when moving the end-effector. For this the eigenvalues of the matrix $J(q)^{-T}\mathbf{I}_r J(q)^{-1}$ can be used, $\mathbf{I}_r$ the mass matrix of the robot. Fig. 12 shows the eigenvalues of this matrix as a function of time.

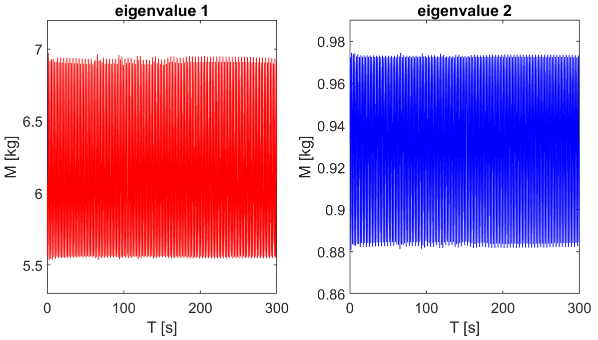

Fig. 12: Operational space mass of the end-effector.

References

[1] Z. Li, B. Huang, Z. Ye, M. Deng, and C. Yang, "Physical human–robot interaction of a robotic exoskeleton by admittance control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9614–9624, 2018.

[2] Weiguang Huo, J. Huang, Yongji Wang, Jun Wu, and Lei Cheng, "Control of upper-limb power-assist exoskeleton based on motion intention recognition," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 2243–2248.

[3] J. E. Colgate, M. Peshkin, and S. H. Klostermeyer, "Intelligent assist devices in industrial applications: a review," in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 3, 2003, pp. 2516–2521 vol.3.

[4] A. Campeau-Lecours, B. Mayer-St-Onge, and C. Gosselin, "Variable admittance control of a four-degree-of-freedom intelligent assist device," pp. 3903–3908, 05 2012.

[5] J. Colgate, "The control of dynamically interacting systems," *PhD thesis*, 01 1988.

[6] N. Hogan, "Controlling impedance at the man/machine interface," in *Proceedings, 1989 International Conference on Robotics and Automation*, May 1989.

[7] J. E. Colgate, "The control of dynamically interacting systems," *Massachusetts Institute of Technology*, 1988.

[8] A. Keemink, "Haptic physical human assistance, chap. 4&5," Ph.D. dissertation, University of Twente, 10 2017.

[9] M. Nabeel, JaeJun Lee, U. Mehmood, A. Jafari, Jung-Hoon Hwang, and J. Ryu, "Increasing the impedance range of admittance-type haptic interfaces by using time domain passivity approach," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 585–590.

[10] B. Hannaford and Jee-Hwan Ryu, "Time-domain passivity control of haptic interfaces," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 1–10, 2002.

[11] F. Ferraguti and C. T. L. et al., "A variable admittance control strategy for stable physical human–robot interaction," *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 747–765, 2019.

[12] D. Lee and K. Huang, "Passive-set-position-modulation framework for interactive robotic systems," *Robotics, IEEE Transactions on*, vol. 26, pp. 354 – 369, 05 2010.

[13] A. Cauchy, "Méthode générale pour la résolution des systèmes d'équations simultanées," *C. R. Acad. Sci. Paris*, pp. 25, 536–538, 1847.

[14] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.

[15] J. Nakanishi and S. Schaal, "Feedback error learning and nonlinear adaptive control," *Neural Netw.*, vol. 17, no. 10, p. 1453–1465, Dec. 2004.

[16] H. Höppner, M. Große-Dunker, G. Stillfried, J. Bayer, and P. van der Smagt, "Key insights into hand biomechanics: Human grip stiffness can be decoupled from force by cocontraction and predicted from electromyography," *Frontiers in Neurorobotics*, vol. 11, 05 2017.

[17] T. E. Milner and C. Cloutier, "Damping of the wrist joint during voluntary movement," *Experimental Brain Research*, vol. 122, pp. 309–317, 1998.

Chapter 3

Thesis discussion

This chapter has the purpose of discussing elements that have not been discussed in section VI or to support claims that are made in the previous sections. It also goes into detail on the limitations of the framework and on future work that could be done.

3.1 Effective mass of end-effector

In section VI, the claim is made that because some virtual mass equilibrium is reached, that virtual mass is the effective mass felt on the end-effector. This section has the purpose of investigating this claim. To do this, the data gathered in section V, specifically the end-effectors velocity and acceleration and the forces exerted by the operator, is used to identify a linear mass-damper system located on the end-effector.

For this, the following system is assumed to be identified:

$$\mathbf{F}_h(k) = \mathbf{M}_p\ddot{\mathbf{x}}_p(k) + \mathbf{D}_p\dot{\mathbf{x}}_p(k) = \begin{bmatrix} M_x & 0 \\ 0 & M_y \end{bmatrix} \ddot{\mathbf{x}}_p(k) + \begin{bmatrix} D_x & 0 \\ 0 & D_y \end{bmatrix} \dot{\mathbf{x}}_p(k). \tag{62}$$

For the system-identification the linear least-square estimate is used of the function $\mathbf{G}(\boldsymbol{\phi}) = \boldsymbol{\phi}^T\boldsymbol{\theta}$, such that an estimate can be found: $\hat{\boldsymbol{\theta}} = \boldsymbol{\phi}^{\dagger}\mathbf{G}(\boldsymbol{\phi})$ [7]. For one time instance k this can be written for this situation as:

$$\mathbf{G}(\boldsymbol{\phi}) = \mathbf{F}_h(k) = \begin{bmatrix} \ddot{x}_{p,x}(k) & \dot{x}_{p,x}(k) & 0 & 0 \\ 0 & 0 & \ddot{x}_{p,y}(k) & \dot{x}_{p,y}(k) \end{bmatrix} \begin{bmatrix} M_x \\ D_x \\ M_y \\ D_y \end{bmatrix} = \boldsymbol{\phi}^T\boldsymbol{\theta}. \tag{63}$$

Both $\mathbf{G}(\boldsymbol{\phi})$ and $\boldsymbol{\phi}^T$ need to be expanded with more rows to determine the estimate $\hat{\boldsymbol{\theta}}$ to any degree of accuracy. This requires the use of multiple time instances. This might somewhat break the assumption in (62) that the parameters to be identified are time-invariant, as the virtual dynamics are certainly time-varying. This means that the identified plant is the least-square estimate over both the time samples used and the potentially time-varying plant variables. This means that there is an error associated with this, mostly related to the time-variation of the virtual dynamics. To guarantee that the matrix $\boldsymbol{\phi}^T$ and as a consequence $\boldsymbol{\phi}^\dagger$, contains sufficient information, the parameters in (62) are estimated over a one-second window, so using 3000 measurements. For this window, it is estimated that for situations where the maximum amount of virtual mass (0.5 kg/s) is added over this window, this error margin could be as high as 0.25 kg. For situations where the maximum amount of mass is lost in this window, this may be 0.015 kg for the passivity preserver alone and 0.075 kg for the passivity preserver with adaptive feedforward.

The procedure described above is for one single set of estimated parameters. To obtain the dynamical properties of the estimated system over the entire duration of the experiment, the procedure is repeated in different points of the simulation.

The identified plant for the first simulation can be found in Fig. 13. It can be seen that the identified mass on both axes is slightly lower than the virtual mass, particularly during the steady-state phase of the simulation. Something that gives some insight into the operation of the passivity preserving framework can be seen in the right plot of Fig. 13. Where the identified damping on the y-axis is only slightly lower than the virtual damping, the identified damping on the x-axis is oscillating close to zero. This is an indication that the virtual damping on that axis is used to dissipate the energy generated due to the difference in admittance between the virtual model and robot on that axis. In doing so, that virtual damping is not felt by the operator on the end-effector, resulting in almost no damping on the x-axis.

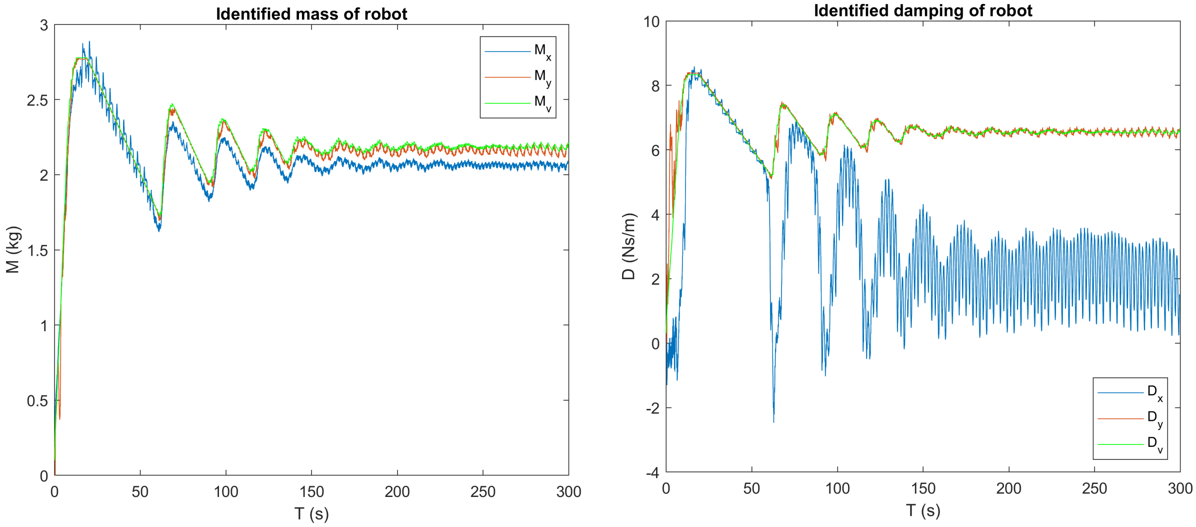

Figure 13: Left: identified M_x and M_y over time for the first simulation. Right: identified D_x and D_y over time for the first simulation. In both, virtual parameters are given in green. Identified negative damping is due to energy, generated by difference in mass between robot and virtual model, being returned to the operator.

The identified plant for the simulation with adaptive feedforward can be seen in Fig. 14. From this can be seen that the effective end-effector mass is almost similar to the virtual mass throughout the entire simulation. Similar to Fig. 13, the virtual damping on the x-axis is much lower than on the y-axis and oscillates around zeros, indicating that the passivity preserving framework again moves towards the lowest virtual mass for which it is still passive. However, now with the adaptive feedforward adapting, over time virtual mass and effective end-effector mass on both axes converge towards the desired 10 g, although on the x-axis the effective mass still varies slightly, at maximum 0.5 gram. This is behavior that is no longer being compensated for by the adaptive feedforward because it falls within the deadband.

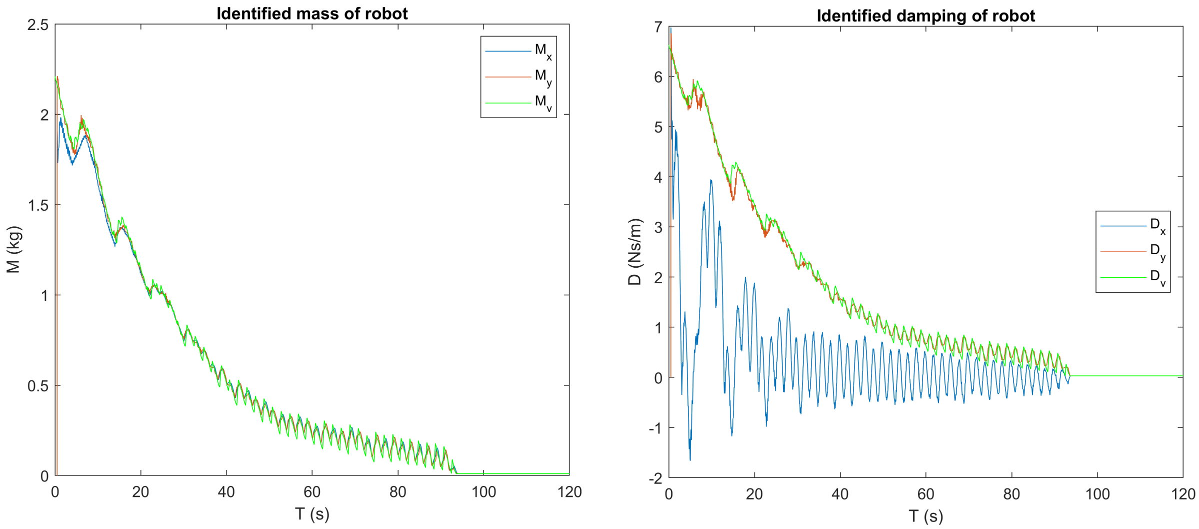

Figure 14: Left: identified M_x and M_y over time for the first simulation. Right: identified D_x and D_y over time for the first simulation. In both, virtual parameters are given in green.In both, virtual parameters are given in green. Identified negative damping is due to energy, generated by difference in mass between robot and virtual model, being returned to the operator.

It can be seen that at times in both simulations the identified damping is negative at times. It was already explained that the virtual damping can be used to dissipate the energy generated by the difference in admittance between robot and virtual model on that axis. At these points the virtual damping is not sufficient to dissipate this energy, resulting in energy returned to the operator. This is identified in the plant as negative damping. It can be seen that the passivity preserving framework generally increases the virtual mass shortly after the virtual damping goes below zero. However, there are also points where the identified damping goes below zero where the virtual mass does not increase as a result. In other words, energy from the difference in mass between robot and virtual mass on that axis is returned in a way that would violate the detection rules, had those been enforced on that degree of freedom separately. However, it doesn't, as more energy is dissipated at the same time on the other axis, keeping the overall interaction passive. It is also because of this that the kinetic energy H_v in Fig. 8 is higher than anticipated based on the reference trajectory and the virtual mass.

3.2 Remarks on parameters

With the number of parameters as used in the passivity preserving framework in section II, as well as the few used in section III, it may be useful to reflect on these parameters, to clarify how some of them were chosen and what potential trade-offs between parameters exist.

A ϕ and η

The parameter ϕ determines how much of the energy dissipated by reducing the virtual mass is stored in the energy tank. In this report, this is chosen 0.5 arbitrarily, mostly because the amount of virtual damping used and sufficiently high virtual velocities guaranteed that the energy tank would be filled regardless of the chosen ϕ. It should be chosen higher when less virtual damping or lower virtual velocities are used. The last remark regarding this mechanism is that in both this report and the paper by Ferraguti et al. [6] there is an oversight in that it is possible to fill the energy tank beyond T_{max}. (10) does not take into account the current energy level of the tank, so it will always add energy to the tank when the mass is reduced. The effect of this on the simulation in this report was considered negligible, but it should be fixed regardless. As for η, the choice here entirely depends on design choices. It is possible to entirely rely on virtual damping and energy recovery from mass reduction, in which case it can be set to zero. Another reason to do this is if sudden velocity changes on the virtual model and via the motion-controller also on the real robot are considered completely unacceptable. A reason to set it closer to one could be that the virtual damping or virtual velocities do not generate enough energy to increase the virtual mass at a sufficient rate.

B $\overline{m}$

This parameter is one of the upper limits that determine the maximum amount of virtual mass added to each degree of freedom when one of the detection rules is violated. For the simulations in section IV, this is set to 0.0033 kg, which could result in a maximum virtual mass increase rate of 0.50 kg/s. Again, this is a design choice, as setting it higher could help with e.g. fixing the initial active behavior as seen in Fig. 8 or providing a quicker response to increases in stiffness of the operator. However, setting it higher will also result in larger mass variations during the steady-state phase that can be seen in Fig. 7, which may be felt by the operator.

C v_{max}

The simplification that the robot-configuration-dependent maximum virtual velocity could be set to v_{max} for all degrees of freedom was mainly because everything is done in simulation. The robot-configuration-dependent maximum virtual velocity would normally be determined by hardware limitations on the robot. For the used simulations this simplification turned out to not have too much influence, as the limit for the mass increase in (16) in almost all cases was $\overline{m}$, not the energy-based limit.

D M_{loss} vs. learning rate ϵ and N samples

The learning rate ϵ, used in the feedforward, is one of the factors that determine the rate at which the feedforward adapts. There is some relation between ϵ and M_{loss}, in that increasing M_{loss} without increasing ϵ as well may lead to more mass variations by the passivity preserving framework. This could already be seen in Fig. 9, but it can be made more clear with another example, which can be seen in section 4.1 in the appendix. In this, the learning rate ϵ has been reduced to 0.01, where all other parameters are kept as described in section IV. It can be seen that where the simulation in section V converged in 100 seconds, the simulation in section 4.1 is still converging after 120 seconds. In this case, an higher ϵ is better, but it should be noted that ϵ has an upper limit, above which convergence cannot be guaranteed. Alternatively, lowering N could be considered when increasing M_{loss}, because it allows for more adaptations of the feedforward per second. The disadvantage of this would be an increased susceptibility of the feedforward to sensor noise.

E Frequency of passivity preserving framework

This frequency, set to 150 Hz for both simulations, determines how many times per second is checked whether the detection rules are violated and the number of increases or decreases of the virtual mass. This results in discrete moments in time where is checked whether passivity is preserved. In theory, it is possible for the system to violate passivity for a very short time, but to be passive again at the next passivity check. This is quite unlikely, but the change of this happening can be further decreased by increasing the frequency. The second thing is that it is possible that the passivity is lost shortly after a check on the passivity is performed, and that until the next check the system is behaving actively. This is unavoidable due to the discrete nature of the controller, but the amount of energy returned in that case can be minimized by increasing the frequency, thus decreasing the sample-time. The only disadvantage of increasing the frequency is the extra computational costs.

3.3 Energy exchange estimation

Estimating the amount of energy exchanged between the operator and the robot is not trivial due to the time-varying nature of both the force exerted by the operator and the position and velocity of the end-effector, as well as the fact that the controller is implemented in discrete time. This section has the purpose of showing a few different integration methods of estimating the energy exchanged, as well as accounting for which one was chosen in the simulations.

Five different energy exchange estimation methods are considered in this section:

1. $E_1(k) = E_1(k-1) + \mathbf{F}_h^T(k)(\mathbf{x}_p(k) - \mathbf{x}_p(k-1))$
2. $E_2(k) = E_2(k-1) + \mathbf{F}_h^T(k-1)(\mathbf{x}_p(k) - \mathbf{x}_p(k-1))$
3. $E_3(k) = E_3(k-1) + \mathbf{F}_h^T(k)\mathbf{v}_p(k)Ts$
4. $E_4(k) = E_4(k-1) + \mathbf{F}_h^T(k)\mathbf{v}_p(k-1)Ts$
5. $E_5(k) = E_5(k-1) + \mathbf{F}_h^T(k-1)\mathbf{v}_p(k)Ts$

The approach used in this report was to pick one of the options from the list above and perform the simulations as described in section IV. Then, using the data obtained, the exchanged energy is calculated using each estimation method. While this is not entirely correct, seeing as the frameworks presented in sections II and III will be influenced by the choice in estimation method, it is a check whether the used estimation method is the best one to use. Best in this case means that it does not overestimate the amount of energy exchanged while remaining accurate. If another estimation method other than the one used calculates a lower amount of energy exchanged, the case can be made that in reality less energy was exchanged. In the very worst case, that could lead to loss of passivity in reality, while the controller thinks it is behaving passively.

As an example, the exchanged energy calculated from the data from the simulation with passivity preserving functionality can be seen in Fig. 15. Three things must be noted from the results. The first is that there is a 1.8 Joule between integration schemes by the end of the simulation, showing that the choice in estimation method is not trivial. Secondly, methods E_3 and E_4 are similar, which is to be expected seeing as for very small T_s $\mathbf{v}(k)$ is similar to $\mathbf{v}(k-1)$. However, most important is that, even though E_1 is the most conservative estimation method, the actual energy curve goes below the used estimation method in the simulation. For reference, the actual energy is measured

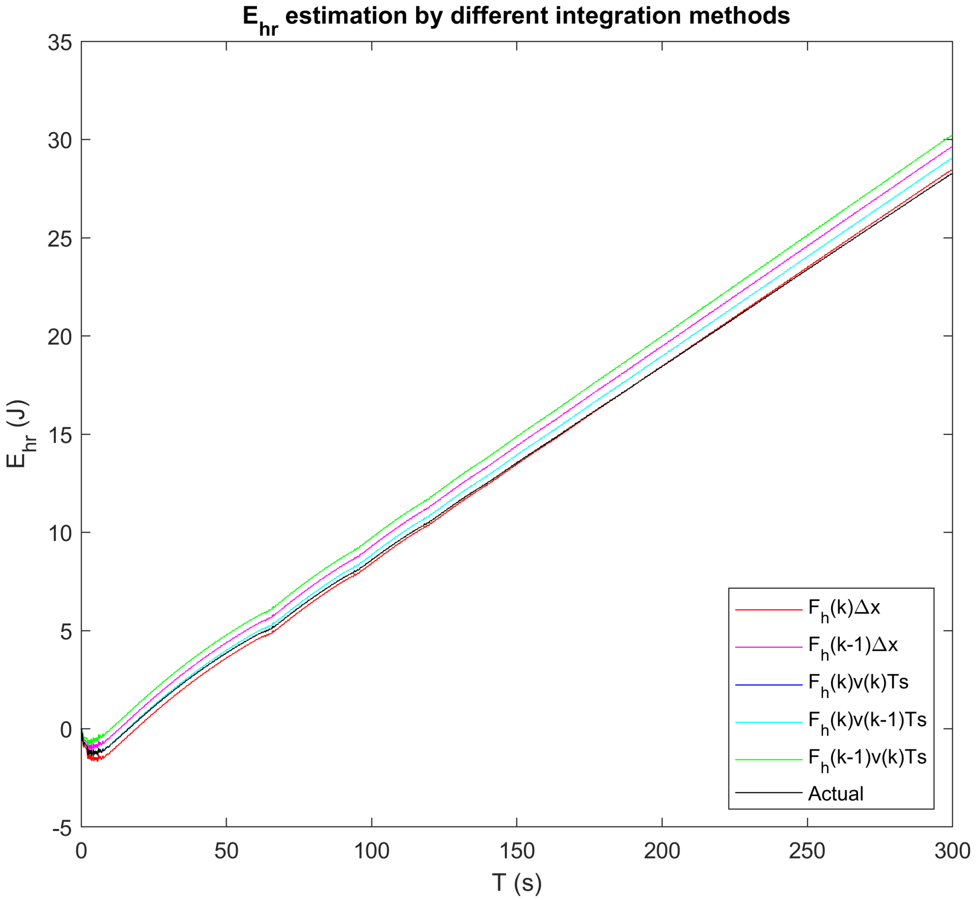

Figure 15: Estimated energy exchanged between robot and operator with different integration schemes. Used integration scheme E_1 in this simulation is shown to be the most conservative estimate but still overestimates the energy exchanged after 200 s.

is by adding it as an additional state in the dynamics. Although the claim still stands that the overall interaction is strictly passive, it does indicate a potential problem and highlights the need for a reliable and accurate exchange estimation. An alternative would have been to calculate the estimated exchanged energy at every time-step using method E_1 trough E_5, and using the lowest value calculated. This would per definition result in the most conservative estimate possible. The reason it was not used in this report is that the energy exchanged was very similar to the estimate by E_1, so for computational reasons it was not used. Other more elaborate energy exchange estimations could also have been used, such as the trapezoidal integration method suggested by Franken [8].

3.4 Limitations of proposed framework

The passivity preserving framework as shown in section II has shown to be able to make the interaction passive. However, although it does not require knowledge of the operator, the virtual mass is affected by the operator impedance. More specifically, for a linear spring-damper model representing the operator, if the stiffness is high, the virtual mass is also high. As an example, reducing the stiffness K_h to 500 N/m yielded an equilibrium of 1.25 kg for the same simulation without feedforward. This would indicate that the achievable mass reduction by the framework is in part dependant on the operator's properties. Increasing the virtual damping will increase the achievable mass reduction further, but may reduce the overall apparent admittance. This would be the case for example on the y-axis in the simulation, seen in figure 13, where the virtual damping is felt by the operator on that axis.

As already noted in section VI, no proof of stability is given for the adaptive feedforward. Although the similarities are noted for a single degree of freedom example when comparing to an existing method based on tracking errors [9], the assumptions used to compare the proposed feedforward to this method mean the proof for stability cannot be immediately used. One additional reason for this is the implementation of the constraint per actuator degree of freedom that $\tau_{h,i}(k)\dot{q}_i(k) > 0$ must hold. It may be an implementation that holds for the simulation with feedforward in section V, but that does not mean that it holds for all systems with multiple degrees of freedom.

Something noted in section 3.1 is that enforcing strict passivity on the overall interaction has a disadvantage in that it is possible to return energy to the operator on one axis that would violate strict passivity. But as long as more energy is dissipated on another axis at the same time, the overall interaction is passive. This behavior is made possible by the presence of (excessive) virtual damping. This behavior may not be desirable, and the recommendation to solve this is to have the detection rules function per virtual degree of freedom and corresponding end-effector degree of freedom. This will increase the average virtual mass during the steady-state phase somewhat but removes the disadvantage of limited active behavior.

Both the passivity preserving functionality and the adaptive feedforward have not been tested on real systems with real humans, due to COVID19 and time restrictions. Although the simulations were made as close as possible to a real-world scenario, not everything is accounted for. Examples of this would be measurement quantization, noise on the sensors, imperfect actuators, non-linear damping or post-sensor dynamics. Some of these could potentially affect the accuracy of the energy estimation between operator and robot. It may be possible to account for quantization by taking it into account when calculating the energy exchanged, such as in [10], sec. 2.2.1.

The passivity preserving functionality is not expected to be affected by high-frequent noise on force sensors, seeing as the force is integrated at least once when calculating the exchanged energy or the virtual velocity. The adaptive feed-forward will be more directly affected by noise on the sensors, for instance via the calculated virtual model acceleration or via the estimated actuator velocities. To a degree, the effect of this will be reduced because the adaptation step uses the average over N measurements. Increasing N may help in dealing with noise, although that will reduce the rate at which the system learns, due to having fewer feedforward adaptations per second.

The proposed framework has not been extensively tested with time-varying or non-linear human models. Some simulations were performed where the human stiffness K_h was increased during the simulation, analogous to the operator grasping the device firmer. In other simulations, the human damping D_h was initially high and suddenly reduced significantly. During these simulations, the framework without feedforward was able to maintain a passive interaction.

3.5 Future work

All simulations in this report use a constant mass-to-damping ratio R_d. Effectively, this will add variable damping to the system. Two things must be remarked about R_d. The first is that increasing R_d has a positive influence on the rate at which the adaptive feedforward converges. This is possibly due to the condition of the feedforward that vF_h must be positive, which happens more often when there is more damping present in the system. However, the second thing to note is that adding excessive amounts of damping to the virtual system will lower the admittance of the overall system. It should however be possible to reduce the mass-to-damping ratio R_d, e.g. once the virtual mass has reached the desired value. Doing so would increase the apparent admittance of the system further. Some simulations were performed to test this, and the results were positive, but due to time constraints, a thorough analysis and implementation could not be done. In theory, it is also possible to disconnect the virtual damping from the mass entirely. This would mean that the method in section II would still operate as designed, but the virtual damping could be chosen to not be a function of time or virtual mass. It is however not advised to set the virtual damping to zero, as the energy tank as currently implemented depends on it to extract energy and it can be used to dissipate away energy generated by active behavior.

It is expected that the feedforward model, (51), would not have been as useful in a scenario where the reference trajectory contained a much larger part of the workspace. This is simply because for the used reference trajectory, (53), the mass matrix of the robot $\mathbf{I}_r$ does not change as much, so the feedforward model could be fitted to a sufficient degree to this situation. In a scenario where much more of the workspace of the robot is used, $\mathbf{I}_r$ is expected to change much more, so a fit of the feedforward with only constants in the matrix I_d may not be sufficient for constant high mass-reduction, resulting in virtual mass variations to keep the overall interaction passive. However, this feedforward model used in the simulation is not the only one that can be used in this method. If more knowledge is available on the system, that knowledge can be used to create a more fitting feed-forward model, for instance by also compensating for the nonlinear dynamics, or by assuming non-linearity in the mass-matrix.

When discussing multiple degrees of freedom, it is important to note that in the proposed feedforward, the power-error is a single number that the feedforward attempts to adapt for and drive to zero. It may be beneficial to split this power-error up in power-errors per actuator degree of freedom. This yields for the i'th actuator an equation of the form $P_{err,i} = \tau_{h,i}(k)(\dot{q}_{d,i}(k) - \dot{q}_i(k))$, and a corresponding change to (40). Doing so would result in a more targeted and potentially quicker adaptation.

The energy tank component was copied in its entirety from Ferraguti et al. However, this method has some drawbacks, such as that z should never become zero and that a time integration is involved. It should, in theory, be possible to use different implementations of energy-tanks, such as the one formulated by Franken et al. [11].

This implementation of the gradient descent method as described in this report is not the only one that could be used. It should be possible to also use methods such as Momentum [12], to reduce the number of fluctuations seen in Fig. 11, or Adam [13] where the learning rate is adapted based on the first and second moment of the gradient. The latter could help with the rate at which the adaptive feed-forward learns, especially when in scenarios where some mass-reduction is already achieved and the associated power-errors are smaller.

The passivity preserving framework with adaptive feed-forward has also been applied in a few simulations on a one-degree-of-freedom robot with internal compliance. Although not enough simulations were performed to make general statements as to if either would also work on compliant robots, it can be said that some simulations were able to achieve some mass-reduction passively. Some simulations with feed-forward present did even reach a mass reduction of ten, at the costs of added virtual damping, which is lower than the theoretical limit derived by Colgate [14], who said that the virtual mass cannot go lower than $m_r\gamma$, with γ being the division ratio of mass on both sides of the compliance. A partial explanation for this is that the passivity preserving framework is based on time-domain passivity, whereas Colgate's limit is based on frequency domain passivity. An example and further elaboration can be found in section 4.2.

3.6 Conclusion

The presented framework for preserving passivity has shown that it can restore and preserve passivity in simulation by modifying virtual mass. The virtual damping present allows for some reduction in mass w.r.t. the actual end-effector mass on one axis but may have resulted in a lower apparent admittance on the other axis. In this framework, only the virtual model and by extent the virtual velocity is modified, so it should work with black box robot control in principle, where the desired velocity or position is put in, provided that the energy exchange between operator and robot can be measured accurately.

The proposed adaptive feed-forward as an addition to the passivity preserving framework has shown to work as well for this simulation, increasing the apparent admittance further. However, the author thinks that the mathematical support for this specific feedforward should be expanded further before applying it in a real setup or having it interact with humans.

Bibliography

[1] Z. Li, B. Huang, Z. Ye, M. Deng, and C. Yang, “Physical human–robot interaction of a robotic exoskeleton by admittance control,” *IEEE Transactions on Industrial Electronics*, vol. 65, no. 12, pp. 9614–9624, 2018.

[2] J. E. Colgate, M. Peshkin, and S. H. Klostermeyer, “Intelligent assist devices in industrial applications: a review,” in *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, vol. 3, 2003, pp. 2516–2521 vol.3.

[3] A. Keemink, “Haptic physical human assistance, chap. 4&5,” Ph.D. dissertation, University of Twente, 10 2017.

[4] J. Colgate, “The control of dynamically interacting systems,” *PhD thesis*, 01 1988.

[5] N. Hogan, “Controlling impedance at the man/machine interface,” in *Proceedings, 1989 International Conference on Robotics and Automation*, May 1989.

[6] F. Ferraguti and C. T. L. et al., “A variable admittance control strategy for stable physical human–robot interaction,” *The International Journal of Robotics Research*, vol. 38, no. 6, pp. 747–765, 2019.

[7] R. Aarts, “System identification and parameter estimation: Lecture 4,” May 2019.

[8] M. Franken, “Control of haptic interaction : an energy-based approach,” Ph.D. dissertation, University of Twente, Netherlands, 9 2011.

[9] J. Nakanishi and S. Schaal, “Feedback error learning and nonlinear adaptive control,” *Neural Netw.*, vol. 17, no. 10, p. 1453–1465, Dec. 2004.

[10] G. Folkertsma and S. Stramigioli, “Energy in robotics,” *Foundations and Trends® in Robotics*, vol. 6, no. 3, pp. 140–210, 10 2017.

[11] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, “Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[12] G. H. D. Rumelhart and R. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.

[13] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[14] E. Colgate and N. Hogan, “An analysis of contact instability in terms of passive physical equivalents,” in *Proceedings, 1989 International Conference on Robotics and Automation*, 1989, pp. 404–409 vol.1.

Chapter 4

Thesis Appendix

4.1 Effect of lower learning rate ϵ

This simulation is identical to the simulation with feedforward in section IV, with the exception of a reduced learning rate ϵ to 0.01.

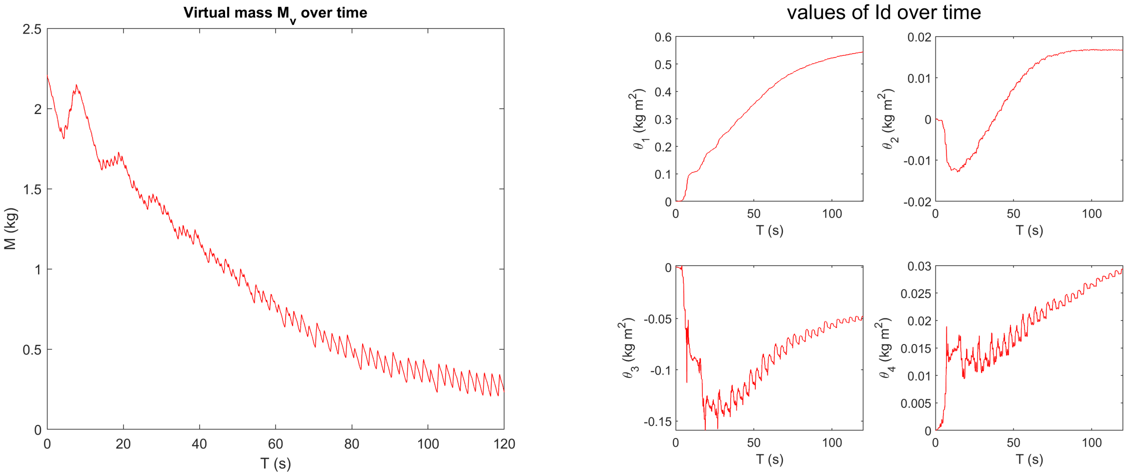

Figure 16: Effect of reduced ϵ. Left: evolution of virtual mass over time. Right: Values in I_d over time. Virtual mass has not reached desired value after 120s. Cause is a lower adaptation speed of the feedforward.

4.2 Simulation of 1DoF compliant robot

As already mentioned, some simulations were performed on a one DoF robot with internal compliance. The situation is shown in Fig. 17. An added challenge in this is that estimating the energy exchanged becomes more difficult, as the exact position of the end-effector is unknown. For the energy exchange estimation between robot and operator the same estimation is used as in section IV, but with actuator coordinates instead of end-effector coordinates:

$$E_{hr}(k) = E_{hr}(k-1) + F_h(k)(x_a(k) - x_a(k-1)). \tag{64}$$

At the same time the actual energy exchanged at the end-effector is also monitored, but this information is not available to the controller:

$$E_{hr,a}(k) = E_{hr,a}(k-1) + F_h(k)(x_p(k) - x_p(k-1)). \tag{65}$$

The virtual model is a mass-damper system attached to the end-effector with constant mass-to-damping ratio $R_d = 2$ N s/m kg. It is instantiated with the exact same virtual mass as actually present in the system in total, which is 2 kg. The mass division within the robot is 50% , so $M_1 = M_2 = 1$ kg. The admittance controller is run at one kilohertz, with the detection rules being checked at 100 Hz. M_{loss} is set to one gram and E_{lim} to 3 Joule.

Similar to the experiments in IV, the end-effector is dragged around. The reference trajectory for this is:

$$x(k) = 0.5\sin(0.3\pi k/Fs) - 0.3\sin(0.5\pi\sin(0.124\pi k/Fs) + 0.81\pi k/Fs), \tag{66}$$

which is a periodic reference trajectory, one for which the period approximately 46.7 seconds, so two periods fit within the simulation time, which is 100 seconds. The feedforward model is a mass-feedforward: $Fff = \theta\dot{v}_d$.

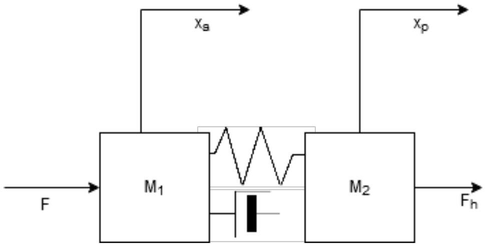

Figure 17: Situation of one DoF robot with internal compliance with internal stiffness $K_{compliance}$ and internal damping $D_{compliance}$

The effect of the internal stiffness and stiffness of the operator on the ability of the passivity preserving framework will be shown. The damping of the operator will be fixed to 1 N s/m, and the internal damping is chosen such that the internal resonance mode has a damping ratio of approx. 0.6.

The internal stiffness was varied between 300 and 800 N/m, while the internal compliance's stiffness is varied between 1 and 25 kN/m. In this simulation, the controller attempts to reduce the virtual mass to 0.2 kg. If at any point it is determined via the actual energy exchanged between robot and operator ($E_{hr,a}$) that strict passivity has been violated, and this is not detected by the energy exchange of the controller (E_{hr}), the simulation is halted and flagged.

The results of this simulation can be found in Fig. 18. It can be seen that a significant portion of the simulation has reached the desired virtual mass of 0.2 kg, which effectively means a mass reduction of ten, at the cost of some damping. In this, we are just going to assume that the virtual dynamics are the same as the dynamics displayed on the end-effector.

However, simply stating that the virtual dynamics have converged to the desired value is not sufficient. It should also be verified that the actual exchanged energy does not violated strict passivity. This can be seen in Fig. 19. There is a area that is similar to the portion in Fig. 18 which is colored

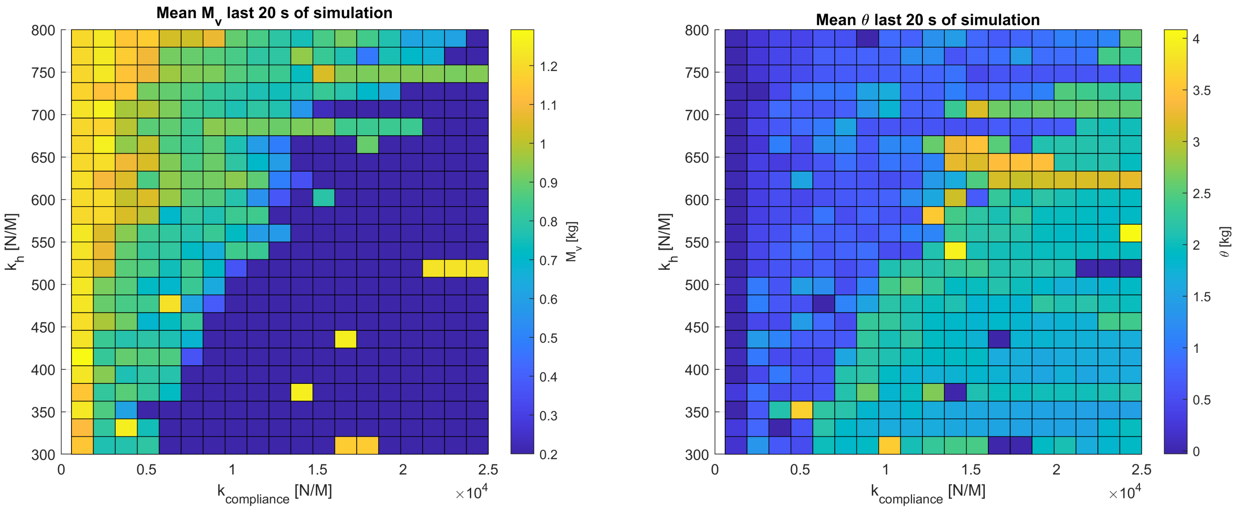

Figure 18: Effect of operator stiffness and compliance stiffness on passivity preserver with feedforward. Left: average virtual mass over last 20 seconds of simulation. Right: average feedforward parameter

blue, which means that strict passivity was not violated once during the simulation. There are a few isolated simulations within this blue area for which passivity is not preserved. In these cases, it was determined that for some unknown reason, the feedforward simply did not make any adaptations and remained zero. This resulted in a situation where at some point strict passivity was violated. Whether the passivity framework could restore passivity at that point could not be determined, as the simulation was halted.

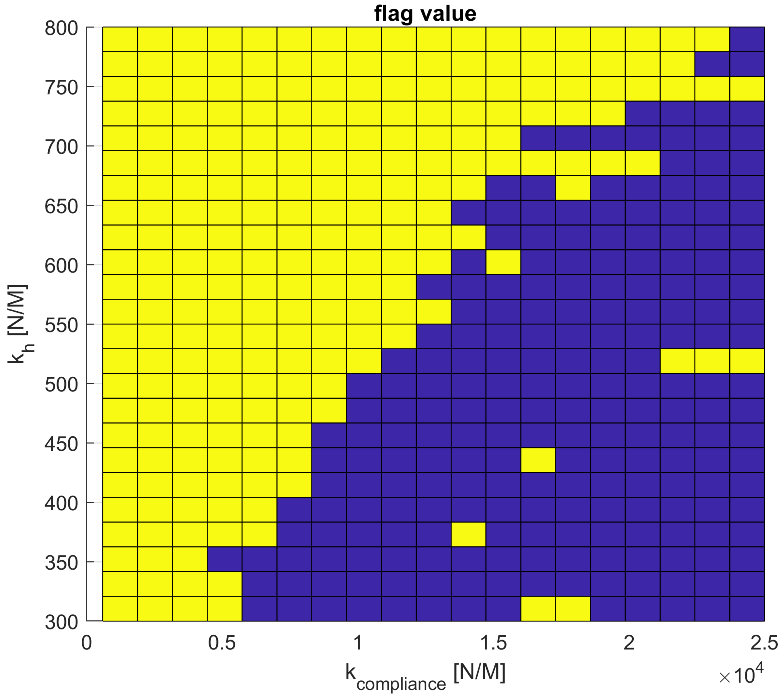

Figure 19: Flag on strict passivity violated. Yellow means violation of passivity, blue means no violation