Control of a multi-robot cooperative team guided by a human-operator

Zwischenbericht zur
MASTERARBEIT
von

cand. ing. Martin Angerer

geb. am 10.06.1991
wohnhaft in:
Steinheilstrasse 5
80333 München
Tel.: 0151 57978548

Lehrstuhl für
INFORMATIONSTECHNISCHE REGELUNG
Technische Universität München

Univ.-Prof. Dr.-Ing. Sandra Hirche

Betreuer: Selma Musić, M.Sc.
Beginn: 01.10.2015
Zwischenbericht: 21.01.2016
Abgabe: 01.04.2016

In your final hardback copy, replace this page with the signed exercise sheet.

Abstract

Cooperative manipulation with human guidance can be used to solve versatile tasks. A new approach to system modelling and control in cooperative manipulation is the use of port-Hamiltonian systems. Starting from modelling of a cooperative manipulation set-up a model-based controller in the framework of Intrinsically Passive Control is derived. In contrast to the quasi-static implementations for robot hands, the controller has fully dynamic impedance relations. The good dynamic performance of the proposed control scheme is shown by simulation and compared to simulations of state-of-the-art controllers in cooperative manipulation.

Zusammenfassung

Hier die deutschsprachige Zusammenfassung

Contents

Chapter 1

Introduction

Cooperative manipulation involves two or more robot arms collectively moving or manipulating a common object [CM08]. Such a set-up overcomes many limitations seen by single robots. The prime example is the transportation of a large or heavy object, where a single manipulator would exert large local torques on the object. Back in the 1940s two remote manipulators were used to handle radioactive goods [Goe52]. Around 1990 the *NASA* researched cooperative manipulation in space construction applications [SC92]. Specifically in [LS05] a cooperative manipulation strategy for the repair of the Hubble Space Telescope is proposed. Other tasks are for example:

- assembly of multiple parts without using special fixtures
- grasping an object without rigid fixture but by exerting a suitable squeezing force
- deforming a flexible object
- coordinated use of tools

For the last point a recent application has been presented at the Fukushima Daiichi Nuclear Power Station. By equipping a two-armed robot with different tools it is capable of grasping an item (e.g. piping) and cutting it with the other, see Fig. 1.1. For sensitive tools like an angle grinder it is favourable that movement and contact forces of both arms are coordinated.

Space and hazardous environments are fruitful environments for cooperative manipulators, often they are directly controlled by a human operator. Combining human reasoning and the enhanced flexibility of a cooperative set-up is a powerful combination in an unstructured environment. A human in the control loop comes with superior foresight and planning capabilities. To fully exploit this additional flexibility we require control architectures able to perform all the fore-mentioned tasks and operable in a comfortable and efficient manner.

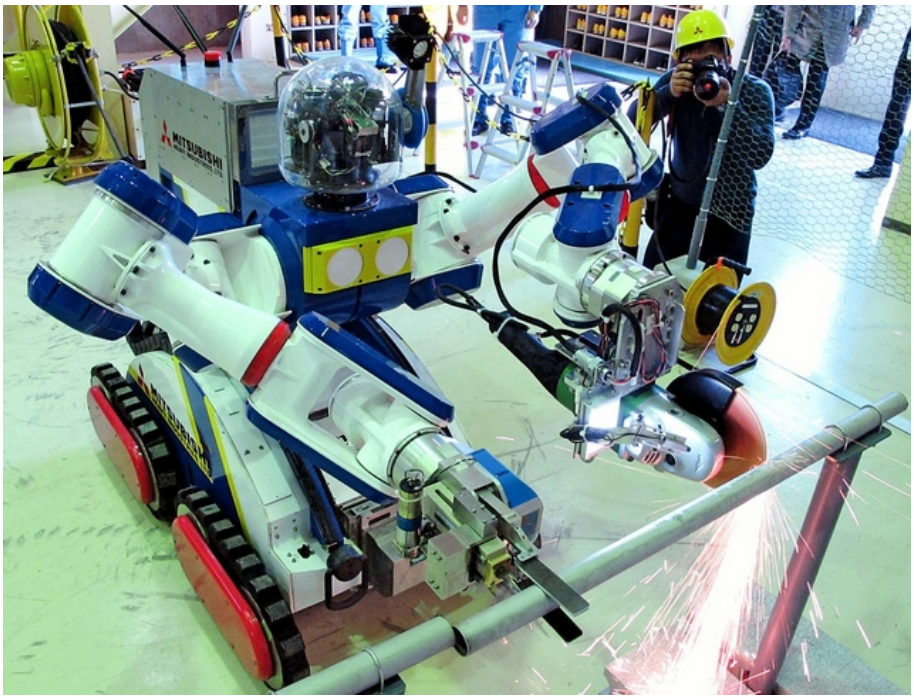

Figure 1.1: Demonstration of the tele-operated MHI MEISTeR robot at Fukushima Daiichi Nuclear Power Station [LTD14]

To address the first requirement the controller realizes changes of formation and relocates the constrained system. A change of formation first of all means the opening and closing of a grasp a around a common object. A virtual object and variable rest-length springs are introduced, the springs couple the virtual and the actual object. The size of virtual object specifies the rest-lengths, by choosing it smaller than the actual a grasping force is exerted. By choosing it bigger the grasp is opened and the actual object is released. Changing of rest-lengths means to change the potential energy stored in the springs. Since the controller itself is passive, the energy is exchanged with a distinct power port. In order to relocate the formation a spring is connected from the virtual object center to the desired new position/orientation. This ensures compliant behavior between (virtual) object and environment. Note that this concept is called *Intrinsically Passive Controller* (IPC) and was introduced by Stramigioli [Str01b], the control architecture described in the main part will closely stick to this concept.
The controller provides a reasonable layer of abstraction and takes responsibilities from the operator, in order to not overstrain her/his attention and let him focus on important elements of the task. To ensure a secure and stable grasp, the formation of robots is locally controlled without the help of the human. For dynamic manipulation tasks grasp force optimization can be utilized to limit internal to the absolute

minimum. The control system provides appropriate feedback to provide better perception on the work environment and to make the job more intuitive. The user specifies system motion and is provided with information about the necessary forces to accomplish the commanded motion. Thereby s/he gets a natural intuition of how much work has been done by the robots. To this end the controller shall never be a source of additional energy, the only way energy is fed into the system is by request of the operator, i.e. the controlled robots are passive. The maximum extractable energy is always bounded, the operator can estimate the stored energy and possible effects when interacting with the environment. Passive robotic systems are always stable when interconnected passive environments/humans and can be stable with some active systems. On the contrary, if the controlled robot is not passive, there is always a passive environment that destabilizes the interconnected system [Str15].

1.1 Problem Statement

Manipulating an unknown object in an unstructured environment is an interesting task for a human-guided cooperative manipulation set-up. Dropping the assumption of a rigid- connection between manipulator and object, the grasp has to be actively stabilized under varying circumstances. During dynamic manipulation a slipping of contact due to the inertia of the object has to be avoided using an automatic mechanism. On the other hand the operator must directly adjust grasp forces for heavy or fragile objects.
Position and velocity of the manipulated object are difficult to track in an everyday setting. Usually only end-effector position and velocity of the robots are available. Object position has to be estimated from the known data.
For grasping an object of unknown or even flexible shape, the determination of a fixed grasp map in advance is not useful. Size and shape of the grasp have to be determined by the operator during the grasp process. The controller has to be flexible to varying grasp geometries during the whole task execution.
Integrating the human into the control loop, energetic passivity of the closed loop system is a meaningful and intuitive way to ensure stability and a natural way of interaction. Energetic passivity limits the extractable mechanical energy from the closed loop system. This means that the potential damage is also limited. Energetic passivity of the robotic system ensures stability of the interconnection stability with any passive systems. Since humans and many relevant environments are passive the closed loop system will always be stable if the overall control scheme is designed energetically passive.

1.2 Related Work

A notable class of control architectures for cooperative manipulation is hybrid position/force control, with a motion control loop for trajectory tracking and a force

control loop for internal forces [WKD92, Hsu93]. Their drawback is the inability to handle non-contact to contact transitions. Hogan introduced impedance control, which enforces a relation between force and motion [Hog84]. Its first application in cooperative manipulation was in [SC92] for realizing compliant object-environment interaction (external impedance control). Bonitz and Hsia [BH96] applied the concept to the manipulator-object relation (internal impedance control).
More recent impedance control schemes can be classified in terms of the information and sensor data available for the problem. Frugal architectures are formation control [SMH15] and the static IPC [WOH06]. Both do not incorporate the object dynamics in control, thus very little knowledge about the object (e.g. dimensions) is necessary. The control loops depend only on relative positions and velocities of the manipulators and do not require object tracking. Neglecting considerable object dynamics is an obvious drawback of the schemes.
The concept of the *Intrinsically Passive Controller* (IPC), introduced by Stramigioli [Str01b] and called dynamic IPC by Wimböck et al. [WOH08], tries to overcome some of the limitations. In the controller the object is represented by a virtual pendent and simulated to reproduce its dynamics for control purpose. This has the advantage that still no tracking of the object is required, object velocity and even acceleration can be obtained from the simulation.
Techniques which rely on exact knowledge of the object motion were introduced by Caccavale et al. [CV01, CCMV08] and more recent [HKDN13] and [DPEZ$^+$15]. It is common to them that they assume rigid fixtures to a rigid object, these conditions overcome the problem of object tracking. The approaches by Caccavale and coworkers use force/torque sensors at the manipulators to establish compliant object environment interaction, for this purpose Heck et al. [HKDN13] assume to have an exact model of the environment. Stramigioli's IPC implements a compliant relation between virtual object and environment.
The human operator must be able control the manipulators at a reasonable degree of complexity, therefore in an direct master-slave approach each robot is controlled independently by a human operator. Exactly coordinating their motions is a difficult task for humans, as a consequence a certain amount of autonomy is left to the robot system, enabling a single operator to control the cooperative system. Lee and Spong [LS05] apply the master-slave scheme but treat the constrained system as a single slave, while the formation is preserved by the robots autonomously. Many master-slave systems give the operator force-feedback, while s/he commands the motion. This helps the operator compensate for resistances and gives a natural feeling of the interaction with the environment. The structure then is fully bi-directional, ones refers to bilateral telemanipulation [NPH08]. Leader-follower [SMH15, SMP14] schemes differ in terms of feedback provided to the operator. Tactile and visual types are non-reactive, i.e. they do not induce operator movements reacting to a back-driving force [MT93]. A purely vision-based architecture is introduced by Gioioso et al. [GFS$^+$14], hand gestures are used to both control the motion of the constrained system and the opening and closing of a grasp. Control architectures that leave

even further autonomy to the robot system and possess a closed local, autonomous control loop, are categorized as supervisory control. They interacts with the operator by continuously sending information about the state and periodically receiving commands [She92].
The assumption of rigid fixtures between manipulators and object is very common in cooperative manipulation, Lee and Spong [LS05] are an exception. Friction grasps are mainly researched in robot hand literature ([WOH06, WOH08, Str01b]). For the stabilization of a friction grasp it is vital to choose appropriate forces depending on the dynamic state. Therefore the Coulomb friction constraints along with other criteria (safety margins, force limits) can be formulated as a cost function for optimization. Buss et al. [BHM96] realized that the Coulomb friction constraints can be formulated as positive definite matrices, Han et al. [HTL00] gave a linear matrix inequality problem. For this type of optimization problems very efficient, real-time solvers exist.

Chapter 2

Main Part

2.1 port-Hamiltonian systems

Port-Hamiltonian systems(PHS) provide a framework to for the geometric description of complex physical systems. The goal is divide the system in simple Hamiltonian subsystems and interconnect them using methods of network theory.
The Hamiltonian equations of motion are related to the better known Euler-Lagrange equations with the Legendre transformation. Both are formulations on energy level thus they are useful for a continuous description of mixed-energy systems, e.g. a pendulum alternating between kinetic and potential energy periodically.

2.1.1 Hamiltonian description of mechanical systems

The classical Hamiltonian equations for a mechanical system are:

$$\dot{q} = \frac{\partial H}{\partial p}(q,p) \tag{2.1}$$

$$\dot{p} = -\frac{\partial H}{\partial q}(q,p) + F \tag{2.2}$$

Where Hamiltonian $H(q,p)$ is the total energy of the system, $q = (q_1, ..., q_k)^T$ denotes the generalized coordinates of the system with k-degrees of freedom and $p = (p_1, ..., p_k)^T$ is the vector of the generalized momenta. F is the input of generalized forces.
An important aspect of PHS is conservation of energy, i.e. supplied work is either stored in the system [vdS06]. The energy balance of the system is:

$$\frac{d}{dt}H = \frac{\partial^T H}{\partial q}(q,p)\dot{q} + \frac{\partial^T H}{\partial p}(q,p)\dot{p} = \frac{\partial^T H}{\partial p}(q,p)F = \dot{q}^T F \tag{2.3}$$

While the generalized force F is the input of the system, the output can be defined as $e = \dot{q}$, for a mechanical system $\dot{q}$ is the generalized velocity. Note that the product

of Force and velocity is mechanical power.
PHS can be described in local coordinates x on the n-dimensional state space manifold $\mathcal{X}$, $x \in \mathcal{X}$:

$$\dot{x} = J(x)\frac{\partial H}{\partial x}(x) + g(x)f \tag{2.4}$$

$$e = g^T(x)\frac{\partial H}{\partial x}(x) \tag{2.5}$$

Where $J(x)$ is the $n \times n$ structure matrix, $J(x)$ is skew-symmetric, i.e. $J^T(x) = -J(x)$. Furthermore $g(x)$ is the $n \times m$ input matrix and $f \in \mathbb{R}^m$ is a general input vector. With the skew-symmetry of $J(x)$ the energy balance of the local coordinate system is $\frac{dH}{dt}(x) = e^T f$.
Any energy conservative physical element can be described in this formulation. To account for dissipative elements the previous can be extended by a positive semi-definite matrix $R(x)$.

$$\dot{x} = (J(x) - R(x))\frac{\partial H}{\partial x}(x) + g(x)f \tag{2.6}$$

$$e = g^T(x)\frac{\partial H}{\partial x}(x) \tag{2.7}$$

The energy balance changes since the supplied power is not only stored but dissipated:

$$\frac{dH}{dt}(x) = e^T f - \left(\frac{\partial H}{\partial x}(x)\right)^T R(x)\frac{\partial H}{\partial x}(x) \tag{2.8}$$

Consider the example of a simple one-dimensional spring-mass-damper system described by $M\ddot{x} = -kx - Dx + F$. Where M, k, D, F denote the mass, stiffness, damping and external force acting on the mass respectively. In the usual state space formulation this is:

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -D \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F \tag{2.9}$$

One obtains an alternative representation by choosing energy variables as state variables, i.e. the energy stored in the spring is described by the displacement x and the energy of the moving mass by the momentum $p = M\dot{x}$:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix}}_{\dot{\chi}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & -D \end{bmatrix}}_{J(\chi)-R(\chi)} \underbrace{\begin{bmatrix} kx \\ \dot{x} \end{bmatrix}}_{\partial_\chi H(\chi)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{g(\chi)} \underbrace{F}_{f} \tag{2.10}$$

This corresponds to the first equation of a PHS of the form. This simple example can be divided into a set of basic mechanical elements:

Table 2.1: PHS variables of mechanical elements

	Spring	Mass	Damper
Effort variable	Force F	Velocity $\dot{x}$	Force F
Flow variable	Velocity $\dot{x}$	Force $F = \dot{p}$	Velocity $\dot{x}$
State variable	Position x	Momentum p	-
Energy function	$E(x) = \frac{1}{2}kx^2$	$E(p) = \frac{p^2}{2m}$	$E(\dot{x}) = D\dot{x}^2$ (diss. co-energy)

- a moving inertia
- a spring
- a damper

Each element is described by a effort and a flow variable, the energy state variable is the time integration of the flow variable. Table 2.1 summarizes the description variables of mechanical elements.

Complex physical systems can be modelled as a network of energy storing and dissipating elements, similar to representation of electrical networks consisting of resistors, inductors and capacitors. The rules of interconnection are Newton's third law (action-reaction), Kirchhoff's laws and power-conserving elements like transformers or gyrators. The aim of PHS-modelling is to describe the power-conserving elements with the interconnection laws as a geometric structure and to define the Hamiltonian function as the total energy of the system.

2.1.2 Interconnection of port-Hamiltonian systems

Holonomic motion constraints, as in cooperative manipulation (see for example [EH15a]) of the form $A\dot{x} = 0$ impose algebraic constraints on the system. This constraints restricts the set of possible state-space of the sub-systems, leading to an *implicit system*. Implicit PHS can by defined by the concept of a *Dirac structure*. The linear vector space V and its dual space V^*. V can be considered as the space of flows and V^* the space of efforts. Then the space of power variables is $\mathcal{P} = V \times V^*$. The power is calculated with the duality product $\langle \, | \, \rangle$:

$$P = \langle e|f \rangle, \;\; e \in V^*, \; f \in V \tag{2.11}$$

In euclidean space this simplifies to the standard scalar product, i.e. $P = e^T f$.
A constant Dirac structure on V is a linear subspace

$$\mathcal{D} \subset V \times V^* \tag{2.12}$$

such that

1. $e^T f = 0$ for all $(e, f) \in \mathcal{D}$

2. $dim\mathcal{D} = dimV$

3. $\mathcal{D} = \mathcal{D}^{\perp}$ where $\perp$ denotes the orthogonal complement with respect to the bilinear form $\langle\ ,\ \rangle_{V\times V^*}$ defined as $\langle (f_1, e_1), (f_2, e_2)\rangle_{V\times V^*} = \langle e_1|f_2\rangle + \langle e_2|f_1\rangle$

The Dirac structure $\mathcal{D}$ on V defines a power power-conserving relation between the power variables $(f, e) \in V \times V^*$ which has maximal dimension [vdS06].
A general PHS represented as a Dirac structure has four ports for energy exchange (see Fig. **??**). These are energy storage port $\mathcal{S}$, the energy-dissipation part $\mathcal{R}$ (accounting for resistive elements) and two external ports, dedicated for energy exchange with the environment $\mathcal{I}$ and with a controller $\mathcal{C}$.

Energy storage port

The port accounts for the internal storage of the system, its port variables are (f_S, e_S). The resulting energy balance is:

$$\frac{d}{dt}H = \left\langle \frac{\partial H}{\partial x}(x)|\dot{x} \right\rangle = \frac{\partial H}{\partial x}(x)\dot{x} \tag{2.13}$$

The flow variable is the energy rate $f_S = -\dot{x}$ and the effort variable is the co-energy $e_S = \frac{\partial H}{\partial x}(x)$.

Energy dissipation port

The port corresponds to internal dissipation and can be used to model resistive elements. The port variables are described by the general resistive:

$$R(f_R, e_R) = 0 \tag{2.14}$$

This always holds $\langle e_R|f_R\rangle \leq 0$ (energy dissipation). For an uncontrolled system that does not interact with the environment, i.e. no energy exchange through the two external ports, the energy balance is:

$$\frac{dH}{dt} = -e_S^T f_S = e_R^T f_R \leq 0 \tag{2.15}$$

External ports

For a complete description the interaction with the environment can be expressed through the interaction port $\mathcal{I}$ and its variables (f_I, e_I). Furthermore the system is open to controller action through the control port $\mathcal{C}$ with the variables (f_C, e_C).
The power balance of the whole system then is

$$e_S^T f_S + e_R^T f_R + e_I^T f_I + e_C^T f_C = 0 \tag{2.16}$$

or by using (2.13)

$$\frac{dH}{dt} = e_R^T f_R + e_I^T f_I + e_C^T f_C \tag{2.17}$$

It is important to notice that the interconnection of two PHS is still a PHS. Consider two systems with open control and environment interaction ports:

$$\dot{x}_i = (J_i - R_i)\frac{\partial H_i}{\partial x_i} + \begin{pmatrix} g_i^C & g_i^I \end{pmatrix} \begin{pmatrix} f_i^C \\ f_i^I \end{pmatrix} \tag{2.18}$$

$$\begin{pmatrix} e_i^C \\ e_i^I \end{pmatrix} = \begin{pmatrix} (g_i^C)^T \\ (g_i^I)^T \end{pmatrix} \frac{\partial H_i}{\partial x_i} \tag{2.19}$$

For notational convenience the usual dependencies on the states have been omitted. The control inputs and outputs are now connected by setting $f_1^C = e_2^C$ and $f_2^C = -e_1^C$. To be consistent with power the minus sign in the last equation is necessary. The power exchanged by the i-th system is $P_i = (e_i^C)^T f_i^C$, therefore the total exchanged energy fulfils $P_1 + P_2 = 0$. The resulting interconnected system has still the environment interaction ports open:

$$\dot{x} = (J - R)\frac{\partial H}{\partial x} + \begin{pmatrix} g_1^I & g_2^I \end{pmatrix} \begin{pmatrix} f_1^C \\ f_2^I \end{pmatrix} \tag{2.20}$$

$$\begin{pmatrix} e_1^I \\ e_2^I \end{pmatrix} = \begin{pmatrix} (g_1^I)^T \\ (g_2^I)^T \end{pmatrix} \frac{\partial H}{\partial x} \tag{2.21}$$

Where $x = (x_1, x_2)^T$ and $H = H_1 + H_2$ is the sum of the two energies. The structure and dissipation matrix become become

$$J = \begin{pmatrix} J_1 & g_1^C (g_2^C)^T \\ -g_1^C (g_2^C)^T & J_2 \end{pmatrix}$$

,

$$R = \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix}$$

2.2 3D-space modelling of mechanical systems

2.2.1 Euclidean space and motions

Coordinate frames

A coordinate frame of the three-dimensional Euclidean space is a 4-tuple of the form $\Psi = (o, \hat{x}, \hat{y}, \hat{z})$. Where o is the three-dimensional vector of the origin and $\hat{x}, \hat{y}, \hat{z}$ are the linear independent, orthonormal coordinate vectors. Consider two coordinate frames Ψ_1, Ψ_2 which share the same origin but differ in orientation due to different choices of $\hat{x}_i, \hat{y}_i, \hat{z}_i,\ i = 1, 2$. The change of orientation from Ψ_i to Ψ_i is described by

the rotation matrix R_i^j. The set of rotation matrices is called *special orthonormal* group ($SO(3)$) [Str01a] and is defined as:

$$SO(3) = \{R \in \mathbb{R}^{3\times3} \mid R^{-1} = R^T, detR = 1\} \tag{2.22}$$

Usually coordinate frames are defined with respect to an inertial frame, and the coordinate vectors $\hat{x}, \hat{y}, \hat{z}$ are chosen equal for all frames, deviations of orientation are represented by a rotation matrix relative to the inertial frame. In general a change of coordinate frames from Ψ_i to Ψ_j can be expressed with the homogeneous matrix

$$H_i^j := \begin{pmatrix} R_i^j & p_i^j \\ 0_{1\times3} & 1 \end{pmatrix}$$

Where $p_i^j = o_j - o_i$ denotes the changes of origins. A point $p^1 \in \mathbb{R}^3$ expressed in Ψ_i is cast into Ψ_j by

$$\begin{pmatrix} p^j \\ 1 \end{pmatrix} = H_i^j \begin{pmatrix} p^i \\ 1 \end{pmatrix} \tag{2.23}$$

. The inverse transformation H_j^i is given by

$$H_i^j = (H_i^j)^{-1} = \begin{pmatrix} (R_i^j)^T & -(R_i^j)^T p_i^j \\ 0_{1\times3} & 1 \end{pmatrix}$$

and is still a homogeneous matrix. The set of homogeneous matrices is called the *special Euclidean* group:

$$SE(3) := \{\begin{pmatrix} R & p \\ 0 & 1 \end{pmatrix} \mid R \in SO((3), p \in \mathbb{R}^3\} \tag{2.24}$$

The $SE(3)$ is a matrix lie group, composed of the set of homogeneous matrices H_i^j and the matrix multiplication being the group operation. For more information on Lie groups see e.g. [Str01b].

Twists and wrenches

Consider any point p not moving in coordinate frame Ψ_i, i.e. $\dot{p}^i = 0$. If p is moving in another coordinate frame Ψ_j, the two frame move with respect to each other. The trajectory can be described as a function of time: $H_i^j(t) \in SE(3)$. By differentiating (2.23) one obtains

$$\begin{pmatrix} \dot{p}^j(t) \\ 1 \end{pmatrix} = \dot{H}_i^j(t) \begin{pmatrix} p^i \\ 1 \end{pmatrix}$$

$\dot{H}_i^j$ describes both motion and a change of the reference frame. A separated representation is

$$\begin{pmatrix} \dot{p}^j(t) \\ 1 \end{pmatrix} = \tilde{T}_i^{j,j} \left(H_i^j \begin{pmatrix} p^i \\ 1 \end{pmatrix} \right) \tag{2.25}$$

More formally speaking $\dot{H}_i^j$ is a tangential vector along the trajectory $H_i^j(t)$ and thus in the tangent space of the $SE(3)$: $\dot{H}_i^j \in T_{H_i^j}SE(3)$. The tangent space $T_{H_i^j}SE(3)$ depends on the relation of Ψ_i and Ψ_j. To obtain a representation of motion which is referenced to a coordinate frame, we can map $\dot{H}_i^j$ to the identity of the $SE(3)$. At the identity e of the $SE(3)$ the tangent space $T_eSE(3)$ has the structure of a Lie algebra. The Lie algebra of the $SE(3)$ is denoted by $\mathfrak{se}(3)$. This is done either by left or right translation, for a definition see [Str01b]. The right translation is used in (2.25) and is written compact

$$\dot{H}_i^j = \tilde{T}_i^{j,j} H_i^j \tag{2.26}$$

The left translation leads to

$$\dot{H}_i^j = H_i^j \tilde{T}_i^{i,j} \tag{2.27}$$

We call $\tilde{T} \in T_eSE(3)$ a twist and the $\mathfrak{se}(3)$ the space of twists. It can already be seen from the upper equations that different representations exist:

- $T_i^{k,j}$ is the twist of Ψ_i with respect to Ψ_j expressed in the frame Ψ_k
- $T_i^j = T_i^{j,j}$ is the twist of Ψ_i with respect to Ψ_j expressed naturally in Ψ_j

The 4×4 matrix $\tilde{T}$ can be decomposed and there exists also a vector representation $T \in \mathbb{R}^6$

$$\tilde{T} = \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} \ , \ T = \begin{pmatrix} \omega \\ v \end{pmatrix}$$

Wherein v is the velocity and ω is the angular velocity. $\tilde{\omega}$ is the skew-symmetric representation of the vector ω

$$\omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} \ \Rightarrow \ \tilde{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & \omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \tag{2.28}$$

Changes of coordinates for twists are of the form

$$\tilde{T}_i^{j,j} = \tilde{T}_i^j = H_i^j \tilde{T}_i^{i,j} H_j^i$$

or for the vector representation

$$T_i^j = Ad_{H_i^j} T_i^{i,j} \ , \ Ad_{H_i^j} = \begin{pmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{pmatrix}$$

The dual vector space of $\mathfrak{se}(3)$ is the space of linear operations from $\mathfrak{se}(3)$ to $\mathbb{R}$. It is denoted by $\mathfrak{se}^*(3)$ and it represents the space of wrenches W.
Wrenches decompose to moments m and forces f.

$$W = (m \ \ f) \ , \ \tilde{W} = \begin{pmatrix} \tilde{f} & m \\ 0 & 0 \end{pmatrix}$$

Again $\tilde{W} \in \mathbb{R}^{4\times4}$ is a matrix while $W \in \mathbb{R}^6$ is the row vector representation.
The change of coordinates for twists is similar to the case of twists:

$$(W^i)^T = Ad^T_{H_i^j}(W^j)^T$$

Here the mapping is in the opposite direction, from Ψ_j to Ψ_i, what is a consequence of the fact that wrenches are duals to twists. Again there are different representations of wrenches:

- $W_i^{k,j}$ is the wrench applied to a spring connecting Ψ_i to Ψ_j on the side of Ψ_i expressed in the coordinate frame Ψ_k.
- $W_i^j = W_i^{j,j}$ is the wrench applied (by a body attached to Ψ_i) to a spring between Ψ_i and Ψ_j expressed in the coordinate frame Ψ_j.

Both representations describe the wrench by a body on a spring. Applying the principle of action and reaction the wrench exerted by a spring on a body is simply $-W_i^j$ or $-W_i^{k,j}$ respectively. Due to the nodicity of a spring we have: $W_i^{k,j} = -W_j^{k,i}$
Recalling Subsection 2.1.2 the power flowing through a port is defined by the dual product of an effort and flow pair, which are from a vector space and its dual respectively. On a vector space level the power port $\mathcal{P}$ is defined by the Cartesian product of the Lie algebra $\mathfrak{se}(3)$ and its dual $\mathfrak{se}^*(3)$: $\mathcal{P} = \mathfrak{se}(3) \times \mathfrak{se}^*(3)$. The mechanical power exchanged at the port is simply $P = WT$.

2.2.2 Dynamics of physical components

Springs

Speaking of a *spring* we mean the ideal, lossless representation of a potential energy storing element. A spring is connected to at least two bodies and is defined by an energy function. The energy is a function of only the relative positions of the bodies to which the spring is attached. Consider a spring between the two bodies B_i and B_j, with coordinate frames Ψ_i and Ψ_j fixed to the respective body. The stored potential energy is positive definite function of the form [SMA99]

$$V_{i,j} : SE(3) \to \mathbb{R};\ H_i^j \mapsto V_{i,j}(H_i^j) \tag{2.29}$$

Recalling the general equations of a PHS and the assignment of the variables from Subsection 2.1.1 we already have the structure of the PHS equations for a spring. The state variable is the relative displacement H_i^j, the effort variable is the wrench $W_i^{j,j}$ and the flow variable is the relative motion $\dot{H}_i^j$

$$\dot{H}_i^j = T_i^j H_i^j \tag{2.30}$$

$$W_i^j = \frac{\partial V_{i,j}}{\partial H_i^j}(H_i^j)^T \tag{2.31}$$

Note that V_i, j has an energetic minimum when H_i^j is the identity matrix I_4. An energetic minimum physically necessary, otherwise infinite energy would be extractable. With $H_i^j = I_4$ the frames Ψ_i and Ψ_j coincide. It is possible to define springs with non-zero rest-length by introducing coordinate frames Ψ_{ic} and Ψ_{jc} rigidly attached to Ψ_i and Ψ_j respectively. The spring is now between the new frames, thus the energetic minimum is $H_{ic}^{jc} = I_4$. The displacements H_i^{ic}, H_j^{jc} are the resulting rest-lengths. Frames Ψ_{ic} and Ψ_{jc} can be chosen to represent the *center of stiffness*, where translation and rotation are maximally decoupled [**?**].

Variable rest-length springs

Varying the rest-lengths of springs in manipulator-object interaction allows to perform a grasp and specify grasping forces. The rest-length influences the energy configuration, to be consistent with the previous an additional power port is introduced. The chosen hinge points Ψ_b, Ψ_j of the spring define an axis, known as the principal axes of stiffness. Changes of the rest-length leave this axis unaffected, i.e. the displacement is in-line. With reference to Fig. 2.1, the change of rest-length is a change of relative displacement of Ψ_b and Ψ_i: H_i^b. Towards a PHS representation

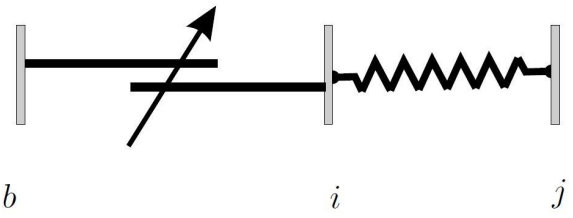

Figure 2.1: Variable rest-length spring [SMA99]

we need to identify the components contributing to the deformation twist of the spring T_i^j: the displacement twist of the bodies attached to the spring T_b^j and the twist resulting from a commanded change of rest-length T_i^b [SD01].

$$T_i^j = T_b^j + Ad_{H_b^j} T_i^b \tag{2.32}$$

Since no additional energy storages were introduced, the Hamiltonian function is equal to the case of the previous simple spring. We can write the two-port Hamiltonian system of a variable length spring as

$$\dot{H}_i^j = \left(\begin{pmatrix} 1 & Ad_{H_b^j} \end{pmatrix} \begin{pmatrix} T_b^j \\ T_i^b \end{pmatrix} \right) H_i^j \tag{2.33}$$

$$\begin{pmatrix} W_b^j \\ W_i^b \end{pmatrix} = \left(\begin{pmatrix} 1 \\ Ad_{H_b^j}^T \end{pmatrix} \frac{\partial V_{i,j}}{\partial H_i^j} \right) (H_i^j)^T \tag{2.34}$$

Inertias

The kinetic energy of a body is a function of its relative motion with respect to an inertial frame Ψ_0. The body's dynamic properties are described by the inertia matrix M_b, which consists of mass and rotational inertia components. Consider the classical description of a body's dynamics:

$$M_b \dot{T}_b^{b,0} = C_b T_b^{b,0} + W_{ext}^b \tag{2.35}$$

Where C_b accounts for Coriolis and Centripetal terms and W_{ext}^b is the external wrench acting on the body.
Recall from Table 2.1 that the energy state variable is the momentum $P_b^b = M_b T_b^{b,0}$ and the flow variable is the wrench, we can reformulate the previous equation to PHS form

$$\dot{P}_b^b = C_b \frac{\partial V_k(P_b^b)}{\partial P_b^b} + I_6 W_{ext}^b \tag{2.36}$$

$$T_b^{b,0} = I_6 \frac{\partial V_k(P_b^b)}{\partial P_b^b} \tag{2.37}$$

The kinetic energy function is given as $V_k = \frac{1}{2}(P_b^b)^T M_b^{-1} P_b^b$. In cooperative manipulation we often deal with heavy objects, it is thus inevitable to model gravity of the virtual object. A potential energy is assigned to a mass in the gravitational field. This can be modelled with a spring connecting the body and an inertial frame associated with the ground. This spring can be formulated as a PHS using the left translation (2.2.1)

$$\dot{H}_b^0 = H_b^0 T_b^{b,0} \tag{2.38}$$

$$W_b^{b,0} = (H_b^0)^T \frac{\partial V_g}{\partial H_b^0} \tag{2.39}$$

Where V_g is the gravitational potential energy of the body. For a combined description the potential and kinetic energy add up: $V_{kg} = V_k + V_g$. Since there are two types of energy stored by *one* body, the twists in both energy systems are equal. The wrenches on the body add up

$$W_{kg}^b = W_{ext}^b + C_b \frac{\partial V_{kg}}{\partial P_b^b} - (H_b^0)^T \frac{\partial V_{kg}}{\partial H_b^0}$$

Note that the negative sign in the upper equation comes from $W_b^b = -W_b^{b,0}$
With this knowledge we can write the combined PHS representation

$$\begin{pmatrix} \dot{H}_b^0 \\ \dot{P}_b^b \end{pmatrix} = \begin{pmatrix} 0 & H_b^0 \\ -(H_b^0)^T & C_b \end{pmatrix} \begin{pmatrix} \frac{\partial V_{kg}}{\partial H_b^0} \\ \frac{\partial V_{kg}}{\partial P_b^b} \end{pmatrix} + \begin{pmatrix} 0 \\ W_{ext}^b \end{pmatrix} \tag{2.40}$$

$$T_b^0 = \begin{pmatrix} 0 & Ad_{H_b^0} \end{pmatrix} \begin{pmatrix} \frac{\partial V_{kg}}{\partial H_b^0} \\ \frac{\partial V_{kg}}{\partial P_b^b} \end{pmatrix} \tag{2.41}$$

Dampers

Dampers do not have a state since they do not store energy, they only dissipate it. For sake of completeness note that energy is not "destroyed" in the dampers but transformed in to thermal energy. This can be modelled with a thermal port connected to the environment, since we are only interested in free energy this port is omitted. The easiest way to achieve damping is a linear resistive element R, such that the wrench is directly proportional to twist. Consider for example a body's motion with respect to the inertial frame

$$W_b^b = RT_b^{b,0} \tag{2.42}$$

Or a damper in parallel with spring

$$W_i^j = RT_i^j \tag{2.43}$$

The dissipated co-energy is $E_d = \frac{1}{2}T^T RT$.

2.3 Control of port-Hamiltonian systems

2.3.1 The Intrinsically Passive Controller (IPC)

The IPC is a control architecture based entirely on physical intuition. The structure can be seen in Fig. 2.2, it is exclusively a geometric interconnection of springs, inertias and dampers. The manipulators are connected with spatial springs to the object. The object is modelled by a virtual inertia and its shape is represented by a sphere that does no necessarily coincide with the actual shape. The resulting virtual object is subjected to forces exerted by the manipulator springs and is connected to the environment with another spatial spring. Note that these springs do not exist in reality, they are models to establish compliant behaviour in the controller domain. The simulated manipulator forces are then applied to the real object, in order to transfer the virtual complaint behaviour to the real world.

Starting point for the controller design is the virtual object. It is already given by equation (2.40). Springs to environment and manipulators exert wrenches on the object, in terms of port-Hamiltonian representation they are connected through the control port. The inertias of the manipulators should be equal to their real equivalents. Inertia re-shaping, using a local feed-back linearising compensation for the robot, can lead to infinite energy supply [Str01b] and would thus violate passivity. In the absence of feed-back linearising the robot exhibits its internal dynamics at the end-effector. Therefore the inertia and Coriolis terms of the manipulators are configuration dependent. For the sake of simplicity we will omit this dependency in the following. Moreover the assume the robots to be gravity compensated and use the simpler port-Hamiltonian inertia representation of equation (2.36).

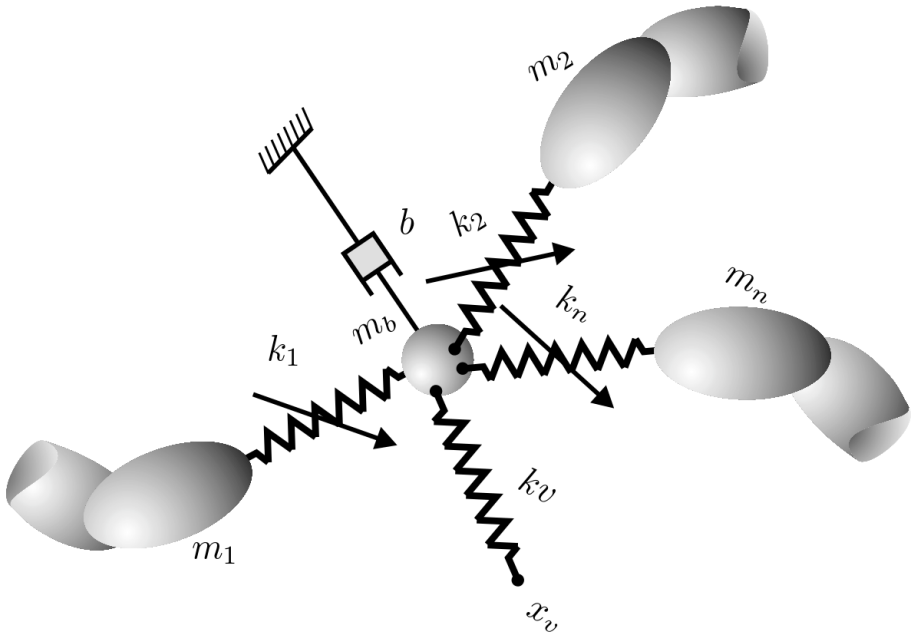

Figure 2.2: Structure of the IPC [SMA99]

From a geometric point of view the object-environment spring directly connects to the virtual object frame Ψ_b. The other end is attached to the desired object position, denoted with frame Ψ_v. This desired object position is the main control objective, by means of this, the motion of the constrained robot-object system is directed. The representation in PHS form is:

$$\dot{H}_v^b = T_v^b H_v^b \tag{2.44}$$

$$W_v^b = \frac{\partial V_{v,b}}{\partial H_v^b}(H_v^b)^T \tag{2.45}$$

Since we already have a spring relation between the virtual object and the inertial frame it is useful to split this spring into two springs: one between desired position and inertial frame and the other between inertial frame and object position: $H_v^b = H_0^b H_v^0$.
Additionally we have a spring connecting object and each of the n-manipulators. As stated before these are variable rest-length springs, to allow for contraction and widening of the formation. Therefore the springs do not connect to the object frame but to a additional supporting frame denoted by $\Psi_v(i)$, which is rigidly attached to Ψ_b. Clearly the frames $\Psi_v(i)$, $i = 1...N$ define the sphere of the virtual object. The springs are situated between the manipulators and the virtual object sphere: $H_{v(i)}^i$ The rest-length is between the sphere and the virtual object frame: $H_b^{v(i)}$, the set-up is illustrated in Fig. 2.3.

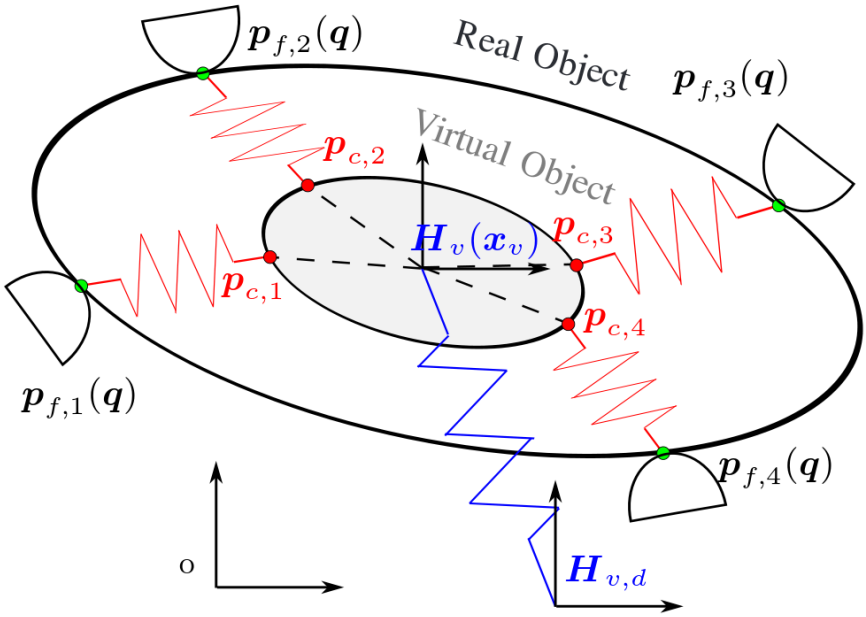

Figure 2.3: Virtual object and springs [WOH08]

Geometric interconnection of springs

To combine all the presented equations to the full system as shown in Fig. 2.2, we express the spring equations with respect to the inertial frame Ψ_0. This is done for the virtual object spring by splitting the twists

$$T_b^v = \begin{pmatrix} Ad_{H_0^v} & -Ad_{H_0^v} \end{pmatrix} \begin{pmatrix} T_b^0 \\ T_v^0 \end{pmatrix}$$

and the wrenches which obey the principle of action and reaction.

$$\begin{pmatrix} W_b^0 \\ W_v^0 \end{pmatrix} = \begin{pmatrix} Ad_{H_0^v}^T \\ -Ad_{H_0^v}^T \end{pmatrix} W_b^v$$

We can write the resulting PHS using the left translation (2.2.1)

$$\dot{H}_b^v = H_b^v \left(\begin{pmatrix} Ad_{H_0^b} & -Ad_{H_0^b} \end{pmatrix} \begin{pmatrix} T_b^0 \\ T_v^0 \end{pmatrix} \right) \tag{2.46}$$

$$\begin{pmatrix} W_b^0 \\ W_v^0 \end{pmatrix} = (H_b^v)^T \left(\begin{pmatrix} Ad_{H_0^b}^T \\ -Ad_{H_0^b}^T \end{pmatrix} \frac{\partial V_{v,b}}{\partial H_b^v} \right) \tag{2.47}$$

This is done with the springs connecting the i-th manipulator to the virtual object sphere in the same manner: $T_{v(i)}^i = Ad_{H_0^i} T_b^0 - Ad_{H_0^i} T_i^0 + Ad_{H_b^i} T_{v(i)}^b$ Note that the twists and wrenches of rest-length changes are not expressed w.r.t. the inertial

frame, since they represent a distinct power port. The resulting PHS for the spring connecting the i-th manipulator to the virtual object using the left translation is

$$\dot{H}^i_{v(i)} = H^i_{v(i)} \left(\begin{pmatrix} Ad_{H_0^{v(i)}} & -Ad_{H_0^{v(i)}} & Ad_{H_b^{v(i)}} \end{pmatrix} \begin{pmatrix} T_b^0 \\ T_i^0 \\ T^b_{v(i)} \end{pmatrix} \right) \tag{2.48}$$

$$\begin{pmatrix} W_b^0 \\ W_i^0 \\ W^b_{v(i)} \end{pmatrix} = (H^i_{v(i)})^T \left(\begin{pmatrix} Ad^T_{H_0^{v(i)}} \\ -Ad^T_{H_0^{v(i)}} \\ Ad^T_{H_b^{v(i)}} \end{pmatrix} \frac{\partial V_{v(i),i}}{\partial H^i_{v(i)}} \right) \tag{2.49}$$

For a simpler presentation we collect the twists and wrenches of the n manipulator springs in a stacked vector notation

$$T_i = \begin{pmatrix} T_1^0 \\ \vdots \\ T_N^0 \end{pmatrix} \ , \ T_{rl} = \begin{pmatrix} T^b_{v(1)} \\ \vdots \\ T^b_{v(N)} \end{pmatrix} \ , \ W_i = \begin{pmatrix} W_1^0 \\ \vdots \\ W_N^0 \end{pmatrix} \ , \ W_{rl} = \begin{pmatrix} W^b_{v(1)} \\ \vdots \\ W^b_{v(N)} \end{pmatrix}$$

As a result we can combine all springs in a PHS

$$\underbrace{\begin{pmatrix} \dot{H}^1_{v(1)} \\ \vdots \\ \dot{H}^N_{v(N)} \\ \dot{H}^v_b \end{pmatrix}}_{\dot{x}_s} = \begin{pmatrix} \phi_b & \phi_v & \phi_i & \phi_{rl} \end{pmatrix} \begin{pmatrix} T_b^0 \\ T_v^0 \\ T_i \\ T_{rl} \end{pmatrix} \tag{2.50}$$

$$\begin{pmatrix} W_b^0 \\ W_v^0 \\ W_i \\ W_{rl} \end{pmatrix} = \begin{pmatrix} \phi_b^T \\ \phi_v^T \\ \phi_i^T \\ \phi_{rl}^T \end{pmatrix} \frac{\partial V_s}{\partial x_s} \tag{2.51}$$

Where $V_s = V_{v,b} + \sum_{i=1}^{N} V_{v(i),i}$ is the sum of energy stored in the springs. The geometric interconnection of the springs is described by the matrices $\phi_b, \phi_v, \phi_i, \phi_{rl}$, their structure can be easily derived from the previous equations where every spring is described

$$\phi_b = \begin{pmatrix} H^1_{v(1)} Ad_{H_0^{v(1)}} \\ \vdots \\ H^N_{v(N)} Ad_{H_0^{v(N)}} \\ H^v_b Ad_{H_0^b} \end{pmatrix} \ , \ \phi_v = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -H^v_b Ad_{H_0^b} \end{pmatrix}$$

,

$$\phi_i = \begin{pmatrix} -H^1_{v(1)} Ad_{H_0^{v(1)}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -H^N_{v(N)} Ad_{H_0^{v(N)}} \\ 0 & \cdots & 0 \end{pmatrix}$$

,

$$\phi_{rl} = \begin{pmatrix} H^1_{v(1)} Ad_{H_b^{v(1)}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & H^N_{v(N)} Ad_{H_b^{v(N)}} \\ 0 & \cdots & 0 \end{pmatrix}$$

Interconnection of springs and inertias

Having obtained a complete geometric description of the springs we can now add the inertias. The virtual object is attached to both, the environment spring and the manipulator springs. In the simulation the external forces acting on the inertia exclusively originate from these springs, thus $W^b_{ext} = -Ad^T_{H^0_b} W^0_b$. The wrench acting on the i-th manipulator stems from the i-th manipulator spring: $W^i_{ext} = -Ad^T_{H^0_i} W^0_i$. We can also derive from equation (2.40) that $T^0_b = Ab_{H^0_b} \frac{\partial V_{kg}}{\partial P^b_b}$. Equivalently the i-th manipulator wrench can be expressed as $T^0_i = Ad_{H^0_i} \frac{\partial V_{k,i}}{\partial P^i_i}$. With this knowledge we can write the PHS representation of the complete control model. In order to keep the formulation compact we us the stacked state vector of the object inertia $x_b = ((H^0_b)^T \ (P^b_b)^T)^T$, the stacked vector of the spring states x_S (defined in eq. (2.50)) and the vector of the manipulator inertia states $x_M = ((P^1_1)^T \ \cdots \ (P^N_N)^T)^T$

$$\begin{pmatrix} \dot{x}_b \\ \dot{x}_S \\ \dot{x}_M \end{pmatrix} = \begin{pmatrix} J_B & -\phi^T_B & 0 \\ \phi_B & 0 & Ad_{H^0_i}\phi_i \\ 0 & -\phi^T_i Ad^T_{H^0_i} & C_M \end{pmatrix} \begin{pmatrix} \frac{\partial V_c}{\partial x_b} \\ \frac{\partial V_c}{\partial x_S} \\ \frac{\partial V_c}{\partial x_M} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ \phi_v & \phi_{rl} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} T^0_v \\ T_{rl} \end{pmatrix} \tag{2.52}$$

$$\begin{pmatrix} W^0_v \\ W_{rl} \end{pmatrix} = \begin{pmatrix} 0 & \phi^T_v & 0 \\ 0 & \phi^T_{rl} & 0 \end{pmatrix} \begin{pmatrix} \frac{\partial V_c}{\partial x_b} \\ \frac{\partial V_c}{\partial x_S} \\ \frac{\partial V_c}{\partial x_M} \end{pmatrix} \tag{2.53}$$

Where $V_c = V_{v,b} + \sum_{i=1}^{N} [V_{v(i),i} + V_{k,i}] + V_{kg}$ is the combined energy function. J_B is given in equation (2.40) and C_M combines the Coriolis terms of all manipulators. The geometric interconnection matrix ϕ_B is composed by

$$\phi_B = \begin{pmatrix} 0 \\ Ad_{H^0_b}\phi_b \end{pmatrix}$$

The system is lossless, since no damping terms have been added. The reference inputs are T_v and T_{rl} the control output is $-\phi_i \frac{\partial V_c}{\partial x_s}$.

2.4 Performance Comparison of control strategies

To evaluate the presented controllers in an objective way, they are implemented in *Simulink* an compared in terms of:

- Trajectory tracking
- Dynamic behaviour
- Internal Forces

2.4.1 Internal impedance control with feed-forward of the object dynamics

De Pascali et al. [DPEZ+15] present combination of impedance control on manipulator/object level and feed-forward object dynamics. The internal impedance control relation (between object and manipulator) encompasses a spring and a parallel damper, inertia is used to feed-forward the desired acceleration:

$$M_i \ddot{x}_i^d + D_i(\dot{x}_i^d - \dot{x}_i) + K_i(x_i^d, x_i) = h^x \tag{2.54}$$

I know I have to use consistent notation and explain the variables, comes later ;) This avoids the necessity of either measuring manipulator acceleration or contact force. Object dynamics is represented with a feed-forward term, mapped to the manipulators with a weighted pseudoinverse G^+ of the grasp matrix:

$$h^d = G^+(M_o \ddot{x}_o^d + C_o \dot{x}_o^d + g_o) \tag{2.55}$$

Note that this term is not an impedance relation and does not adjust if the environment hinders motion. The combined control law is $h^\Sigma = h^x + h^d$.
The set-up consists of four manipulators, distributed symmetrically around the object. In the first case translation in x-direction commanded. Results in Fig. 2.4 show good tracking behaviour: no position errors in steady state and only small deviations from the desired values during transient phase. No internal stress is exerted on the object. Due to fast translation high manipulator forces occur.
In the second test case the object is rotated around the z-axis. This is done at a significantly lower speed of at most 1 rad/s, thus the manipulator forces are smaller, desired and actual object trajectory cannot be distinguished in Fig. 2.5. However some small internal forces can be seen. Interestingly they are proportional to the simulation step size (running *Simulink's ode3* solver), i.e. a ten-times smaller step size gives ten-times smaller internal forces. Internal forces are calculated based on the geometry in the last simulation step. The correlation between step size and values indicates that these forces are rather due to the discrete nature of the simulation than of the control law generating internal stress. Simulation of the constrained system dynamics as well as calculation of internal wrench is done as described in [EH15b].

2.4.2 Internal and external impedance based reference trajectory generation

Caccavale and Villani [CV01] combine both internal and external impedance control. With external we mean a compliant relation between object and (external) environment. The architecture is cascaded, consisting of a two level reference trajectory generation and a motion control loop below. On top-level an impedance relation between object and environment is used to generate a compliant trajectory subject to environmental forces:

$$\alpha M_o(\ddot{x}_o^d - \ddot{x}_o^r) + D_o(\dot{x}_o^d - \dot{x}_o^r) + K_o(x_o^d, x_o^r) = h_{env} \tag{2.56}$$

The constant α scales the object inertia proportionally to a desired value. The control output is the reference object acceleration $\ddot{x}_o^r$, h_{env} is an input. This is sometimes called admittance control, admittance being the inverse of impedance. h_{env} has to be known, but is not easily measured in a practical set-up. Recalling (**??**) the environmental forces can be expressed as:

$$h_{env} = M_o\ddot{x}_o^r + C_o\dot{x}_o^r + g_o - G^\dagger h \tag{2.57}$$

Herein $G^\dagger$ is a generalized inverse of the grasp matrix, selecting the motion inducing components from the measured contact wrench h. $\dot{x}_o^r, x_o^r$ are calculated from $\ddot{x}_o^r$ by integration. From the compliant object trajectory $(\ddot{x}_o^r, \dot{x}_o^r, x_o^r)$ the desired trajectories of the manipulator $(\ddot{x}_i^d, \dot{x}_i^d, x_i^d)$ using the kinematic constraints. The reference manipulator trajectory, enforcing compliant behaviour between manipulators and object, is calculated from manipulator dynamics and internal forces:

$$M_i(\ddot{x}_i^d - \ddot{x}_i^r) + D_i(\dot{x}_i^d - \dot{x}_i^r) + K_i(x_i^d, x_i^r) = VV^\dagger h \tag{2.58}$$

The control output is the reference acceleration of the i-th manipulator $\ddot{x}_i^r$, $\dot{x}_i^r, x_i^r$ are obtained from integration. These variables are the inputs the inner motion control loop (PD-type). The strategy of compliant trajectories allows for high gains in the motion controller. Knowledge of object dynamics and measurement of the contact wrenches is required.
Results for translational motion (see Fig. 2.6) are very similar to that of the previous control scheme.
When it comes to rotation, higher internal forces can be observed in Fig. 2.7. In this case they are not influenced by numerical parameters of the simulation.
This architecture in contrast to the previous makes use of measured contact wrenches. Contact wrench as measured can be obtained from the constrained system simulation. As described in [CM08] this wrench is than decomposed by kineostatic filters in internal and external components. In this simulation this not cancel out undesired internal stress but magnifies it: when the contact wrench is not fed back and set to zero in the internal force impedance controller, results are slightly better. Note that the simulation represents an ideal case, where all parameters are exactly known

and no deviations in grasp positions occur. Behavior in a real experiment may be different and this observation does not mean that the kineostatic-filtered feed-back of contact wrench is unjustified in general.

2.4.3 Intrinsically Passive Controller (IPC)

Introduced by Stramigioli [Str01b] and implemented by Wimböck et al. [WOH08], the architecture has been detailed throughout this work. In contrast to Stramigioli dampers are used in parallel with the manipulator springs and in contrast to Wimböck all springs have 6-DoF. Simulation results for pure translation can be seen in Fig. 2.8. The object trajectory falls slightly behind the reference input, the dynamic behaviour is inferior to the previous approaches. Despite very stiff springs the magnitude of force seen in the previous approaches is not reached. No internal forces can be observed.
When it comes to rotation (see Fig. 2.9) again the dynamic behaviour falls short of the two other approaches, while the manipulator wrench is higher. Significant internal wrenches are present, they amount up to 50% of the manipulator wrenches.

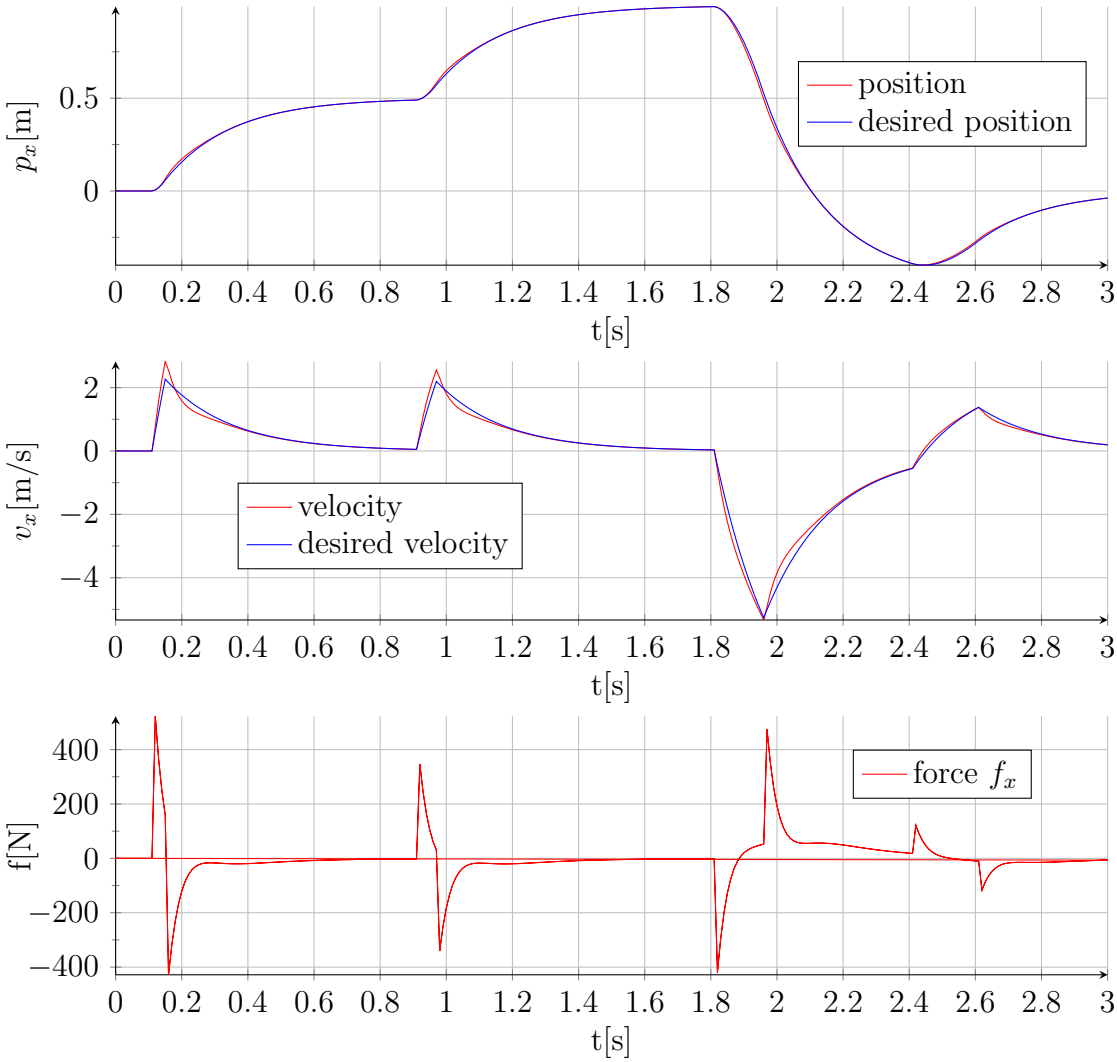

Figure 2.4: Internal force impedance control with feed-forward of the object dynamics: Translation; Graphs from top: Position (desired/actual), Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

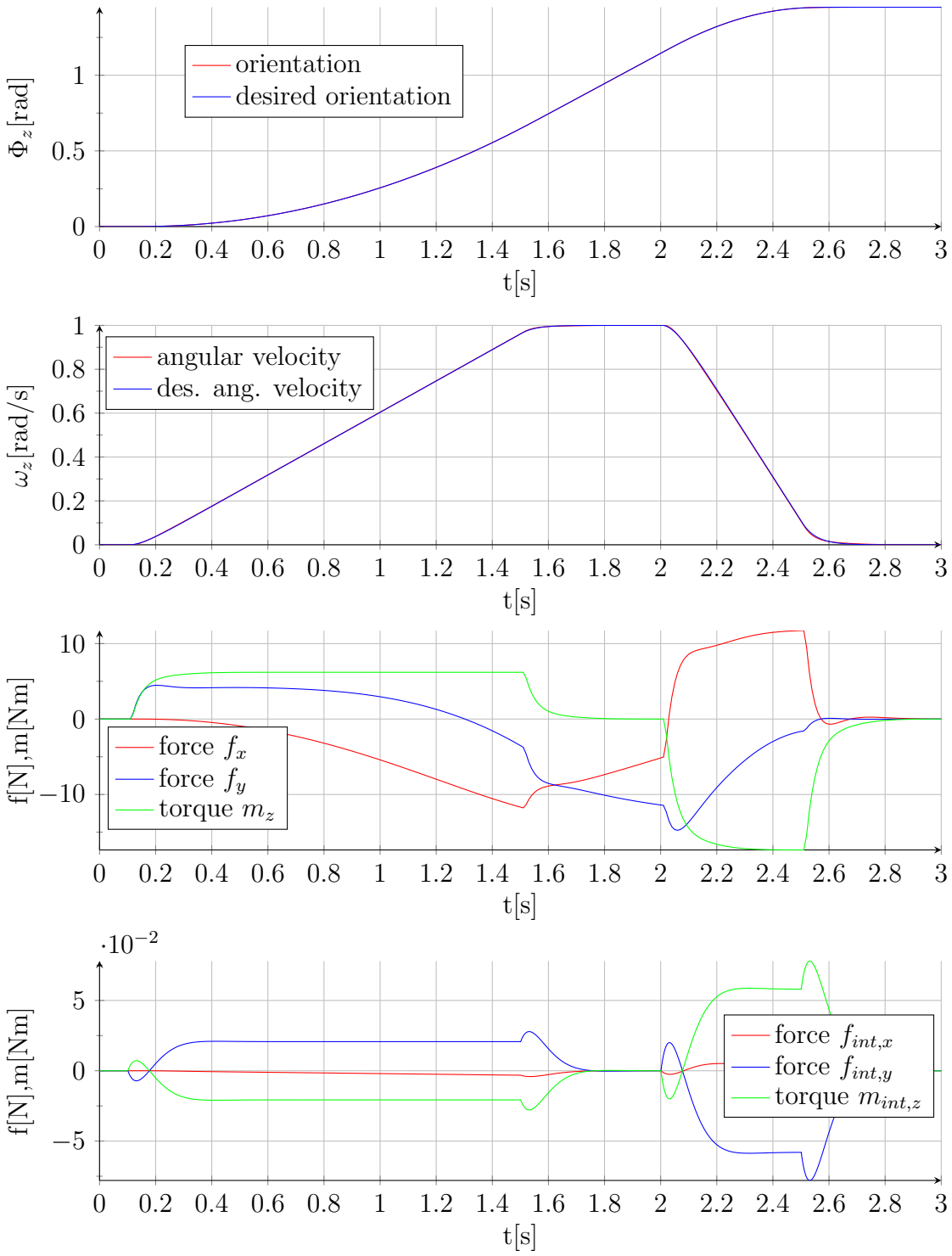

Figure 2.5: Internal force impedance control with feed-forward of the object dynamics: Rotation; Graphs from top: Orientation (desired/actual), Angular Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

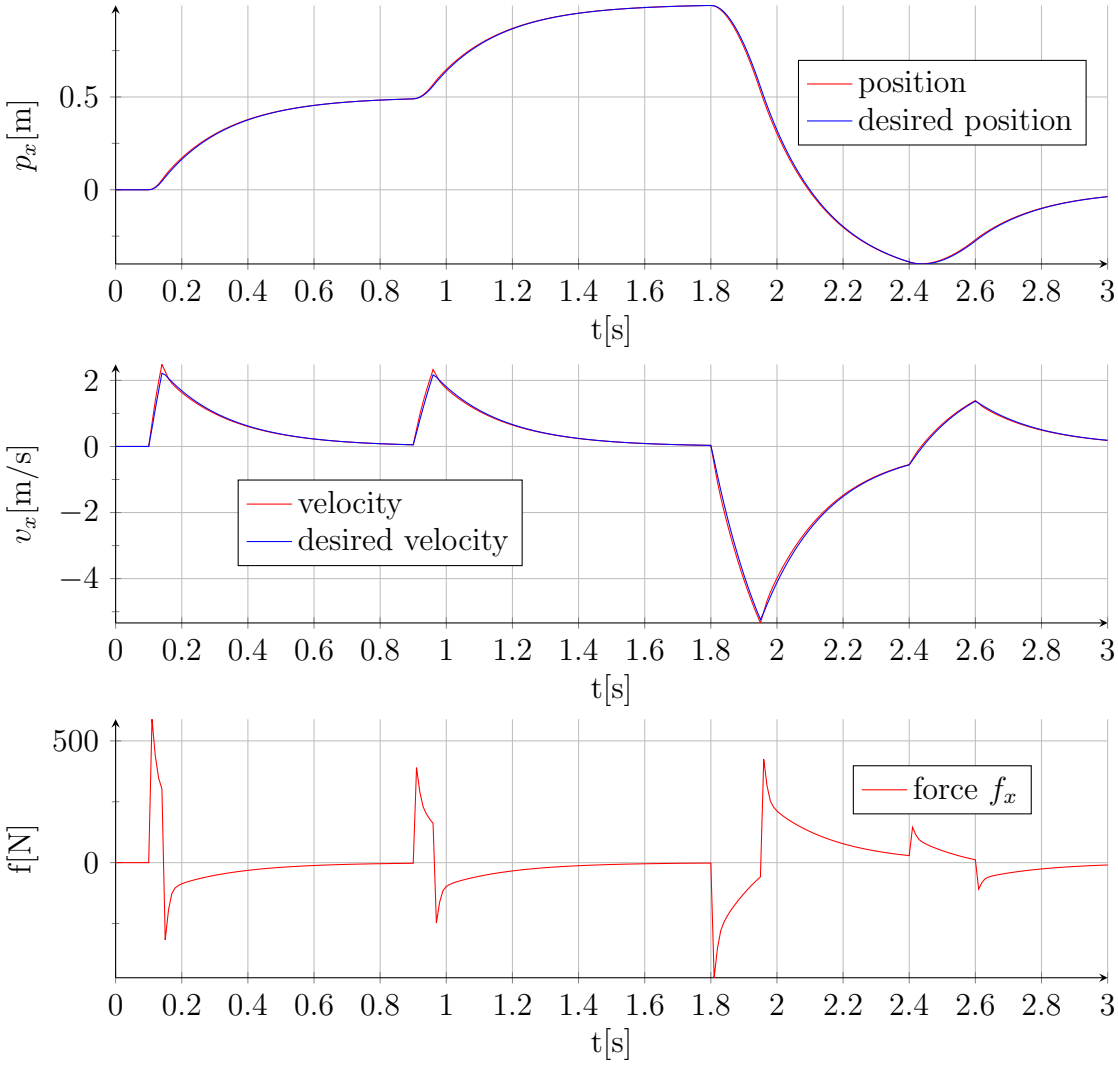

Figure 2.6: Internal and external impedance based reference trajectory generation: Translation; Graphs from top: Position (desired/actual), Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

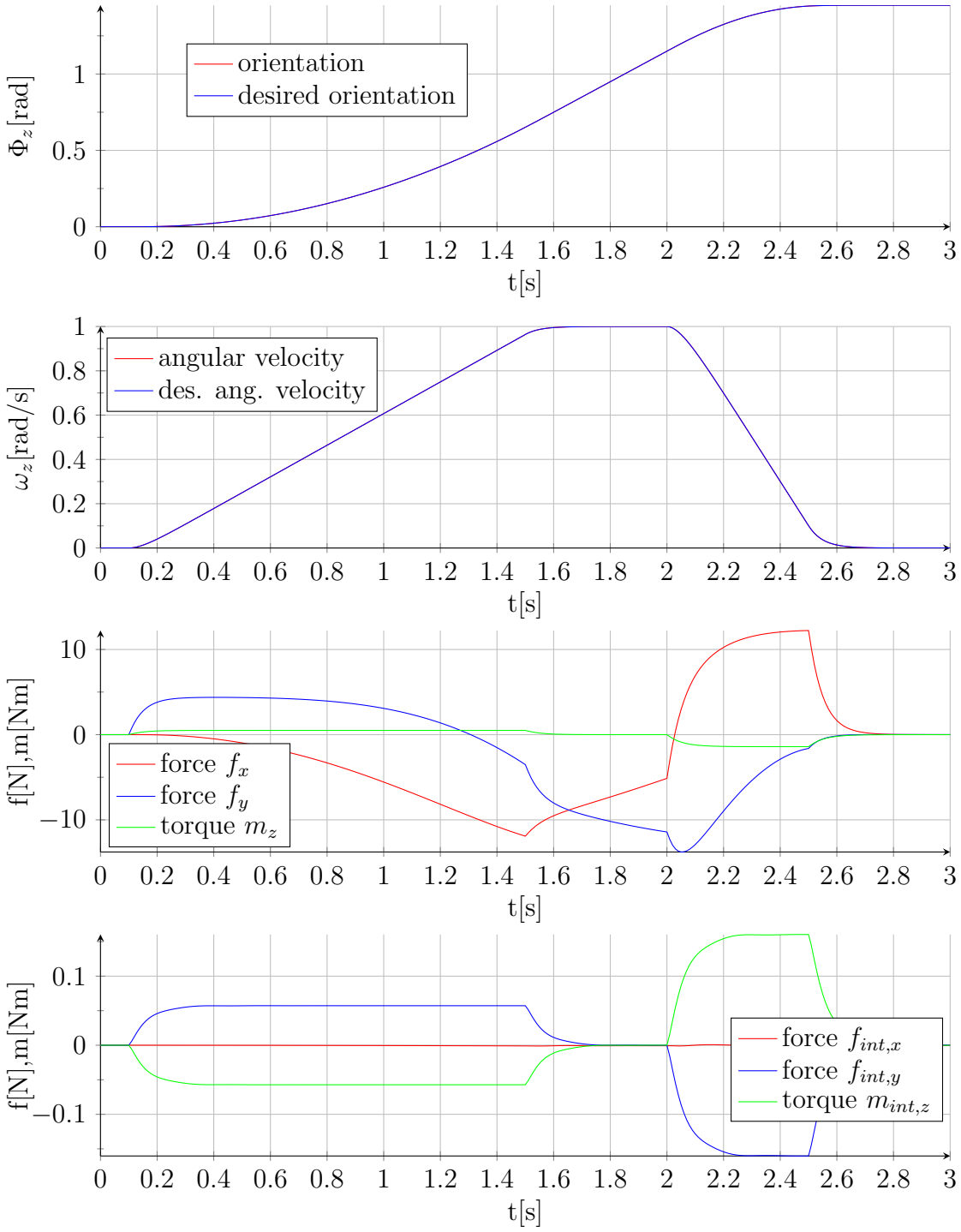

Figure 2.7: Internal and external impedance based reference trajectory generation: Rotation; Graphs from top: Orientation (desired/actual), Angular Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

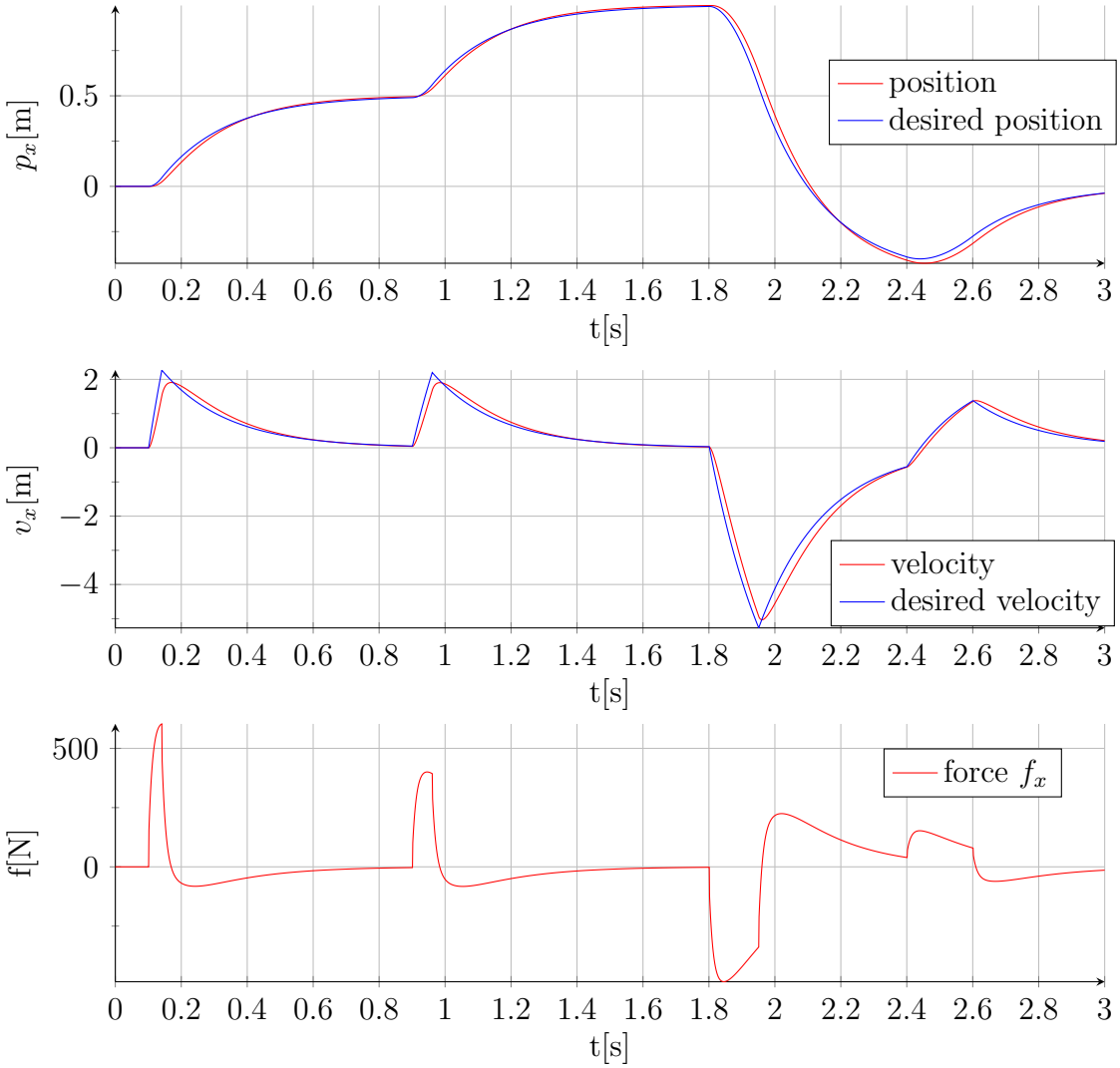

Figure 2.8: Intrinsically Passive Controller: Translation; Graphs from top: Position (desired/actual), Velocity (desired/actual), Force exerted by one manipulator

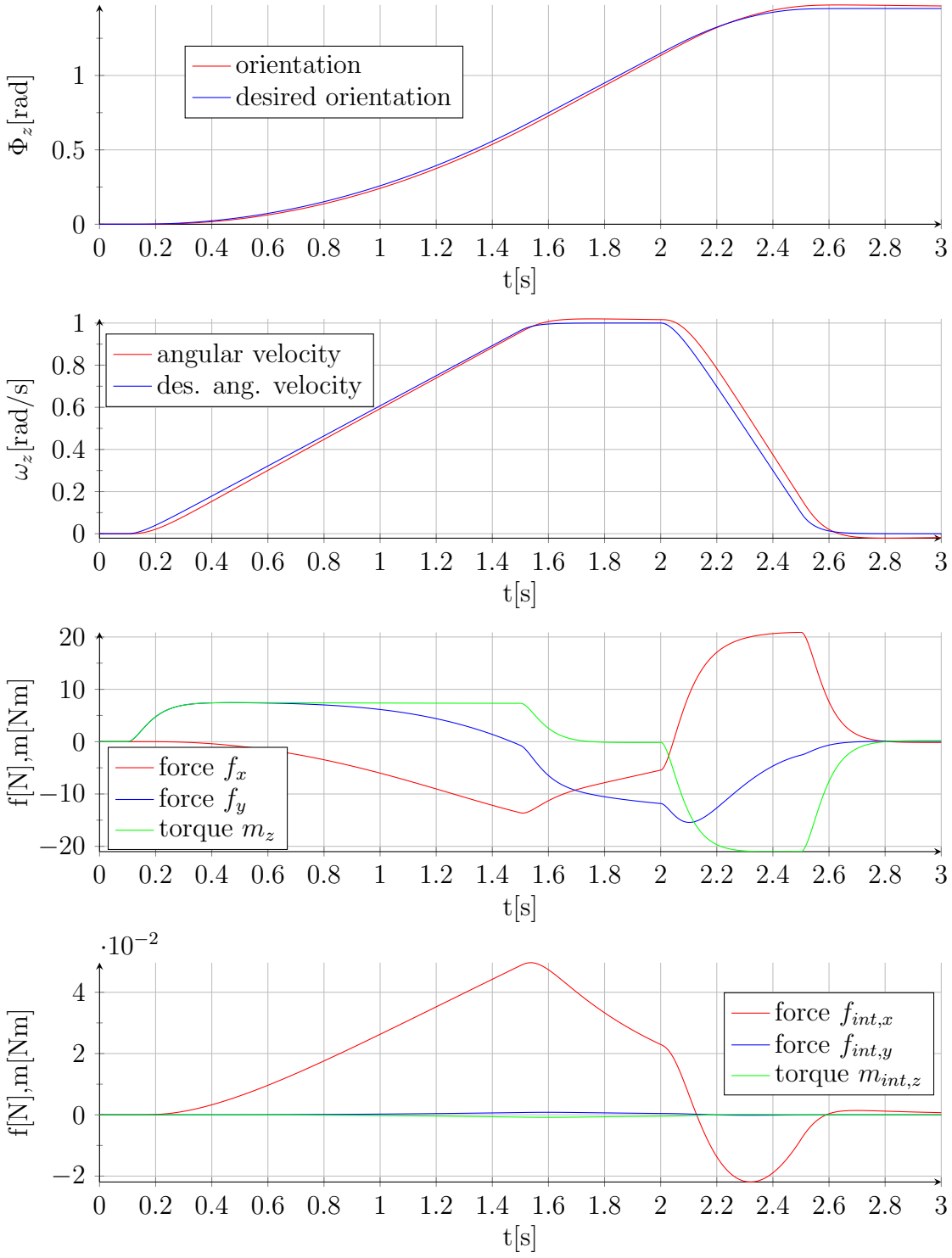

Figure 2.9: Intrinsically Passive Controller: Rotation; Graphs from top: Orientation (desired/actual), Angular Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

Chapter 3

Conclusion

List of Figures

Bibliography

[BH96] R.G. Bonitz and T.C. Hsia. Internal force-based impedance control for cooperating manipulators. *Robotics and Automation, IEEE Transactions on*, 12(1):78–89, 1996.

[BHM96] M. Buss, H. Hashimoto, and J.B. Moore. Dextrous hand grasping force optimization. *Robotics and Automation, IEEE Transactions on*, 12(3):406–418, 1996.

[CCMV08] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-dof impedance control of dual-arm cooperative manipulators. *Mechatronics, IEEE/ASME Transactions on*, 13(5):576–586, 2008.

[CM08] F. Caccavle and Uchiyama M. Cooperative Manipulation. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, chapter 29, pages 701–718. Springer Berlin Heidelberg, May 2008. ISBN 978-3-540-23957-4.

[CV01] F. Caccvale and L. Villani. An impedance control strategy for cooperative manipulation. *Advanced Intelligent Mechatronics, IEEE/ASME International Conference on*, 1:343–348, 2001.

[DPEZ+15] L. De Pascali, S. Erhart, L. Zaccarian, F. Biral, and S. Hirche. A decoupling scheme for force control in cooperative multi-robot manipulation tasks. *Manuscript submitted for publication*, 2015.

[EH15a] S. Erhart and S. Hirche. Internal force analysis and load distribution for cooperative multi-robot manipulation. *Robotics, IEEE Transactions on*, 31(5):1238–1243, 2015.

[EH15b] S. Erhart and S. Hirche. Model and analysis of the interaction dynamics in cooperative manipulation tasks. *Manuscript submitted for publication*, 2015.

[GFS+14] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo. The flying hand: A formation of UAVs for cooperative aerial telemanipulation. *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 4335–4341, May 2014.

[Goe52] R.C. Goertz. Fundamentals of general-purpose remote manipulators. *Nucleonics*, 10(11):36–45, 1952.

[HKDN13] D. Heck, D. Kostic, A. Denasi, and H. Nijmeijer. Six-dof impedance control of dual-arm cooperative manipulators. *Control Conference (ECC), 2013 European*, pages 2299–2304, July 2013.

[Hog84] N. Hogan. Impedance control: An approach to manipulation. *American Control Conference*, pages 304–313, June 1984.

[Hsu93] P. Hsu. Coordinated control of multiple manipulator systems. *Robotics and Automation, IEEE Transactions on*, 9(4):400–410, 1993.

[HTL00] L. Han, J.C. Trinkle, and Z.X. Li. Grasp analysis as linear matrix inequality problems. *Robotics and Automation, IEEE Transactions on*, 16(6):663–674, 2000.

[LS05] Dongjun Lee and M.W. Spong. Bilateral teleoperation of multiple cooperative robots over delayed communication networks: Theory. pages 360–365, April 2005.

[LTD14] Mitsubishi Heavy Industries LTD. "meister" remote control robot completes demonstration testing at fukushima daiichi nuclear power station, 2014. Press Information 1775, February 20, 2014; Online, accessed January 13, 2016. URL: `https://www.mhi-global.com/news/story/1402201775.html`.

[MT93] M.J. Massimino and Sheridian T.B. Sensory substitution for force feedback in teleoperation. *Presence, MIT Press Journals*, 2(4):344–352, 1993.

[NPH08] G. Niemeyer, C. Preusche, and G. Hirzinger. Telerobotics. In Bruno Siciliano and Oussama Khatib, editors, *Springer Handbook of Robotics*, chapter 31, pages 741–757. Springer Berlin Heidelberg, May 2008. ISBN 978-3-540-23957-4.

[SC92] S.A. Schneider and R.H. Cannon. Object impedance control for cooperative manipulation: theory and experimental results. *Robotics and Automation, IEEE Transactions on*, 8(3):383–394, 1992.

[SD01] S. Stramigioli and V. Duindam. Variable spatial springs for robot control applications. 4:1906–1911, 2001.

[She92] T.B. Sheridian. *Telerobotcis, Automation and Human Supervisory Control*. MIT Press, Cambridge, MA, 1992.

[SMA99] S. Stramigioli, Claudio Melchiorri, and S. Andreotti. A passivity-based control scheme for robotic grasping and manipulation. 3:2951–2956, 1999.

[SMH15] D. Sieber, S. Music, and S. Hirche. Multi-robot manipulation controlled by a human with haptic feedback. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2440–2446, Sep 2015.

[SMP14] S. Scheggi, F. Morbidi, and D. Prattichizzo. Human-robot formation control via visual and vibrotactile haptic feedback. *Haptics, IEEE Transactions on*, 7(4):499–511, Oct 2014.

[Str01a] Stefano Stramigioli. Geometric modeling of mechanical systems for interactive control. In Alfonso Banos, Francoise Lamnabhi-Lagarrigue, and FranciscoJ. Montoya, editors, *Advances in the control of nonlinear systems*, volume 264 of *Lecture Notes in Control and Information Sciences*, pages 309–332. Springer London, 2001.

[Str01b] Stefano Stramigioli. *Modeling and IPC Control of Interactive Mechanical Systems: A Coordinate-Free Approach*. Springer-Verlag London, London, UK, 2001.

[Str15] Stefano Stramigioli. Energy-aware robotics. In M. K. Camlibel, A. A. Julius, R. Pasumarthy, and J. Scherpen, editors, *Mathematical Control Theory I*, volume 461 of *Lecture Notes in Control and Information Sciences*, pages 37–50. Springer London, 2015.

[vdS06] Arjan van der Schaft. Port-hamiltonian systems: an introductory survey. In M. Sanz-Sole, J. Soria, J.L. Varona, and J. Verdera, editors, *Proceedings of the International Congress of Mathematicians Vol. III: Invited Lectures*, pages 1339–1365, Madrid, Spain, 2006. European Mathematical Society Publishing House (EMS Ph).

[WKD92] J. Wen and K. Kreutz-Delgado. Motion and force control of multiple robotic manipulators. *Automatica*, 28(4):729–743, 1992.

[WOH06] T. Wimboeck, C. Ott, and G. Hirzinger. Passivity-based object-level impedance control for a multifingered hand. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4621–4627, Oct 2006.

[WOH08] T. Wimboeck, C. Ott, and G. Hirzinger. Analysis and experimental evaluation of the intrinsically passive controller (ipc) for multifingered hands. *Robotics and Automation, IEEE International Conference on*, pages 278–284, May 2008.

License