

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: 11.01.2016

Abgabe: 01.04.2016

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

Abstract

A short (1–3 paragraphs) summary of the work. Should state the problem, major assumptions, basic idea of solution, results. Avoid non–standard terms and acronyms. The abstract must be able to be read completely on its own, detached from any other work (e.g., in collections of paper abstracts). Don't use references in an abstract.

Zusammenfassung

Hier die deutschsprachige Zusammenfassung

Contents

1	Introduction	5
1.1	Problem Statement	7
1.2	Related Work	8
1.2.1	Control schemes for rigidly connected manipulators	9
1.2.2	Friction grasps in Cooperative Manipulation	12
1.2.3	Determination of desired internal forces	14
2	Main Part	17
2.1	port-Hamiltonian systems	17
2.1.1	Hamiltonian description of mechanical systems	17
2.1.2	Interconnection of port-Hamiltonian systems	19
2.2	3D-space modelling of mechanical systems	21
2.2.1	Euclidean space and motions	21
2.2.2	Dynamics of physical components	24
2.3	Control of port-Hamiltonian systems	27
2.3.1	The Intrinsically Passive Controller (IPC)	27
2.4	Performance Comparison of control strategies	31
2.4.1	Internal force impedance control with feed-forward of the ob- ject dynamics [DPEZ ⁺ 15]	31
2.4.2	Internal and external impedance based reference trajectory generation [CV01]	32
3	Conclusion	35
	List of Figures	37
	Bibliography	39

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. Therefore different control architectures exist, which can be classified by the level of autonomy of the robot system. In an direct master-slave approach each robot is controlled independently by a human operator. This is impractical due to formation requirements. Lee [LS05] treats



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

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 she/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, one refers to bilateral telemanipulation. The human is part of the closed loop, since her/his behaviour is directly influenced by the experienced force-feedback. For a remotely located set-up, the time delays in communication may lead to undesired motions, stability of the human-in-the-loop system becomes an issue. Tactile or visual feedback eases this problem, since they are non-reactive. This means they do not induce operator movements reacting to a back-driving force [MT93]. A control architecture that leaves further autonomy to the robot system is Supervisory control, described by a closed local, autonomous control loop. It interacts with the operator by continuously sending information about the state and periodically receiving commands [She92].

However a difficulty is the control of the set-up which possess extra degrees-of-freedom. There must be an intuitive way to control the formation of robots. The control scheme shapes each robot's motion to commit to the common task. A disadvantage of having a remotely located operator is an inevitable delay in the control loop. This restricts the operators possibilities to react to unexpected situations.

The major challenge in cooperative manipulation is to simultaneously control the forces between the manipulators and the object, so to say the internal stress exerted on the object, and the forces between the object and the environment, i.e. the absolute motion of the object. Strategies that map force and velocity of the object to the manipulators have to be considered.

One important assumption is the nature of the connection between manipulator and object. It can be seen as rigid or more general as friction-based. With a rigid fixture the grasp is always stable, on the contrary friction grasps need to be actively stabilized. To comply with the friction constraints a contact normal force has to be set, dependent on the tangential components present during manipulation.

Towards a versatile usage of cooperative manipulation it is vital to let the human operator control the robotic system intuitively and naturally. A promising approach is gesture control. Gioioso et al. [GFS⁺14] demonstrated this for a formation of unmanned aerial vehicles. They are able to control opening and closing of the grasp. In a grasped state the objects's pose and the grasping forces can be changed, just by tracking gesture of a single human hand. Directly integrating the human into the control scheme makes use of the visual tracking capacities of the human and eases operation in unknown environments. Regardless of the operator being present on-site with the robots or not, hand tracking and analysis requires a certain time. This delay in the control loop can cause instability. The problem can be modelled as a transmission line with delay and treated with methods known from tele-operation. The human takes the role of a supervisor, representing a high-level part of the control scheme. He is responsible for planning, choosing a trajectory and if necessary setting of grasping forces. On a local low-level robots need to coordinate their motion compliant with the commanded trajectory and adjust their formation to realize the desired internal forces. For non-rigid fixtures the local robot controller requires the position of the other manipulators. The local communication between the robots causes delay, which has to be treated with tele-operation techniques in order to ensure stability.

Haptic feedback??

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 to manipulate (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 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 co-workers use force/torque sensors at the manipulators to establish compliant object environment interaction, for this purpose Heck [HKDN13] assumes to have an

exact model of the environment. Stramigioli's IPC implements a compliant relation between virtual object and environment.

Control strategies suitable for Cooperative Manipulation can be classified dependent on the nature of the manipulator-object connection: rigid-fixture and friction-grasps. An import group within the former are the hybrid position/force control approaches. Control action is decomposed in a motion control loop, accounting for the tracking of the desired trajectory for the overall system and a force control loop to control the internal forces on the object. A pre-defined selection matrix determines for every workspace coordinate whether it is position or force controlled. Controlling the robot on joint level, the necessary torques can be computed using a PD plus gravity compensation control scheme. The desired joint positions and velocities are computed by inverse kinematics [WKD92]. Object level control architectures often use feedback linearisation of the non-linear manipulator dynamics to achieve a linear and decoupled control of motion and grasping force [Hsu93]. Force control is only meaningful if contact between manipulator and object. Motion control leads to unbounded forces if the motion direction is not consistent with the constraints, since force in this direction is not controlled. The transition between contact and no contact during grasping cannot be handled with these approaches. For these reasons in the following only the better suited impedance control schemes are considered. Impedance control, introduced by Hogan [Hog84], is a well known strategy to avoid unbounded forces when it comes to interaction of manipulators with objects/environment. It does not control one state variable (position, velocity, force) dynamically enforces a relation between forces and system state. Contact directions do not have to be pre-specified. The controller is stable independently of the contact state. A physical analogy the impedance control scheme is a spring-mass-damper system.

1.2.1 Control schemes for rigidly connected manipulators

A rigid fixture between manipulator and object means that both forces and moments are exchanged at the grasp points: all translational and rotational motion components can be transmitted through the contact points. The rigid connections are expressed by a number of kinematic constraints, which reduce the Degrees-of-Freedom (DoF) of the constrained system. The attempt to move in the constrained directions results in internal loading of the object. Therefore trajectories of the manipulators have to be generated to avoid internal stress on the object. Assuming a rigid connection normally no grasping force is needed to stabilize the grasp, i.e. internal forces are undesired because possibly harmful to the object. Nevertheless some papers propose possibilities to specify or control internal forces.

Among the first to use impedance control in cooperative manipulation were Schnei-

der and Cannon [SC92]. They replaced the actual object by a virtual mass and connected it to the environment with a spring and a parallel damper. The resulting impedance equation is:

$$M_{o,d}(\ddot{x}_o^d - \ddot{x}_o) + D_o(\dot{x}_o^d - \dot{x}_o) + K_o(x_o^d, x_o) = h_{env} \quad (1.1)$$

Here $M_{o,d}$ denotes the inertia of the virtual object, D_o is the damping along the spring, which is represented by $K_o(x_d, x)$. $x_o = (p_o^T, Q_o^T)^T$ is the stacked vector of position and orientation of the object. x^d denotes a desired quantity. The external wrench, exerted by the environment on the object, $h_{env} = (f_{env}^T, t_{env}^T)^T$ consists of force and torque vectors. The dynamics of the virtual object is then given as:

$$M_{o,d}\ddot{x} = -h_{env} + h_o^\Sigma = -h_{env} + M_{o,d}\ddot{x}_d + D_o(\dot{x}_d - \dot{x}) + K_o(x_d, x) \quad (1.2)$$

The resulting impedance control output h_o^Σ is then mapped to the manipulators using a generalized inverse of the grasp matrix:

$$h^\Sigma = G^\dagger h_o^\Sigma \quad (1.3)$$

Performance of this strategy depends on choosing the virtual inertia close to the actual, i.e. good knowledge of the object is required. Furthermore gravity compensation would be necessary to model the full object dynamics. In general the equation of motion for a rigid object is:

$$M_o\ddot{x}_o + C_o\dot{x}_o + g_o = h_{env} + h^\Sigma \quad (1.4)$$

Wherein M_o is the actual object inertia, C_o is the Matrix accounting for the Coriolis force and g_o is the gravity. For notational convenience the pose dependencies of the parameter matrices, e.g. $M_o(x_o)$, are omitted.

Bonitz and Hsia [BH96] propose to model each manipulator as an impedance relation. Thereby a relation between manipulator dynamics and internal forces on the object is enforced:

$$M_i(\ddot{x}_i^d - \ddot{x}_i) + D_i(\dot{x}_i^d - \dot{x}_i) + K_i(x_i^d, x_i) = h_{int} \quad (1.5)$$

$M_i, D_i, K_i(x_i^d, x_i)$ are the desired inertia, damping and stiffness representations of the i -th manipulator. The internal wrench $h_{int} = VV^\dagger h$ is calculated from the measured contact wrench h . V is a basis of the nullspace of the grasp matrix, V^\dagger denotes the pseudoinverse. No assumptions on the object are necessary because it is not considered in the control scheme.

The first approach ensures compliant object-environment interaction, while the second limits internal forces even in case of end-effector displacements. The both have been combined by Caccavale and Villani [CV01]. Their control 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:

$$\alpha M_o(\ddot{x}_o^d - \ddot{x}_o^r) + D_o(\dot{x}_o^d - \dot{x}_o^r) + K_o(x_o^d, x_o^r) = h_{env} \quad (1.6)$$

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 (1.4) the environmental forces can be expressed as:

$$h_{env} = M_o \ddot{x}_o^r + C_o \dot{x}_o^r + g_o - G^\dagger h \quad (1.7)$$

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:

$$M_i(\ddot{x}_i^d - \ddot{x}_i^r) + D_i(\dot{x}_i^d - \dot{x}_i^r) + K_i(x_i^d, x_i^r) = VV^\dagger h \quad (1.8)$$

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.

In a frequently cited follow-up paper Caccavale et al. [CCMV08] combine the concept of *geometrically consistent stiffness* with the same architecture. This allows for selective stiffness behaviour along different directions.

Recently Heck et al. [HKDN13] worked with this architecture, they assume knowledge of the environment instead of the object. The environmental wrench is then derived as:

$$h_{env} = \Sigma D_{env}(\dot{x}_o^d - \dot{x}_o^r) + \Sigma K_{env}(x_o^d, x_o^r) \quad (1.9)$$

It is assumed here that the environment has a constant stiffness and damping $(K_{env}(x_o^d, x_o^r), D_{env})$. Σ is a diagonal selection matrix modelling the direction of contact.

A combination of impedance control on manipulator/object level and feed-forward object dynamics is presented in [DPEZ⁺15]. The manipulators are modelled as springs and dampers, inertia is used to feed-forward the desired acceleration:

$$M_i \ddot{x}_i^d + D_i(\dot{x}_i^d - \dot{x}_i) + K_i(x_i^d, x_i) = h^x \quad (1.10)$$

This avoids the necessity of either measuring manipulator acceleration or contact force. Object dynamics is represented with feed-forward term, mapped to the manipulators with a weighted pseudoinverse G^+ of the grasp matrix:

$$h^d = G^+(M_o \ddot{x}_o^d + C_o \dot{x}_o^d + g_o) \quad (1.11)$$

Note that this term is not an impedance relation and does not adjust if the environment hinders motion. The combined control law is $h^\Sigma = h^x + h^d$.

1.2.2 Friction grasps in Cooperative Manipulation

The assumption of rigid-fixture between object and manipulators only holds if the manipulators are equipped with appropriate grasping devices. In analogy to a human this device would be fingers wrapping around an inhomogeneity on the object, to achieve a rigid grasp. For an even object (e.g. a ball) humans use their hand palms to realize a friction grasp. Grasp stability is ensured by pressing the object. Technical applications where cooperative manipulators should be able to do a friction grasp are numerous, e.g. when no appropriate grasping device is available (or shall be avoided) or the object structure is not graspable.

Research of friction grasps is mainly done with robotic hands, the fingertips being the contact points. Depending on the nature of the tip a certain number of contacts is required to achieve a firm grasp: for non-deformable and small but frictional contacts, three are required to hold an object with 6 *DoF* tightly. This contact model is called *Hard Finger* or *Point Contact with Friction (PCwF)*. It allows to transmit forces along the contact normal in pushing direction and for forces in tangential plane. If the contact is deformable or of considerable size, two are sufficient contact points form a firm grasp. This is called *Soft Finger* model. In addition to the *Hard Finger* torques around the contact normal are transferred.

The problem of grasping divides into three sub-problems: selection of suitable grasp points, coordinated motion of the constrained system and control of the grasping force especially when interacting with the environment. The problem of coordinated motion is very similar to the case of rigid-fixture, while the control is exclusive to friction grasps. With robot hands it is most important to maintain a stable grasp. Undesired or unnecessary and possibly harmful internal forces occur during dynamic manipulation and interaction with the environment. Since discrete robots in cooperative manipulation are often powerful, proper control of the internal forces becomes more important.

For the reasons stated in the previous subsection hybrid position/force control approaches are less suitable than impedance control and will not be considered. The control architectures for rigid-fixture can be re-used if a internal force is applied in addition. By definition internal forces do not contribute to the motion of the constrained system and thus do not interfere with the control goals of rigid-fixture. Formally this can be described as $Gh_{int} = 0$, i.e. the internal forces h_{int} lie in the null-space of the grasp matrix G . To calculate desired internal forces a suitable vector z is multiplied with a basis of the null-space of the grasp matrix V : $h_{int}^d = Vz$. When it comes to position uncertainties of contact points or deformable objects, calculating internal forces from a fixed grasp matrix may lead to undesired results. One alternative are *Artificial Potential Fields (AFP)* to realize a desired formation of manipulators relative to each other in 3D space. Sieber et al.[SMH15] use *AFP* to build up a formation of robots around an object. The object is grasped by varying the desired distances between the manipulators. The internal forces can thus be set in a physically meaningful way. The generated force of manipulator i in Cartesian

direction k due to the springs connecting to the other manipulators $j \in N_i$ is:

$$f_{i,k}^d = - \sum_{j \in N_i} k_{ij} \frac{p_{i,k} - p_{j,k}}{|p_{i,k} - p_{j,k}|} (|p_{i,k} - p_{j,k}| - d_{ij,k}) \quad (1.12)$$

Notably here $p \in \mathbb{R}^3$ only represents the position of the manipulator, orientation information is omitted. Using the *Hard Finger* contact model a pure translational representation is sufficient since it does not transmit torques. $k_{ij} > 0$ is a control gain. The desired distance is defined by $d_{ij,k}$. The approach controls only the formation of the robots and does not take the object dynamics into account. Positional control for the object is done assuming the object to be in the middle of the manipulators: $p_o = \frac{1}{N} \sum_{i=1}^N x_i$. This has important advantages: no need to know the object's dynamics and no need for tracking the object.

Similar to the concept of *AFP* are 1-*DoF* springs with a desired rest-length. They only allow to set the absolute distance between two points in space, while *AFP* control the distance along each direction. 1-*DoF* can be more intuitive to control, because only one distance parameter has to be changed in contrast to a relative formation in 3D space.

One approach using 1-*DoF* springs is the *Static IPC* by Wimböck et al. [WOH06]. The concept of the *intrinsically passive controller (IPC)* was introduced by Stramigioli [Str01b] and will be covered later on. Here the rest-length springs connect to the middle of the manipulator set-up. The force generated by a single spring connecting from manipulator i to the center is: $k_i \frac{p_i - p_o}{|p_i - p_o|} (|p_i - p_o| - d_i)$. Here $k_i \in \mathbb{R}^3$ is the stiffness matrix and d_i is the rest-length of the i -th spring. Considering the cross-coupling of the springs the generated forces are:

$$f^d = - \sum_{i=1}^N \left[\frac{\partial(p_i - p_o)^T}{\partial p} k_i \frac{p_i - p_o}{|p_i - p_o|} (|p_i - p_o| - d_i) \right] \quad (1.13)$$

Wherein $\frac{\partial(p_i - p_o)^T}{\partial p} \in \mathbb{R}^{3N \times 3}$ is the cross-coupling matrix. Furthermore this approach uses another spring to connect environment and object. This is a 6-*DoF* spring, thus the object's position and orientation is controlled. Force and torque are computed as:

$$f_o^d = K_{o,t} R_o^T (p_o - p_o^d) \quad (1.14a)$$

$$t_o^d = 4 J_{\omega\epsilon}^T K_{o,r} \epsilon_d \quad (1.14b)$$

The translational and rotational stiffness matrices are denoted by $K_{o,t}$, $K_{o,r}$. R_o is a rotation matrix indicating the object orientation, it is defined by the manipulator positions. ϵ_d is the rotational error represented by the vector part of the quaternion, $J_{\omega\epsilon}^T$ denotes the reduced quaternion product.

Damping along each spring is proposed. This is important especially for the cross-coupling springs to suppress oscillatory internal motions.

While f^d acts on the manipulators, f_o^d and t_o^d act on the object. This is resolved by the hand geometry, different hand *Jacobian* matrices cast the wrenches into joint torques. If no kinematic structure is present (e.g. when using discrete robots) often a weighted pseudoinverse of the grasp matrix ([SC92, DPEZ⁺15]) is used. According to Stramigioli[Str01b] the physically meaning of the weighting coefficients is small. Erhart and Hirche[EH15a] recently presented weighted pseudoinverses based on inertia and geometry of object and manipulators. However (in some cases) this requires knowledge of the object properties. Stramigioli therefore introduces a virtual object, the wrench acts directly onto the virtual object and thus influences its dynamics. The manipulators are connected to the virtual object with rest-length springs, so the object wrench is mapped to the manipulators via the virtual object dynamics. Wimböck et al. implement Stramigioli's *IPC* in a simplified version on a four fingered robotic hand[WOH08]. The dynamics of the virtual object are:

$$M_v \ddot{x}_v + C_v \dot{x}_v + g_v = w_v \quad (1.15)$$

The virtual object wrench w_v is composed of environment interaction w_{vo} and manipulator forces f_c : $w_v = w_{vo} + G_v f_c$. Where G_v is the virtual grasp matrix. Environment interaction is described by a 6 *DoF* spring: $w_{vo} = K(x_v^d, x_v)$. The springs connecting to the manipulators exert the wrench on the virtual object: $G_v f_c = G_v K_c(p - G_v^T x_v)$. The control law for the manipulators is already given as:

$$f = -f_c = K_c(G_v^T x_v - p) \quad (1.16)$$

The stiffness matrix $K_c = \text{blockdiag}(k_1 I_3, \dots, k_n I_3)$ is isotropic. The rest-lengths of the springs are chosen by the size of the virtual object, i.e. by specifying the virtual grasp matrix G_v .

One assumption in [WOH08] is that internal forces are sufficient to ensure a firm grasp. This is done in a physically meaningful way by choosing the rest-lengths.

1.2.3 Determination of desired internal forces

Non-rigid grasps object and manipulators are connected with friction constraints. Contact force towards the object generates frictional force opposed to external forces tangential to the contact point. If the tangential forces exceed the frictional force the friction constraint is violated and contact slippage occurs. Squeezing or internal forces do not contribute to the motion of the constrained system. They need to be chosen sufficiently high to maintain a stable grasp under external forces acting on the object. On the other hand they must be bounded to not damage the object. These opposed requirements can be formulated as a cost function.

Buss et al. [BHM96] propose a cost function consisting of two factors: one is a barrier function tending to infinity for violating the friction constraint limit, the other is linear function rising with the exerted force, making high forces costly. The optimal desired internal force is the minimum of the cost function, thus the forces

are chosen as small as possible, preserving a stale grasp.

Non-linear friction constraints are given for the *point contact friction* and the *soft contact friction* model:

The *point contact friction* constraint allows transmitting force in tangential and normal (only towards the object) direction. In addition the *soft contact* constraint transmits the torsional moment around the normal axes. Buss' main contribution is that friction constraints can be reformulated equivalent to positive definite matrices. These matrices are used to construct the cost function. This results in a convex optimization problem, thus an optimum is always a global optimum.

Bicchi et al. [BPA98] limit their contribution to the *point contact friction* model but expand the cost function. A maximum contact force is introduced, this is useful for rather loosing contact than destroying the object. Furthermore a minimum force in normal direction is specified to ensure grasp stability even under uncertainties. This two add up to with the friction constraint to a function, which has a negative scalar value if the set of constraints are respected. The sum of this functions for all contact points forms the cost function.

Han et al. [HTL00] reformulated Buss' positive definite matrix approach as a *linear matrix inequality*(LMI) problem. Being a common optimization task, efficient solvers (e.g. interior point algorithm) exist. Calculation time could be significantly reduced.

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:

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

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

Where Hamiltonian $H(q, p)$ is the total energy of the system, $q = (q_1, \dots, q_k)^T$ denotes the generalized coordinates of the system with k -degrees of freedom and $p = (p_1, \dots, p_k)^T$ 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:

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

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} , $x \in \mathcal{X}$:

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

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

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 $R(x)$.

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

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

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

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

Consider the example of a simple one-dimensional spring-mass-damper system described by $M\ddot{x} = -kx - D\dot{x} + F$. 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:

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

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 $p = M\dot{x}$:

$$\underbrace{\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & -D \end{bmatrix}}_{J(x)-R(x)} \underbrace{\begin{bmatrix} kx \\ \dot{x} \end{bmatrix}}_{\frac{\partial H}{\partial x}(x)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{g(x)} \underbrace{F}_f \quad (2.10)$$

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 be 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 $\langle | \rangle$:

$$P = \langle e | f \rangle, \quad e \in V^*, \quad f \in V \quad (2.11)$$

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

$$\mathcal{D} \subset V \times V^* \quad (2.12)$$

such that

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

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

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

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:

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

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:

$$R(f_R, e_R) = 0 \quad (2.14)$$

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:

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

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

$$e_S^T f_S + e_R^T f_R + e_I^T f_I + e_C^T f_C = 0 \quad (2.16)$$

or by using (2.13)

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

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:

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

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

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:

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

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

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

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

,

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

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:

$$SO(3) = \{R \in \mathbb{R}^{3 \times 3} \mid R^{-1} = R^T, \det R = 1\} \quad (2.22)$$

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

$$H_i^j := \begin{pmatrix} R_i^j & p_i^j \\ 0_{1 \times 3} & 1 \end{pmatrix}$$

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

$$\begin{pmatrix} p^j \\ 1 \end{pmatrix} = H_i^j \begin{pmatrix} p^i \\ 1 \end{pmatrix} \quad (2.23)$$

. The inverse transformation H_j^i is given by

$$H_i^j = (H_j^i)^{-1} = \begin{pmatrix} (R_i^j)^T & -(R_i^j)^T p_i^j \\ 0_{1 \times 3} & 1 \end{pmatrix}$$

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

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

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

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

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

$$\dot{H}_i^j = \tilde{T}_i^{j,j} H_i^j \quad (2.26)$$

The left translation leads to

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

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 $T \in \mathbb{R}^6$

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

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

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

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

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

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 .

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

Again $\tilde{W} \in \mathbb{R}^{4 \times 4}$ 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:

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

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 is the wrench applied to a mass attached to Ψ_i expressed in the coordinate frame Ψ_k .
- $W_i^{k,j}$ 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 .

Note that the first representation is mainly used for bodies, the wrench is exerted *on* the body. The second is useful for springs, the wrench is exerted on the spring *by* the body: $W_i^k = -W_i^{k,j}$. Also it is clear from Newton's third law (action - reaction) that $W_i^{k,j} = -W_j^{k,i}$, thus $W_i^k = W_j^{k,i}$.

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]

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

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

$$\dot{H}_i^j = T_i^j H_i^j \quad (2.30)$$

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

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

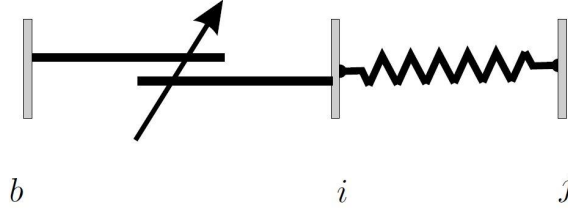


Figure 2.1: Variable rest-length spring [SMA99]

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].

$$T_i^j = T_b^j + Ad_{H_b^j} T_i^b \quad (2.32)$$

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

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

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

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:

$$M_b \dot{T}_b^{b,0} = C_b T_b^{b,0} + W_b^b \quad (2.35)$$

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

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

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

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)

$$\dot{H}_b^0 = H_b^0 T_b^{b,0} \quad (2.38)$$

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

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

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

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

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

$$W_b^b = RT_b^{b,0} \quad (2.42)$$

Or a damper in parallel with spring

$$W_i^{j,j} = RT_i^j \quad (2.43)$$

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 not 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. The simulated manipulator forces are then applied to the real object, in order to transfer the virtual compliant 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 manipulator exert wrenches on the object, in terms of port-Hamiltonian representation they are connected through the control port.

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:

$$\dot{H}_v^b = T_v^b H_v^b \quad (2.44)$$

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

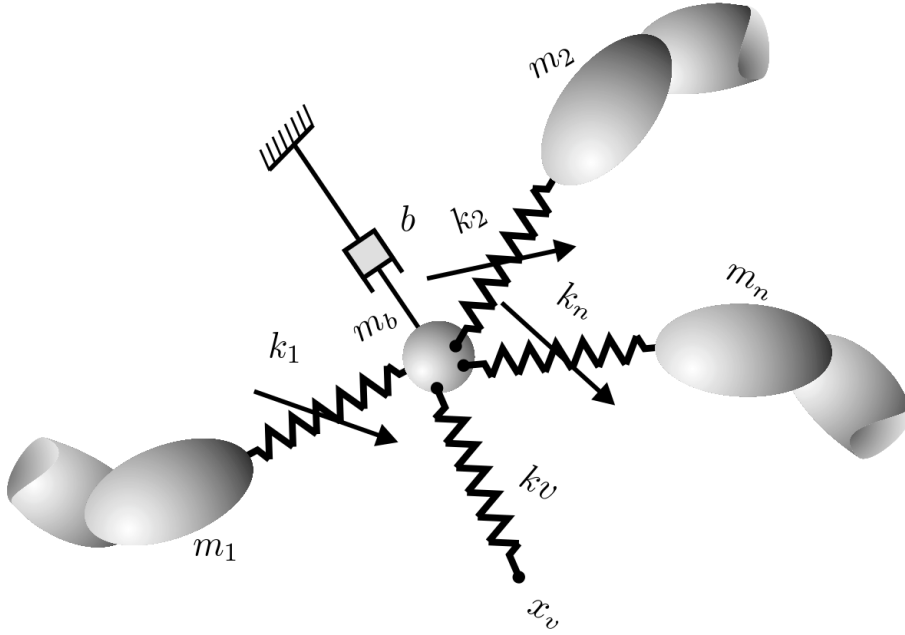


Figure 2.2: Structure of the IPC [SMA99]

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 they 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 $\Psi_v(i)$, $i = 1 \dots N$ 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. The spring forces act on the virtual object in the virtual contact points $\Psi_{v(i)}$. Therefore we need to map the spring forces to the object. This is done by the grasp map, a well known concept in cooperative manipulation (see e.g. [CM08]). It relates both wrenches and motion between manipulators and object. For example the relation of the i -th spring to the object (angular) velocity $(\omega_b^T, v_b^T)^T$ is

$$\begin{pmatrix} \omega_{v(i)} \\ v_{v(i)} \end{pmatrix} = \begin{pmatrix} \omega_b \\ v_b + p_{v(i)}^b \times \omega_b \end{pmatrix}$$

We can combine this for all n -manipulators and obtain the grasp matrix

$$G = \begin{pmatrix} S(p_{v(1)}^b) & I_3 & \cdots & S(p_{v(N)}^b) & I_3 \\ I_3 & 0_3 & \cdots & I_3 & 0_3 \end{pmatrix}$$

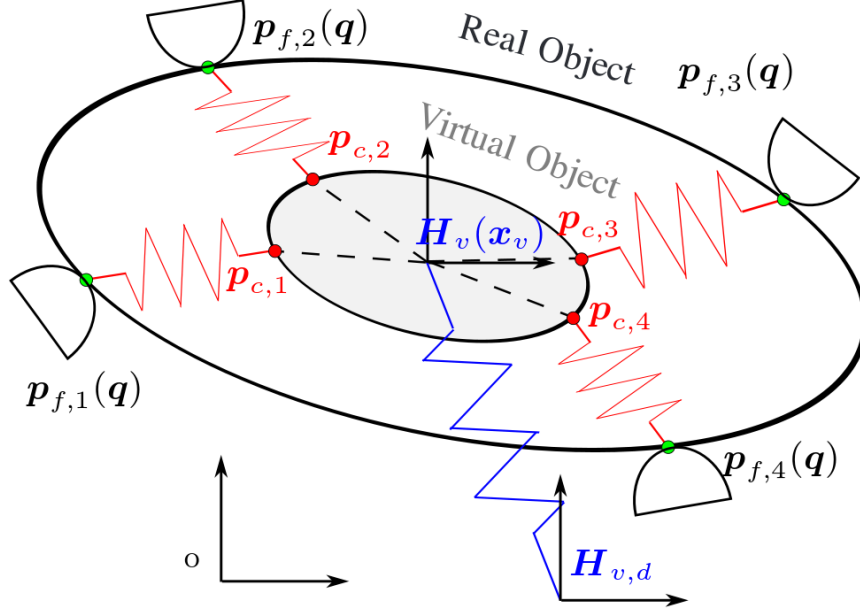


Figure 2.3: Virtual object and springs [WOH08]

Wherein $S(\cdot)$ denotes the skew symmetric operation, defined in (2.28). The grasp matrix is a useful tool relating both twists and wrenches between object and manipulators:

$$W_b = GW, \quad T = G^T T_b \quad (2.46)$$

Where $W = (W_{v(1)}^T, \dots, W_{v(N)}^T)^T$ and $T = (T_{v(1)}^T, \dots, T_{v(N)}^T)^T$ are the stacked vectors of wrenches and twists respectively.

Since we have defined geometric structure of the springs, we can give the PHS representation (see also (2.47))

$$\dot{H}_{v(i)}^i = \left(\begin{pmatrix} 1 & Ad_{H_b^i} \end{pmatrix} \begin{pmatrix} T_b^i \\ T_{v(i)}^i \end{pmatrix} \right) H_{v(i)}^i \quad (2.47)$$

$$\begin{pmatrix} W_b^{i,i} \\ W_{v(i)}^{b,b} \end{pmatrix} = \left(\begin{pmatrix} 1 \\ Ad_{H_b^i}^T \end{pmatrix} \frac{\partial V_{v(i),i}}{\partial H_{v(i)}^i} \right) (H_{v(i)}^i)^T \quad (2.48)$$

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

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

and the wrenches which obey the principle of action and reaction. From now on we will express wrenches acting on bodies no more on springs, since this is more convenient when various springs act on a single body. Recall from subsection 2.2.1 that $W_v^b = -W_v^{b,b}$.

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

We can write the resulting PHS

$$\dot{H}_b^v = (Ad_{H_0^v} - Ad_{H_0^v}) \begin{pmatrix} T_b^0 \\ T_v^0 \end{pmatrix} H_b^v \quad (2.49)$$

$$\begin{pmatrix} W_b^0 \\ W_v^0 \end{pmatrix} = \begin{pmatrix} -Ad_{H_0^v}^T \\ Ad_{H_0^v}^T \end{pmatrix} \frac{\partial V_{v,b}}{\partial H_b^v} (H_b^v)^T \quad (2.50)$$

This is done with the manipulator springs 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 due to rest-length changes are not brought to the inertial frame, since they represent a distinct power port. As a result we can combine springs in a PHS

$$\underbrace{\begin{pmatrix} \dot{H}_{v(1)}^1 \\ \vdots \\ \dot{H}_{v(N)}^N \\ \dot{H}_b^v \end{pmatrix}}_{\dot{x}_s} = (\phi_b \quad \phi_v \quad \phi_i \quad \phi_{rl}) \begin{pmatrix} T_b^0 \\ T_v^0 \\ T_1^0 \\ \vdots \\ T_N^0 \\ T_{v(1)}^b \\ \vdots \\ T_{v(N)}^b \end{pmatrix} \quad (2.51)$$

$$\begin{pmatrix} W_b^0 \\ W_v^0 \\ W_1^0 \\ \vdots \\ W_N^0 \\ W_{v(1)}^b \\ \vdots \\ W_{v(N)}^b \end{pmatrix} = \begin{pmatrix} \phi_b^T \\ \phi_v^T \\ \phi_i^T \\ \phi_{rl}^T \end{pmatrix} \frac{\partial V_s}{\partial x_s} \quad (2.52)$$

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 with the matrices $\phi_b, \phi_v, \phi_i, \phi_{rl}$. The relation of the virtual object twist to the relative motion of the springs is

$$\psi_b \Rightarrow ($$

Interconnection of springs and virtual object

...

This ensures compliant behaviour between object and environment and manipulators and environment. Furthermore object movement is damped with respect to the inertial frame. Dynamic behaviour of the virtual object is simulated in real-time, so no tracking of the object is necessary. The only values that have to be measured are the end-effector positions, the object speed required for the damping term is computed in the simulation. Dampers parallel to the manipulator-object springs are applicable if manipulator speed is measurable. By varying the rest-length of the manipulator springs a the grasp can be opened or closed. Furthermore in a closed state grasping forces can be specified via the rest-length.

The higher control part has the role of planning and scheduling. It controls the state of energy of the IPC and robot. Trajectory and formation of the manipulators are determined, e.g. moving the robots independently to open or close the grasp, directing the formation of robots to move the object. The grasping process can be controlled by changing the virtual object sphere size. Translation or rotation of the grasped object can be commanded by setting the desired virtual object pose. A human operator taking tasks of the Supervisor causes a delay which can be modelled as transmission line with delay between Supervisor and IPC. The control architecture is still valid in a tele-operation setting. Using scattering the passivity of the IPC can be preserved.

2.4 Performance Comparison of control strategies

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

- Trajectory tracking
- Dynamic behaviour
- Internal Forces

2.4.1 Internal force impedance control with feed-forward of the object dynamics [DPEZ⁺15]

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

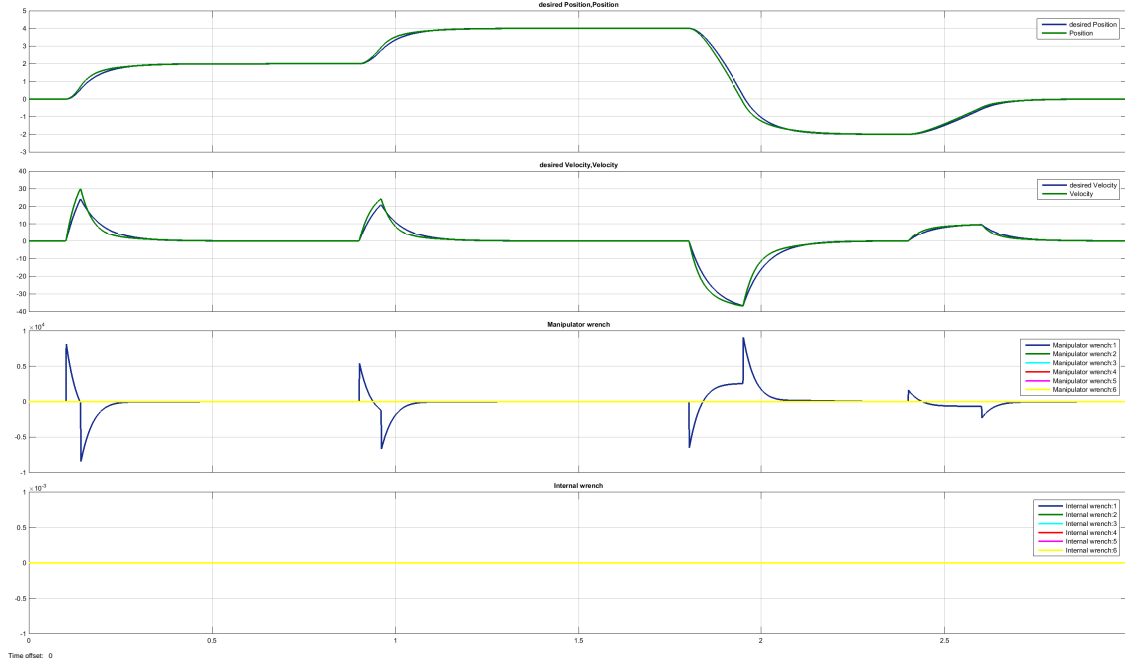


Figure 2.4: Internal force impedance control with feed-forward of the object dynamics: Translation; Graphs from top: Position (desired/actual), Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

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 of 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 [CV01]

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 subsection 1.2.1 this wrench is then decomposed by kineostatic filters in internal and external components. This does not cancel out undesired internal stress but magnifies it: when the contact wrench is not fed back

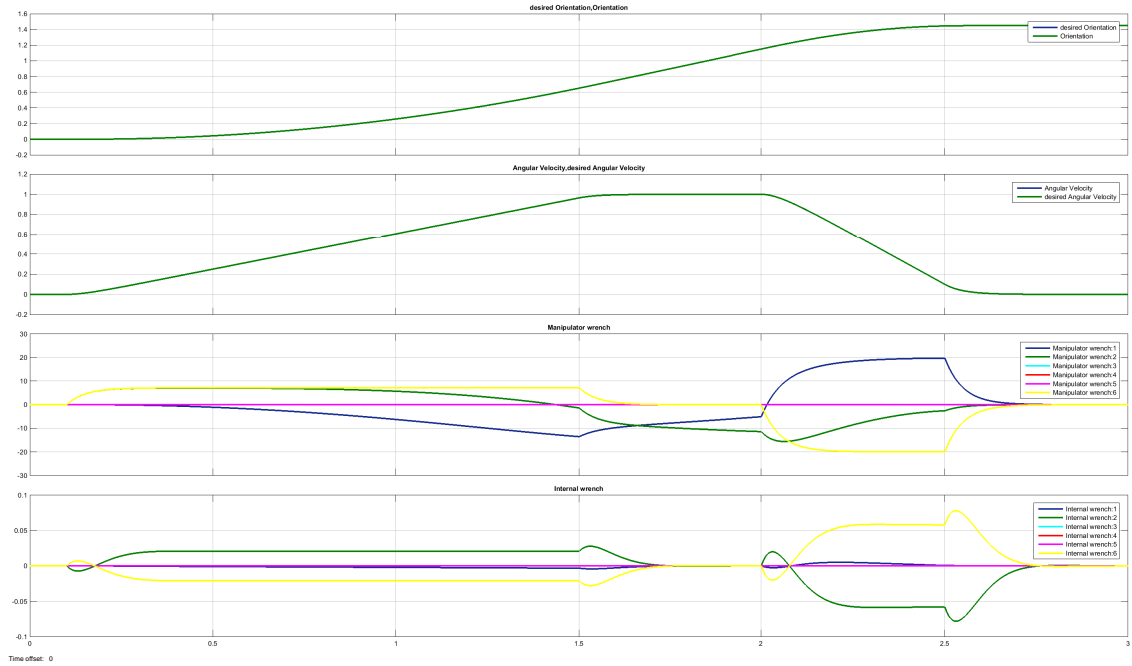


Figure 2.5: Internal force impedance control with feed-forward of the object dynamics: Rotation; Graphs from top: Orientation (desired/actual), Angular Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

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. Behaviour 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.

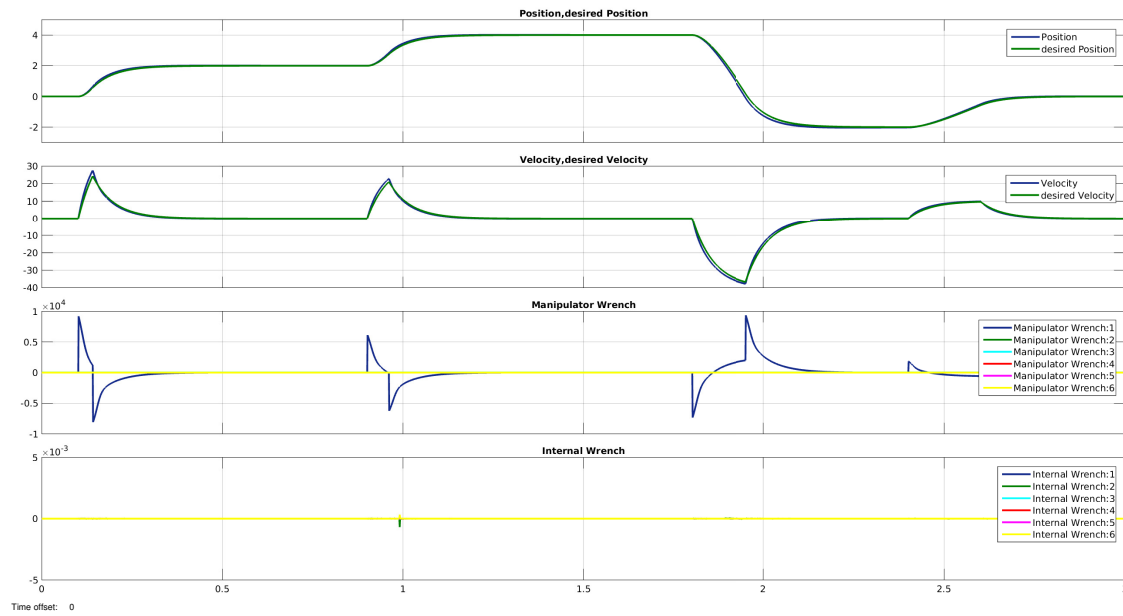


Figure 2.6: Internal and external impedance based reference trajectory generation: Translation; Graphs from top: Position (desired/actual), Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

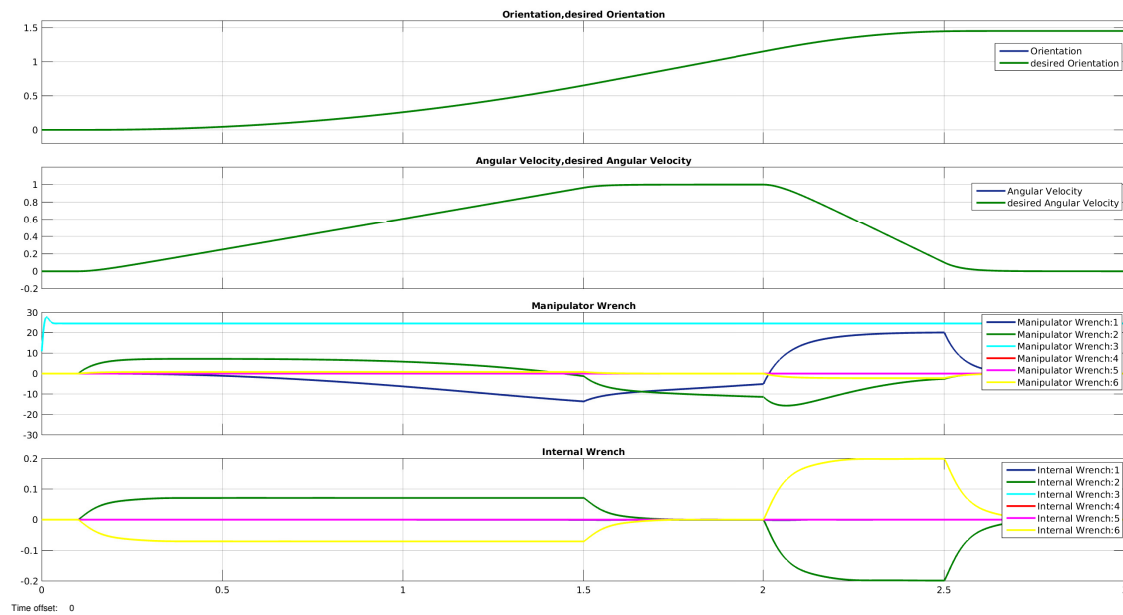


Figure 2.7: Internal and external impedance based reference trajectory generation: Rotation; Graphs from top: Orientation (desired/actual), Angular Velocity (desired/actual), Force exerted by one manipulator, Internal wrench

Chapter 3

Conclusion

List of Figures

1.1	Demonstration of MHI MEISTeR at Fukushima Daiichi NPS	6
2.1	Variable rest-length spring	25
2.2	Structure of the IPC	28
2.3	Virtual object and springs	29
2.4	Internal force impedance control with feed-forward of the object dynamics: Translation	32
2.5	Internal force impedance control with feed-forward of the object dynamics: Rotation	33
2.6	Internal and external impedance based reference trajectory generation: Translation	34
2.7	Internal and external impedance based reference trajectory generation: Rotation	34

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.
- [BPA98] A. Bicchi, D. Prattichizzo, and G. Antognetti. Optimization of internal forces in force-closure grasp. *Proc. of the the 6th IEEE Mediterranean Conf. on Control and Systems*, 16(6), 1998.
- [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.
- [MT93] M.J. Massimino and Sheridan T.B. Sensory substitution for force feedback in teleoperation. *Presence, MIT Press Journals*, 2(4):344–352, 1993.
- [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. Sheridan. *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.

- [Str01a] Stefano Stramigioli. Geometric modeling of mechanical systems for interactive control. In Alfonso Banos, Francoise Lamnabhi-Lagarigue, 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.
- [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

This work is licensed under the Creative Commons Attribution 3.0 Germany License. To view a copy of this license, visit <http://creativecommons.org> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.