



**Joint impedance
control of a
stiffness adjustable
tendon driven
anthropomorphic
hand**

Diplomarbeit

Jens Reinecke



DIPLOMARBEIT

**JOINT IMPEDANCE CONTROL OF A
STIFFNESS ADJUSTABLE
TENDON DRIVEN ANTHROPOMORPHIC
HAND**

Freigabe:

Der Bearbeiter:

Unterschriften

Jens Reinecke



Betreuer:

Dipl.-Ing. Maxime Chalon

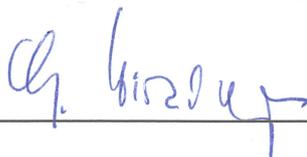


Dipl.-Ing. Thomas Wimböck

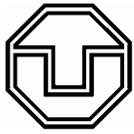


Der Institutsdirektor

Prof. Dr. G. Hirzinger



Dieser Bericht enthält 89 Seiten, 40 Abbildungen und eine Tabelle.



DIPLOMARBEIT

zum Thema

Joint impedance control of a stiffness adjustable tendon driven
anthropomorphic hand

Gelenkimpedanzregelung für eine seilgetriebene anthropomorphe Hand mit
einstellbarer Steifigkeit

vorgelegt von Jens Reinecke
im Studiengang Mechatronik, Jg. 2006
geboren am 13.07.1987 in Karl-Marx-Stadt

zur Erlangung des akademischen Grades

Diplomingenieur
(Dipl.-Ing.)

Betreuer: Dipl.-Ing. Maxime Chalon (DLR)
Dipl.-Ing. Thomas Wimböck (DLR)
Verantwortlicher Hochschullehrer: Prof. Dr. techn. Klaus Janschek
Tag der Einreichung: 27.09.2011



Aufgabenstellung für die Diplomarbeit

für

Herrn Jens Reinecke

Joint impedance control of a stiffness adjustable tendon driven anthropomorphic hand

Objectives:

The new Hand-Arm-System of the DLR provides an anthropomorphic robot hand that is designed to be as robust, strong and sized as the human hand. To realize the size and an human like strength, the joints are driven antagonistically by tendons that allows the actuators to be placed in the forearm. The robustness is achieved by passive nonlinear spring elements in the path of each tendon. The advantage of such elements is that they can store the energy of impacts in the spring and act as a mechanical low pass filter. Consequently, actuators have more time to react and prevent damages. To further improve the robustness no sensors are located in the hand. Moreover, the antagonistic configuration provides the possibility to adjust pretension of tendons and the nonlinear spring elements realize an adjustable stiffness of each joint.

Based on previous DLR work on impedance control and control of flexible robots the main topic of this thesis is the derivation and implementation of controllers for flexible joint robots with varying stiffness. In general adjustable stiffness can be used to achieve suitable characteristics for each kind of task. One fundamental question is how to distribute the effective robot joint stiffness between the mechanics and the controller. The nonlinear kinematics (thumb, wrist) as well as the use of non collocated sensors increases the overall complexity.

Finally, an abstraction layer is needed to integrate with high level planning (object level control, grasp planner). While moving around without grasping an object low stiffness is needed to compensate as much as possible impact energy. For grasping tasks or fine manipulation an interface for adjusting the impedance for stiffness adaptation is useful.

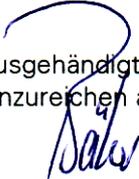
Main tasks of the diploma thesis:

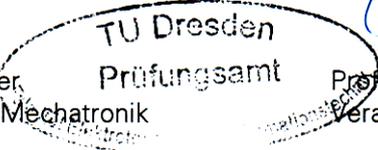
- requirements definition according to TUD-IfA standards
- literature review on modeling and control of robots with variable stiffness
- derivation of a suitable robot hand model
- derivation and implementation of an joint impedance controller
- experimental validation of closed-loop behavior and performance evaluation
- design and implementation of a stiffness distribution algorithm (which should incorporate mechanical constraints)
- support the implementation a synergy controller based on previous work (optional)
- extension to coordinate the fingers and the thumb with the wrist (optional)
- thesis documentation in English

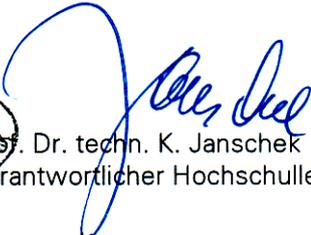
Betreuer: Maxime Chalon, Thomas Wimböck (Institute of Robotics and Mechatronics German Aerospace Center (DLR), Wessling)

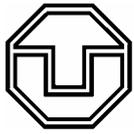
Ausgehändigt am: 28.03.2011

Einzureichen am: 27.09.2011


Prof. Dr.-Ing. Bernard Bäker,
Vors. Prüfungsausschuss Mechatronik

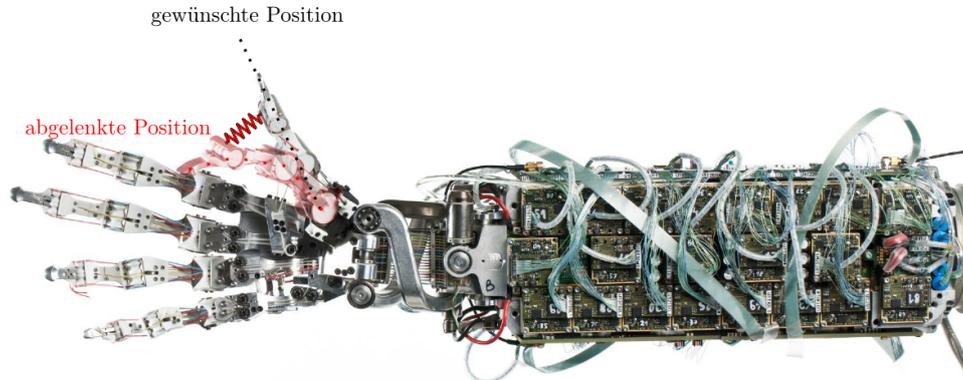

TU Dresden
Prüfungsamt


Prof. Dr. techn. K. Janschek
Verantwortlicher Hochschullehrer



Gelenkimpedanzregelung für eine seilgetriebene anthropomorphe Hand mit einstellbarer Steifigkeit

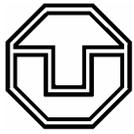
Das DLR Hand-Arm-System ist der erste Teil eines neuen humanoiden Roboters, der so robust und anthropomorph wie möglich ist. Jedes Gelenk des Systems besitzt einen verstellbaren Federmechanismus. Somit kann die mechanische Steifigkeit des Gelenks angepasst werden. Die Federn erhöhen außerdem die mechanische Robustheit des Systems gegenüber Kollisionen. In dieser Arbeit wurde die Regelung der Hand dieses Systems entwickelt. Dazu wurde ein Reglermodul implementiert, mit dem die gewünschte Position und Steifigkeit im System vom Benutzer eingestellt werden kann. Das Reglerverhalten bleibt für jede Benutzereingabe vorhersehbar, da die Gelenkansschläge und Seilkraftgrenzen überwacht werden. Die mechanische Steifigkeit und die Reglersteifigkeit werden so eingestellt, dass die gewünschte effektive Steifigkeit in den Gelenken erzielt wird. Das Systemmodell wurde am realen System verifiziert und das Reglermodul am realen System bezüglich der Benutzeranforderungen validiert.



Betreuer: Dipl.-Ing. Maxime Chalon (DLR)
Dipl.-Ing. Thomas Wimböck (DLR)
Hochschullehrer: Prof. Dr. techn. Klaus Janschek
Tag der Einreichung: 27.09.2011

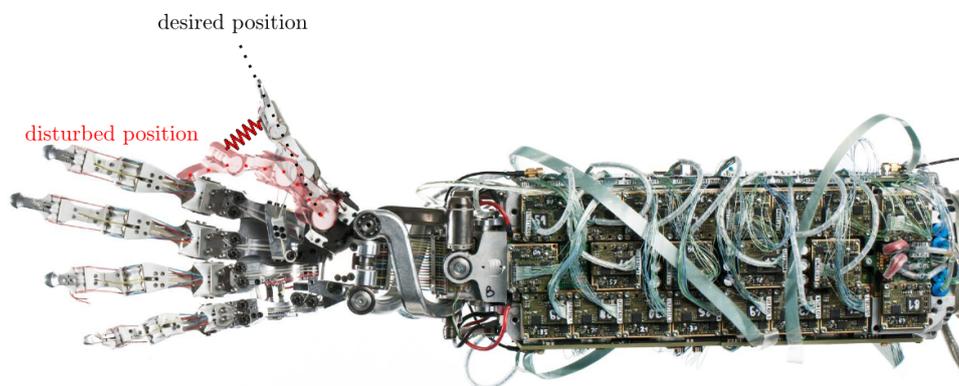
DIPLOMARBEIT

Bearbeiter: Jens Reinecke



Joint impedance control of a stiffness adjustable tendon driven anthropomorphic hand

The DLR Hand-Arm-System is the first component of a new humanoid robot which is as robust and anthropomorphic as possible. Each joint of the system has an adjustable spring mechanism, which enables to adjust the mechanical stiffness. Additionally, the springs increase the mechanical robustness of the system against impacts. In this thesis the compliance control of the hand of the Hand-Arm-System is developed. A control unit is implemented, where the user can set the desired joint positions as well as the desired joint stiffness. The controller remains predictable for any input, because a state machine monitors the system. The mechanical as well as the controller stiffnesses are set in the system such that the desired effective stiffness in the joints is achieved. The system model is verified with experiments on the real system and the control unit is validated regarding the user specification.



Tutor: Dipl.-Ing. Maxime Chalon (DLR)
Dipl.-Ing. Thomas Wimböck (DLR)
Supervisor: Prof. Dr. techn. Klaus Janschek
Day of Submission: 27.09.2011

DIPLOMA THESIS

Author: Jens Reinecke

Contents

1	Introduction	1
1.1	Hand of the DLR Hand-Arm-System	2
1.2	Compliance control	3
2	Related work	5
2.1	Coupling matrix for tendon driven systems	5
2.2	Mechanical stiffness control of tendon driven manipulators	7
2.3	Flexible Joints	8
2.4	Variable stiffness	8
2.5	Observer	9
3	Requirements definition	11
3.1	User specifications	11
3.2	Structural analysis	11
3.2.1	Context analysis	11
3.2.2	Data flow analysis	13
3.3	Functional constraints	18
4	Modeling the anthropomorphic hand	21
4.1	Anatomical terms	21
4.1.1	Hand bones	21
4.1.2	Hand joints	21
4.1.3	Finger motions	22
4.2	System model of a tendon driven robotic hand	23
4.2.1	Tendon kinematics	23
4.2.2	Derivation of the equations of motion	25
4.3	System model validation	26
4.4	Cascaded system validation	28
5	Design and discussion of compliant controllers	31
5.1	Compliance controller	31
5.1.1	Admittance control	31
5.1.2	Impedance control	34

5.1.3	Compliant controller choice	38
5.2	Control of the joint stiffness	39
5.2.1	Constant impedance gain and constant pretension	39
5.2.2	Controlling constant effective stiffness	40
5.2.3	Stiffness distribution controller choice	42
6	Implementation	45
6.1	Experimental setup	45
6.1.1	Hardware setup	45
6.1.2	Adaption of the control software	46
6.2	Implementing of the control model	47
6.2.1	General control architecture	47
6.2.2	Implementation of the controller elements	50
7	Validation	55
7.1	Step response	55
7.2	Frequency analysis	56
7.3	Stiffness control	57
8	Conclusion	59
	Appendix A Simulink required paths	63
	Appendix B Step response	65
	Bibliography	67

List of Figures

1.1	Hand-Arm-System [DLR]	2
1.2	Hand of the Hand-Arm-System [DLR]	3
1.3	Kuka and light weight robot	4
2.1	Simple antagonistic example	5
2.2	Simple nullspace example	7
2.3	Flexible joint	8
2.4	Controller structure from [Wim+08]	9
3.1	Control module	11
3.2	Context analysis	12
3.3	Data flow analysis	13
3.4	Data flow analysis F1	14
3.5	Data flow analysis F2	15
3.6	Data flow analysis F3	16
3.7	Data flow analysis F4	17
3.8	Data flow analysis F5	18
4.1	Hand Bones [Bun09]	22
4.2	Joints and finger motions	23
4.3	PIP and DIP joint coupling	24
4.4	Base joint coupling	24
4.5	Applied method of [Jan09]	27
4.6	Comparison of simulation and real system	28
4.7	Comparison of inner and outer system	29
5.1	Discussion overview	32
5.2	Measurements of the system	32
5.3	Admittance control	33
5.4	Impedance controller on motor side	35
5.5	Spring element tendon elongation	37
5.6	Impedance controller with extended joint angle feedback	37
5.7	Impedance controller with observer	38
5.8	Stiffnesses serial connection overview	39

5.9	Impedance controller with pretension	40
5.10	Impedance controller with effective stiffness adjustment . . .	41
6.1	Hardware overview	45
6.2	Connection overview	46
6.3	Spring element curve	48
6.4	Simulink model	53
7.1	Error due to step amplitude and effective stiffness	56
7.2	Comparison of the open and closed system	57
7.3	Stiffness validation	58
B.1	Step response measurements	65

List of Tables

A.1 Required paths 63

List of Abbreviations and Symbols

Abbreviations

DLR	Deutsches Zentrum für Luft und Raumfahrt
DOF	Degree of freedom
GUI	Graphical User Interface
HAL	Hardware Abstraction Layer (Driver for the HASy)
HASy	Hand-Arm-System
MF	Main function
QNX	Realtime Unix-like system
SVD	Singular Value Decomposition
TCP	Transmission Control Protocol

Symbols

m	Number of actuators (Hand-Arm-System hand: 38) (\mathbb{R})
n	Number of joints (Hand-Arm-System hand: 19) (\mathbb{R})
$r_{1..8}$	Joint radius (\mathbb{R})
r_m	Motor radius (\mathbb{R})
$\mathcal{N}(\mathbf{P})$	Nullspace basis of the coupling matrix ($\mathbb{R}^{m \times (m-n)}$)
ω	Sinus signal angular frequency (\mathbb{R})
θ_0	Amplitude for frequency analysis input signal ($\mathbb{R}^{m \times m}$)
$\boldsymbol{\tau}_m$	Motor torque (\mathbb{R}^m)
$\boldsymbol{\tau}_q$	Torque on the joints (\mathbb{R}^n)
$\boldsymbol{\tau}_{\theta,fric}$	Friction in the motor and tendon guiding (\mathbb{R}^m)
$\boldsymbol{\tau}_{ext}$	External torque e.g. disturbances (\mathbb{R}^n)
$\boldsymbol{\tau}_{q,fric}$	Friction in the joints (\mathbb{R}^n)
$\boldsymbol{\theta}$	Motor position (\mathbb{R}^m)
$\boldsymbol{\theta}_{des}$	Desired motor position (\mathbb{R}^m)
$\boldsymbol{\theta}_{lever}$	Spring element position (\mathbb{R}^m)
\mathbf{B}	Motor inertia matrix ($\mathbb{R}^{m \times m}$)
$\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$	Finger centrifugal and Coriolis terms (\mathbb{R}^n)
\mathbf{D}_p	Damping of the motor position control ($\mathbb{R}^{m \times m}$)
\mathbf{D}_{imp}	Impedance controller D gain matrix ($\mathbb{R}^{m \times m}$)
\mathbf{E}	Motor side coupling matrix ($\mathbb{R}^{m \times m}$)
\mathbf{f}_t	Tendon force (\mathbb{R}^m)

\mathbf{f}_{int}	Internal forces (\mathbb{R}^n)
\mathbf{f}_{pre}	Forces to set the mechanical stiffness manually (\mathbb{R}^n)
$\mathbf{f}_{t,des}$	Desired tendon force (\mathbb{R}^m)
$\mathbf{g}(\mathbf{q})$	Link-side gravity vector (\mathbb{R}^n)
\mathbf{h}	Tendon elongation (\mathbb{R}^m)
$\mathbf{h}_q(\mathbf{q})$	Tendon elongation due to the joint motion (\mathbb{R}^m)
\mathbf{h}_{lever}	Tendon elongation due to the spring element motion (\mathbb{R}^m)
\mathbf{I}	Entity matrix ($\mathbb{R}^{m \times m}$)
\mathbf{K}_p	Stiffness of the motor position control ($\mathbb{R}^{m \times m}$)
\mathbf{K}_{ad}^{-1}	Admittance gain ($\mathbb{R}^{n \times n}$)
\mathbf{K}_{eff}	Effective stiffness ($\mathbb{R}^{m \times n}$)
\mathbf{K}_{force}	Stiffness of the force control loop ($\mathbb{R}^{m \times m}$)
\mathbf{K}_{imp}	Impedance controller stiffness matrix ($\mathbb{R}^{m \times m}$)
\mathbf{K}_m	Mechanical stiffness ($\mathbb{R}^{n \times n}$)
$\mathbf{M}(\mathbf{q})$	Link inertia matrix ($\mathbb{R}^{n \times n}$)
$\mathbf{P}(\mathbf{q})$	Coupling matrix between tendon and joint velocity ($\mathbb{R}^{n \times m}$)
\mathbf{P}^+	Pseudo inverse of the coupling matrix ($\mathbb{R}^{m \times n}$)
\mathbf{q}	Joint position (\mathbb{R}^n)
\mathbf{q}_{des}	Desired joint position (\mathbb{R}^n)
\mathbf{q}_{obs}	Joint position computed from the observer (\mathbb{R}^n)
\mathbf{q}_{real}	Joint position measured directly on the joint (\mathbb{R}^n)
C	Cost function of the gradient search (\mathbb{R})

1 Introduction

This diploma thesis was conducted at the Institute of Robotics and Mechatronics at the DLR (Deutsches Zentrum für Luft und Raumfahrt). The DLR is the national research center of space and aerospace of Germany. Goals of the institute are based on the idea of relieving man of tedious or dangerous tasks. A good example of this aim is a complete new prototype: the Hand-Arm-System (figure 1.1). This system is the crucial component of a new humanoid robot that is anthropomorphic and robust enough to perform in a wide range of human tasks. Its main characteristics are that each joint has an adjustable stiffness which can be low, when the robot expects impacts and set to stiff if high precision or bandwidth in positioning is needed. The variable stiffness also enables to store energy in the drive system. This can be used to reach higher performance in dynamic motions like throwing. If the potential energy stored in the spring is transformed into kinetic energy at the right time, the link is able reach a higher velocity than the actuators would achieve. In this thesis the control design of the hand of this system is the main focus. A control unit shall be developed which controls the compliant behavior of the joints specified by a user given position and stiffness. For any user input the model has to have predictable behavior as well as to remain stable.

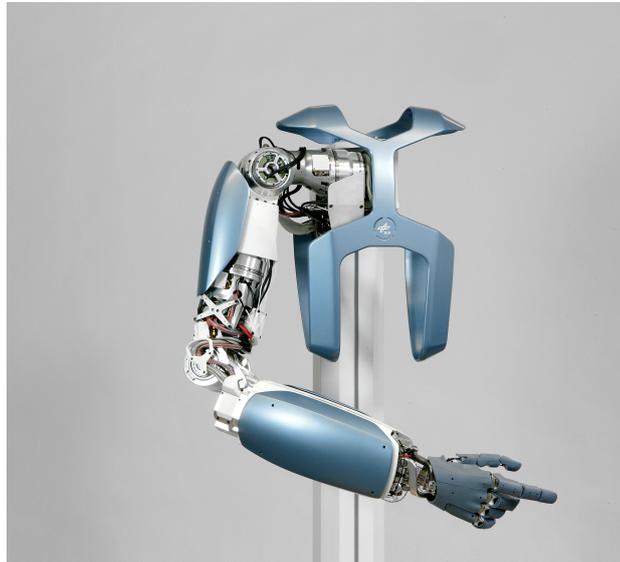


Figure 1.1: Hand-Arm-System [DLR]

1.1 Hand of the DLR Hand-Arm-System

Since 1998, the institute develops anthropomorphic hands such as DLR Hand I,II, which are now used in many applications (e.g. preparing ice tea or catching thrown balls). However, these hands are larger than the human hand and they can be damaged by impacts. The new generation of hands (figure 1.2) has three major characteristics: anthropomorphism, robustness¹ and energy storage. To obtain anthropomorphic size and strength, the actuators are placed in the forearm. The joints of the fingers are antagonistically connected, each via two tendons to two actuators (figure 4.3). This configuration is similar to the design of a real human hand with muscles for flexion and extension. The new hand has 19 joints which is less than the approximated number of joints of a human (20 joints model used in [CA09]). To actuate the 19 joints, 38 motors are needed, which requires tremendous modular electronic and mechanical integration. The second major attribute is the mechanical robustness, which is achieved by guiding the tendon through a spring mechanism that acts as a mechanical low pass filter. In case of an impact on the fingers the energy is initially stored in the springs and not transmitted to the gear or the motor directly. Additionally, the antagonistic drive system combined with the nonlinear springs allows the user to set the

¹In the context of this thesis the expression robustness refers to the mechanical property.

mechanical stiffness in the joints. This feature can also be used to store potential energy in the spring and release it when needed, e.g. for snapping. The design with nonlinear springs enables these characteristics at the cost of increased control complexity. Hence, the major topic of this diploma thesis is to design and evaluate a controller which provides controlled compliant behavior and is also able to modify the stiffness in the joints.

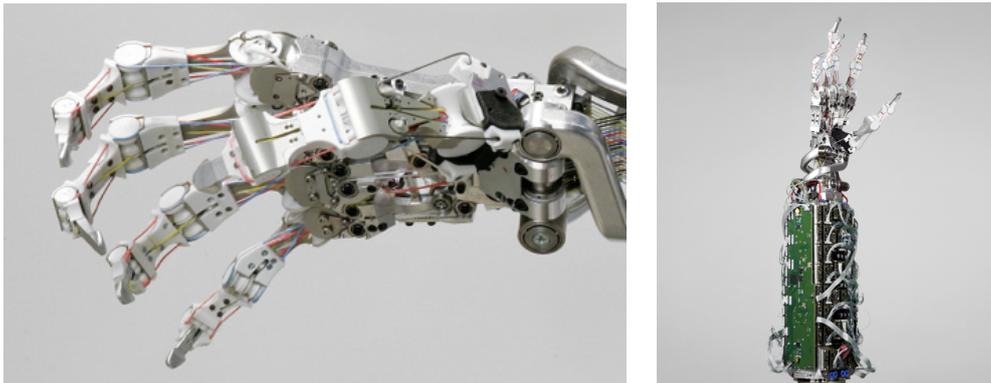


Figure 1.2: Hand of the Hand-Arm-System [DLR]

1.2 Compliance control

In most industrial applications robots have rigid links and minimal mechanical compliance in the joints like the robot in the left of figure 1.3. They are mainly controlled by a stiff position controller, which is not suitable for interaction with humans or tight fit assembly tasks. If force torque sensors are added to the joints a compliant behavior can be achieved. Therefore, new control strategies have been developed in the past. For example an impedance controller, which defines a spring between the desired joint position and the real position obtained by measurements. Hence, the robot joints can be moved away from a desired configuration by applying an external force. The steady state error is therefore defined by the external load and the prescribed controller stiffness. The right picture of figure 1.3 shows a robot utilizing compliance control, which allows to mount parts with tight fit.



Figure 1.3: Left: Kuka 240 [Wim09] Right: Light weight robot controlled compliantly [DLR]

2 Related work

This chapter presents the work related. Authors performed simulations and developed theories about topics like variable stiffness and flexible joint behavior for several decades. But since there are only few nonlinear antagonistic tendon driven systems built, there are no applied controllers for such type of manipulators. The problem is divided into the tendon kinematics, the stiffness control of tendon driven systems and the flexible joint problem. Some related work based on observers is also reported.

2.1 Coupling matrix for tendon driven systems

Joints are connected to the motors via the tendons (e.g. figure 2.1). The length of a tendon is a function of the joint angle. This relation is referred to as the tendon kinematics $\mathbf{h}_q(\mathbf{q})$. The computation of the coupling of such

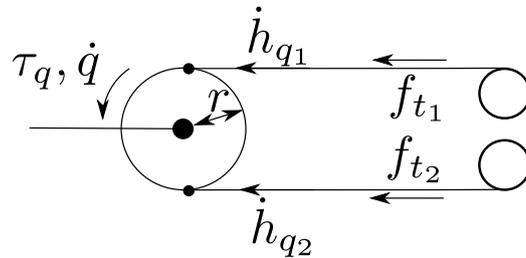


Figure 2.1: Simple example: Antagonistic joint driven by two motors

systems has been done for robotic systems by Kobayashi [KHO98] and Murray [MLS94] in detail. More specific examples are reported in [Tah+06], where a study of a dual-finger model with synergistic actuation is shown or in [BP00], where a framework of tendinous actuation for biomorphical designed robots is given. A more detailed view is given in [Pal07]. As shown in [MLS94] the coupling matrix $\mathbf{P}(\mathbf{q})$ is defined as the derivative of the tendon length \mathbf{h}_q with respect to the joint position \mathbf{q} . From that equation we can also obtain

the relationship between the joint velocity and the tendon velocity:

$$\dot{\mathbf{h}}_q = \frac{d}{dt} \mathbf{h}_q(\mathbf{q}) = \underbrace{\left(\frac{\partial \mathbf{h}_q(\mathbf{q})}{\partial \mathbf{q}} \right)}_{\mathbf{P}^T(\mathbf{q})} \frac{d}{dt} \mathbf{q}. \quad (2.1)$$

The conservation of power in a tendon driven system is formulated such that the tendon power on the left is equal to the joint power on the right of (2.2). The tendon power is the product of the tendon velocity $\dot{\mathbf{h}}_q$ and tendon force \mathbf{f}_t and the joint power is the product of the joint velocity $\dot{\mathbf{q}}$ and torque $\boldsymbol{\tau}_q$:

$$\dot{\mathbf{h}}_q^T \mathbf{f}_t = \dot{\mathbf{q}}^T \boldsymbol{\tau}_q. \quad (2.2)$$

If equation (2.1) is inserted in (2.2), the mapping between torques and tendon forces is obtained:

$$\boldsymbol{\tau}_q = \mathbf{P}(\mathbf{q}) \mathbf{f}_t. \quad (2.3)$$

To get an idea of how the coupling matrix is derived, a small example (figure 2.1) follows, in which the tendon kinematics is linear and therefore not dependent on the joint position \mathbf{q} . The tendon velocity can be computed by using the joint radius and the correct direction. To get the joint torque from the tendon forces the coupling matrix has to be transposed. If the coupling matrix is not dependent on the joint position \mathbf{q} , the equation can be integrated and the coupling is also valid for the joint position \mathbf{q} and tendon elongation \mathbf{h}_q . Then, the integration constant is the tendon length in zero position of the joint. Tendon velocity and joint torque can be achieved by:

$$\begin{pmatrix} \dot{h}_{q1} \\ \dot{h}_{q2} \end{pmatrix} = \begin{pmatrix} r \\ -r \end{pmatrix} \dot{q} \quad (2.4)$$

$$\boldsymbol{\tau}_q = \begin{pmatrix} r & -r \end{pmatrix} \begin{pmatrix} f_{t1} \\ f_{t2} \end{pmatrix}, \quad (2.5)$$

with r the radius of the joint.

2.2 Mechanical stiffness control of tendon driven manipulators

Examples of tendon driven systems that can change the mechanical stiffness are presented in [LK+91], [KB02] and [KOY04]. In most tendon driven systems it is possible to achieve the same joint torque τ_q by different combinations of tendon forces. There are combinations that do not create a torque on the joint, but change of the tendon forces may have an influence on the joint stiffness. A simple example is a simple joint with two antagonistic actuators, which drive the link with tendons, containing a nonlinear spring (figure 2.2). The joint moves, if they move in the opposite direction c). If they move in the same direction b) the joint will stay and the load of the two nonlinear springs is increasing. This gives an additional DOF for setting the stiffness in the system (equation (2.6)) or to maintain pulling constraints like positive tendon forces. The vector \mathbf{f}_t defines the force of each tendon, which can be decomposed into the torque generating forces and the pretension \mathbf{f}_{int} in (2.6). The possibility of changing the stiffness raises the complexity. Finding a correct α while satisfying the constraints is challenging. Suitable approaches can be found in [Abd+10],[PJ+11] and [Wim+08].

$$\mathbf{f}_t = \mathbf{P}^+ \boldsymbol{\tau} + \mathbf{f}_{int} \quad \text{with} \quad \mathbf{f}_{int} = \mathcal{N}(\mathbf{P})\boldsymbol{\alpha} \quad (2.6)$$

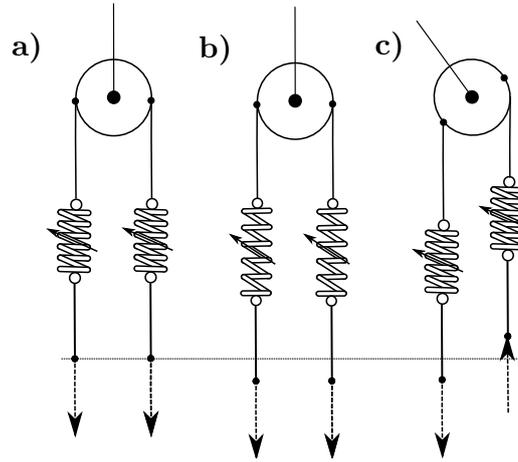


Figure 2.2: Simple nullspace example

2.3 Flexible Joints

In contrast to industrial robots with easily controllable rigid joints, flexible joints controlled by a single motor position (velocity) feedback behave unsatisfactorily in practice. Flexible joints are often modeled by a simple linear spring between the actuator and the link as shown in figure 2.3. One implementation of a simple PD controller with gravity compensation is derived in [Tom91]. Better results using additional link side torque and position sensors are outlined in [ASH00]. Additionally, this approach can be extended to fully compensate the whole robot dynamics. Other approaches deal with complete feedback linearization [DLL98] or partial linearization [LG95], as well as suitable feed-forward terms in [DL00] and a backstepping approach in [OL97]. One way to separate the problem is to use a two loop system in a singular perturbation approach manner (like in [Ott08]), where a fast timescale part of the system model can be separated from the slow timescale part. The main problem of all the approaches is to prove stability, especially when using link side sensors, which might feed additional energy into the system. If a nonlinear spring is used, the theoretical issues are complicated.

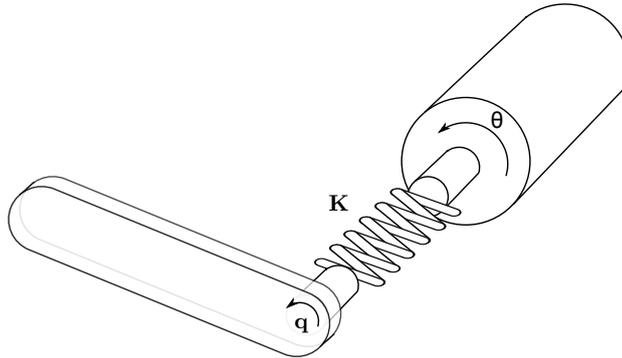


Figure 2.3: Flexible joint with joint position q , mechanical joint stiffness K and motor position θ

2.4 Variable stiffness

Only little work deal with the control of variable stiffness tendon driven systems. One interesting approach is done in [Wim+08] as depicted in figure 2.4. It is about a controller that is examined in simulation on a two DOF example. The desired joint position q_d and stiffness S_d are the user

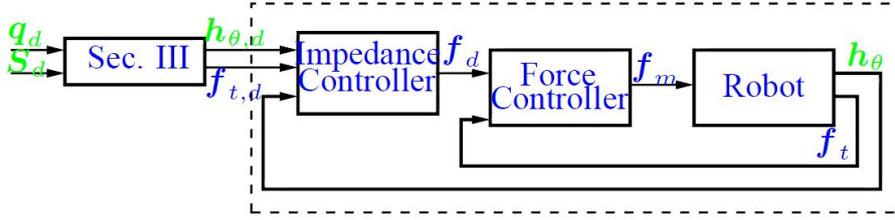


Figure 2.4: Controller structure from [Wim+08]

inputs. These quantities are used to compute the desired tendon forces and elongations. The controller structure is an impedance controller. The complexity arises from the block (Sec. III) where all constraints for the tendons and the joint are taken into account to obtain the desired joint angle and stiffness. This analytic approach provides the basics for such an antagonistic driven system.

2.5 Observer

An observer is needed to obtain state quantities that cannot be directly measured. It is part of the controller structure and provides additional information about the system. Approaches based on observers on related systems can be found in [CPK03] and [Cor+01]. In [AO93] also a set-point controller for flexible joints with an observer is proposed. An exact knowledge of the model required to design an observer that gives a full-state feedback. In systems like the DLR Hand-Arm-System with high dimensionality, complex friction models and transmission chains, the observers are difficult to design.

3 Requirements definition

In this chapter, the diploma thesis objectives will be explained in detail by defining all constraints and issues. First a user specification is given, which gives a convenient overview of the tasks. The structural analysis provides a more explicit view and is divided into two parts: context analysis and data flow analysis. The first one shows the overall process; which terminators are involved and how they are connected. In the latter part, the main process is with a finer granularity to explain the interaction in detail. The data flows are used to define the interfaces of the functions.

3.1 User specifications

The user specification is shown in figure 3.1. The user can define joint position q_{des} and stiffness K_{eff} , as well as simply selecting on and off each finger. The control module is zeroing autonomously and controlling the finger compliantly. It outputs the torques, the current position and error messages. Additionally, the control module should behave predictable with respect to any user input combination. For example saturating desired joint position to the minimum or maximum angle.

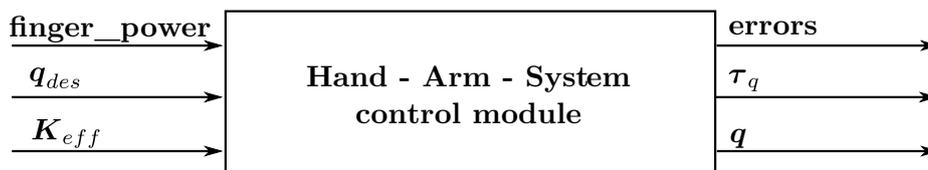


Figure 3.1: Control module

3.2 Structural analysis

3.2.1 Context analysis

In the center of figure 3.2, the main function (MF) is the compliant joint controller. The user enables the controller by control flow D1 and is able to

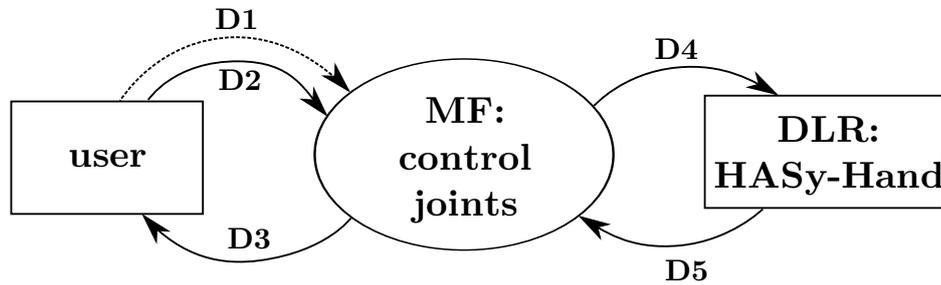


Figure 3.2: Context analysis

adjust the controller by sending the parameters (desired position, stiffness) in data flow D2. The controller sends the desire values for the actuator low level controllers via data flow D4 and is able to switch them on and off. All measurements of the hand of the Hand-Arm-System are transmitted via data flow D5. It contains all needed information for the controller loop. The states of the system are formatted and returned to the user in D3.

Terminators:

- **User:** The user starts the controller and specifies the input of the controller (e.g. desired position, stiffness). It is also possible that the user is replaced by a coordination module, which drives the arm and the hand.
- **Hand-Arm-System:** The hand of the Hand-Arm-System is the hardware which is controlled by the MF. It provides sensor values (motor current, motor position and velocity, spring element angle, temperatures and error flags)

Data flows:

- D1 start command; to start the controller
- D2 desired controller values; joint position and stiffness
- D3 user feedback; measurements and error messages
- D4 desired hardware values; for low level controller (current or position) on the motor boards
- D5 measured hardware values; motor current, motor position and velocity, spring element angle, temperatures and error flags

3.2.2 Data flow analysis

Data flow of the main function

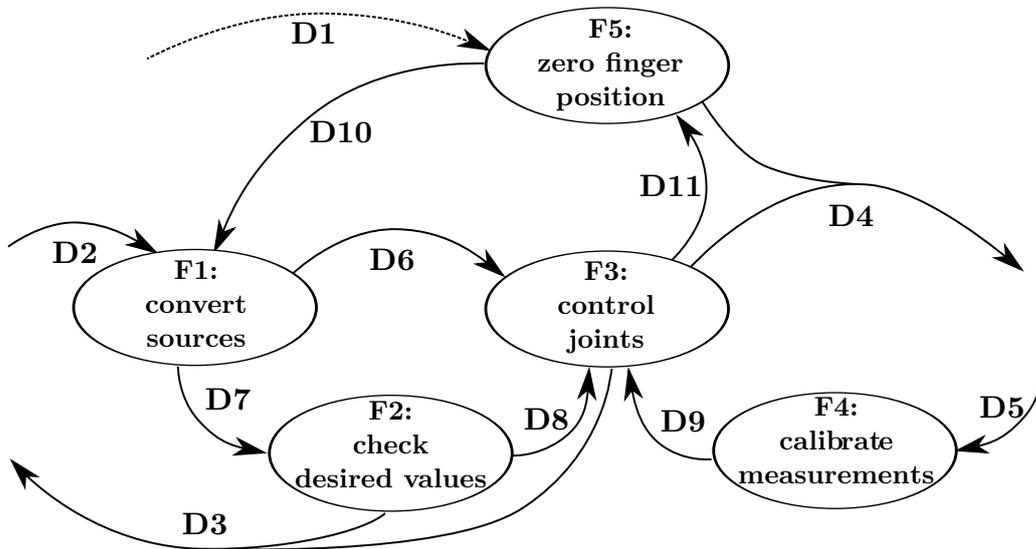


Figure 3.3: Data flow analysis

In figure 3.3 the MF is presented in detail. The function F1 organizes the interface between the controller and the user given data. When the data is provided by external sources (GUIs or other robots) the data is prepared and transformed into the required units. It also contains parameters for the controller. The transformed desired variables are sent to function F2 via data flow D7, which is monitoring the observance of the system borders and sending the correct variables to the controller in data flow D8. The variables that are not set by the user, are sent directly from function F1 to the function F3 via data flow D6. F3 contains the controller and is computing the desired values for the hardware components, which are sent via D4. The measurements in D5 are transformed into the proper units by using function F4. Therefore, D9 provides all measurements of the Hand-Arm-System in the correct units for the controller. D9 is especially important for the controller feedback. Additionally, function F2 and F3 are creating user output to inform the user about progress and errors in the system. F5 is controlling the zeroing of the hand. It sends desired values indirectly to the controller function (F3) through function F1. It controls the power states of the actuators by using

data flow D4. Its main purpose is to check all sensors and drive the fingers in a known position before beginning of operation. When the zeroing procedure is completed, F5 receives a message from the controller via D12 and switches the hand to the normal control mode (compliant control) via the data flow D10.

Data flows:

- D6 controller gains; for the compliant controller
- D7 desired values; from user in SI units and organized in buses
- D8 checked desired values; for the impedance controller
- D9 calibrated and converted values; from the measurements
- D10 desired zeroing parameter, controller mode; to drive the fingers to the reference position
- D11 controller state, provides information about the reached desired values

Data flow of the function F1

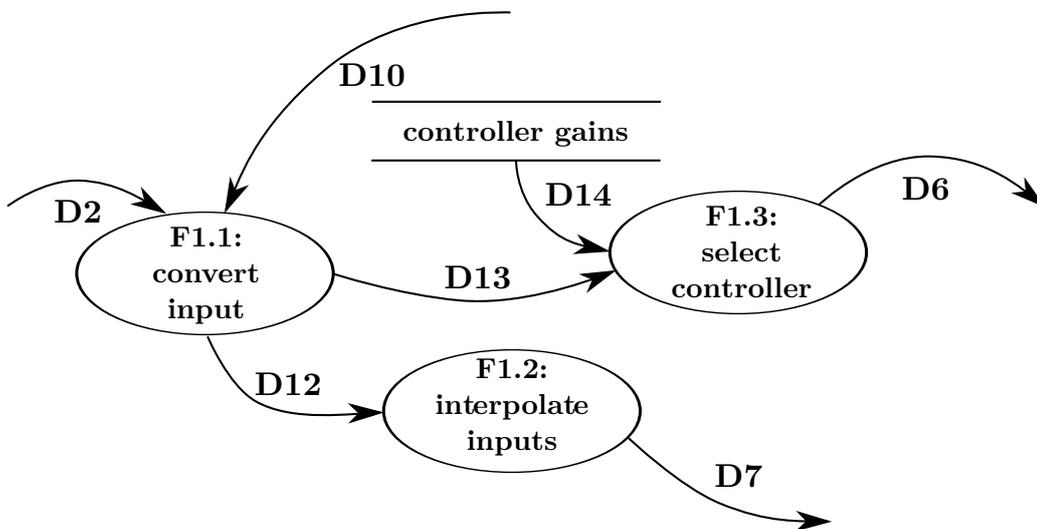


Figure 3.4: Data flow analysis F1

The function F1 controls the different inputs (figure 3.4). F1.1 converts the incoming signal D2 of the user or the input of the zeroing function F5 to suitable units. It also separates the input of data flow D2 into the desired

control mode in data flow D13 and desired variables for the system in data flow D12. The function F1.3 selects the correct gains from the database depending on the input of data flow D14. The function F1.2 interpolates the user inputs to transform the signals in the controller frequency and sends the data via data flow D7.

Data flows:

- D12 converted desired values; for controllers.
- D13 desired controller type; chosen by the user or zeroing block
- D14 controller gains and settings; for different types of controllers

Data flow of the function 2

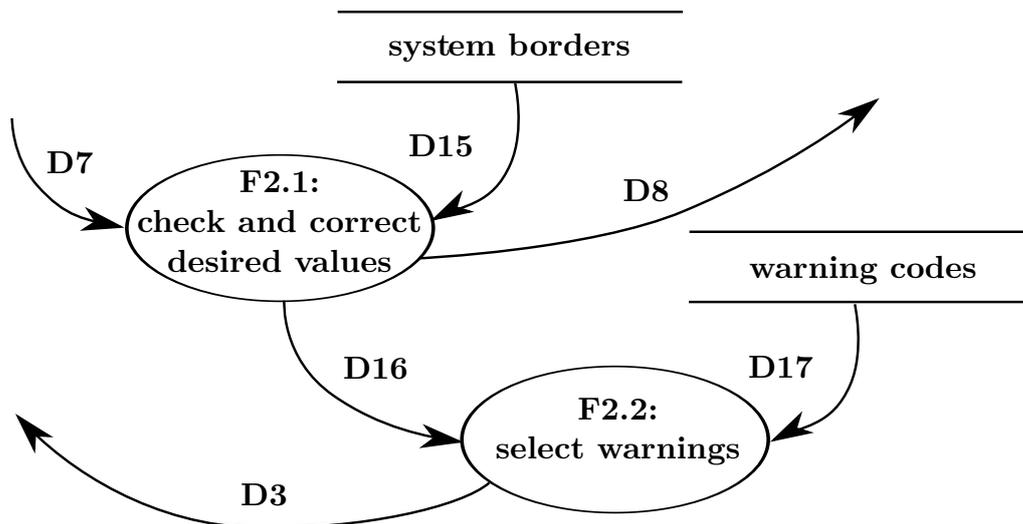


Figure 3.5: Data flow analysis F2

Function F2 (figure 3.5) describes the safety level which ensures that no user input destabilizes the system. In function F2.1, the desired values (e.g. joint position, velocities, stiffness) are checked if they are within the range. In case of an input outside the system borders, the value is corrected and the error information is given to function F2.2 via D17, where a warning code is selected from the warning database and sent back to the user via D3. The corrected and suitable values are given to next function F3 via data flow D8.

Data flows:

- D15 controller gains; for impedance controller.
- D16 desired values; from user in the correct units and structure
- D17 checked desired values; for the impedance controller

Data flow of the function 3

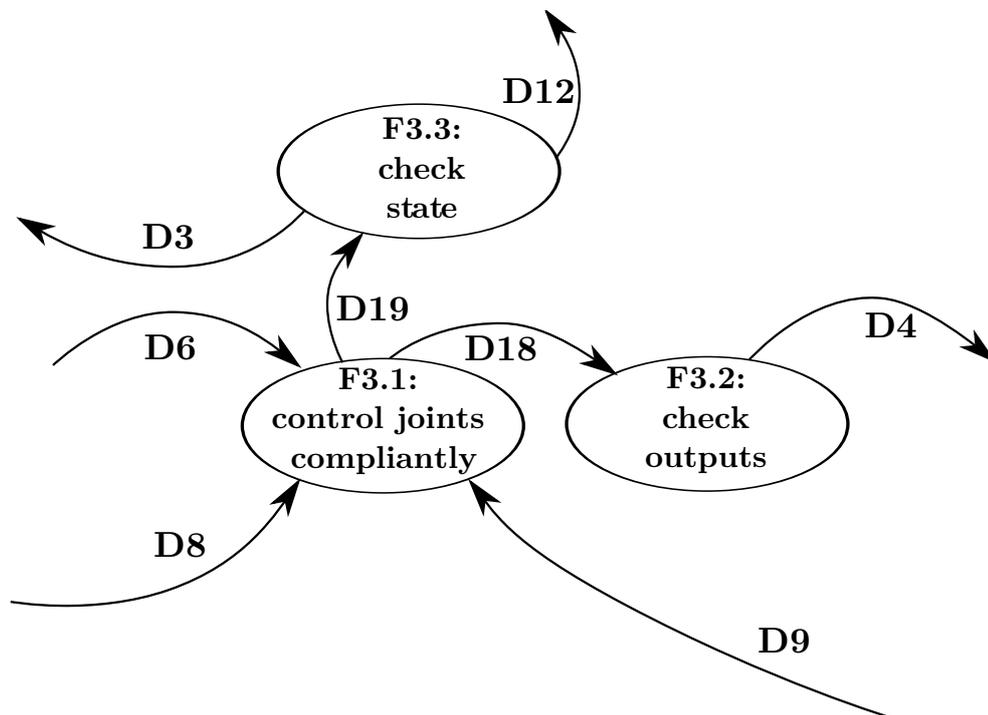


Figure 3.6: Data flow analysis F3

In figure 3.6, the core of the controller is specified. The choice of the controller is determined later, within the discussion and testing chapter 5. Therefore the specification of the function 3 can only be done in general. It mainly contains three functions: the main controller in function F3.1, which receives controller gains from data flow D6 and desired values from user using data flow D8; the feedback of the measurements and observed quantities come from D9. Observed data is computed in function F4 and extends the measurements by quantities that can be determined from the model knowledge. The controller output values in data flow D18 are checked

by function F3.2. The same output is send via data flow D19 to the function F3, to check the controller state. Function F3.2 ensures that the low level controllers of the actuators do not receive values out of the feasible range. The final output is send to the hardware via data flow D4. The state of the controller such as "position reached" is send to function F5 via data flow D12 to ensure correct zeroing.

Data flows:

- D18 controller output; desired values for low level motor controller.
- D19 state output, e.g. position reached

Data flow of the function 4

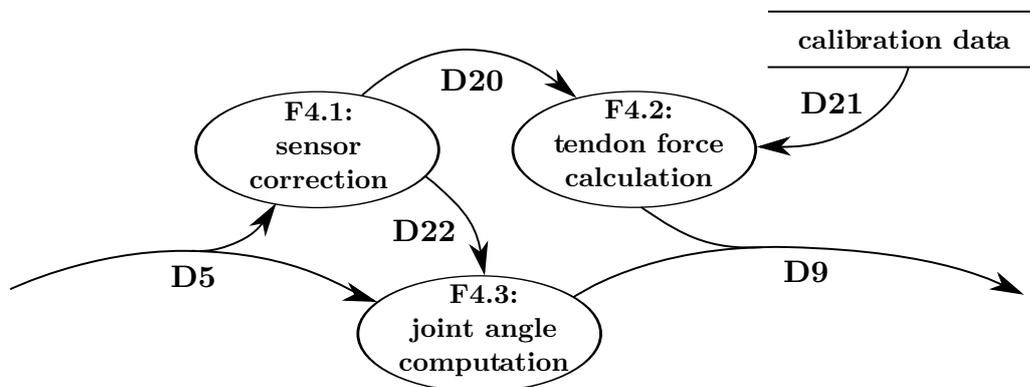


Figure 3.7: Data flow analysis F4

Function F4 (figure 3.7) has the main goal to calibrate the incoming measurements of the hardware from data flow D5. In function F4.1 the measurements of the magnetic hall sensors are corrected if they jump due to underflow or overflow. This corrected measurement is send via data flow D20 to function 4.2, where the tendon forces are calculated with the calibration data from data flow D21. The corrected measurement is also used to compute the joint angles of the fingers in function F4.3 send by data flow D22. The extended measurement data is send via D9 to the controller function F3.

Data flows:

- D20 corrected measurements; for force calculation.
- D21 calibration curves; for force calculation
- D22 corrected measurements; for joint angle computation

Data flow of the function 5

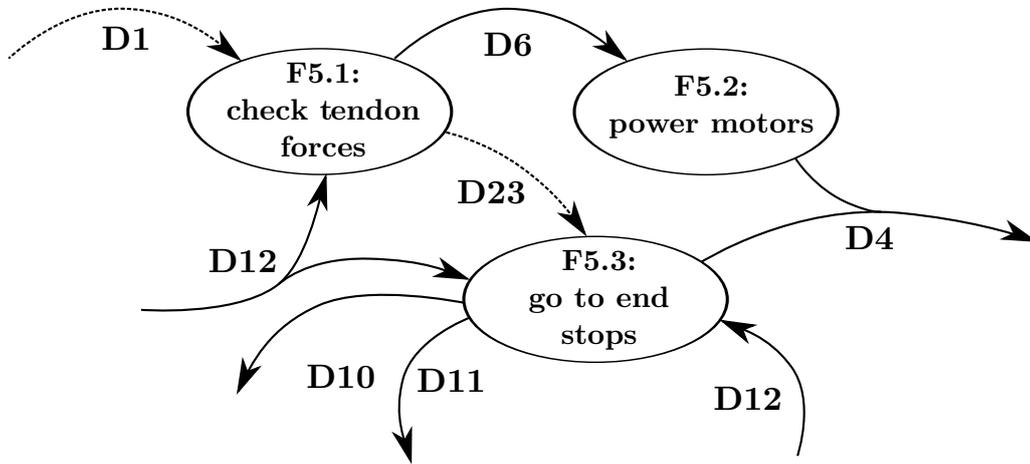


Figure 3.8: Data flow analysis F5

The zeroing is done in function F5 (figure 3.8). It receives a start command via data flow D1 and starts checking the force values in function F5.1 if they are in a good range, to check if they work properly. Function F5.1 sends the information, which motors work properly to function F5.2 via D6. F5.2 is sending a motor power vector, which contains a on/off signal for each of the 38 motors. Using data flow D23, the function F5.1 triggers also the function F5.3, which drives the fingers in the hardware end stops by giving the desired forces an unbalanced force setting via data flow D11 and setting the controller to only force control mode via data flow D10. If the joint position is not changing for five seconds, function F5.3 zeros the joint angles via data flow D4 and the controller is switched back to impedance position control via D10.

Data flows:

D23 start command; for end stop movement.

3.3 Functional constraints

To be able to implement the controller, some constraints are given:

Usage of the hand of Hand-Arm-System: The hand is provided by the institute of robotics of the DLR. It is part of the Hand-Arm-System but it must be usable separately.

Usage of Simulink: The realtime control of the hand software is generated by the realtime workshop of Simulink. C-code is generated from the Simulink model which is compiled to run on the real time system.

Usage of the Linux system All drivers and simulink are prepared to run on a Linux machine in connection with the real time system QNX.

Usage of the real time system QNX To run the hand, the hardware is connected to a real time system QNX, which runs the compiled Simulink code.

4 Modeling the anthropomorphic hand

Due to the anthropomorphic design of the hand the anatomical terms are used to describe the system. They are presented in the beginning of this chapter for a better understanding. Furthermore, the kinematics are derived for one finger as well as the equations of motion. To verify the theoretical model a simulation is compared to the real system using a frequency analysis. In the last section the different time scales in the system are tested.

4.1 Anatomical terms

One of the key ideas of the Hand-Arm-System was to design it as highly anthropomorphic as possible¹. Therefore, almost all human bones and joints can be found in the hand of the Hand-Arm-System. Only the tendons and ligaments of a human differ from the robot hand. A human hand is characterized in picture 4.2(a). The terms are explained in the following (according to [Bun09]).

4.1.1 Hand bones

The bone names are shown in figure 4.1. The human hand mainly consists of 27 bones, which can be divided into Carpals, Metacarpals and Finger phalanges. The Carpals build the connection between the wrist and the hand. The Metacarpals are bones which mainly build the base of each finger. Therefore, the human hand has five metacarpals. The finger Phalanges are the segments of each finger and can also be divided into three groups: proximal, middle and distal bones.

4.1.2 Hand joints

The joint anatomical names derive from the bone names. There are several joint types in the fingers, which are listed and shown below. Picture 4.2(a)

¹With the underlying hypothesis that this will enable the robot to perform the same tasks the human does in the environment that is optimized for the human.

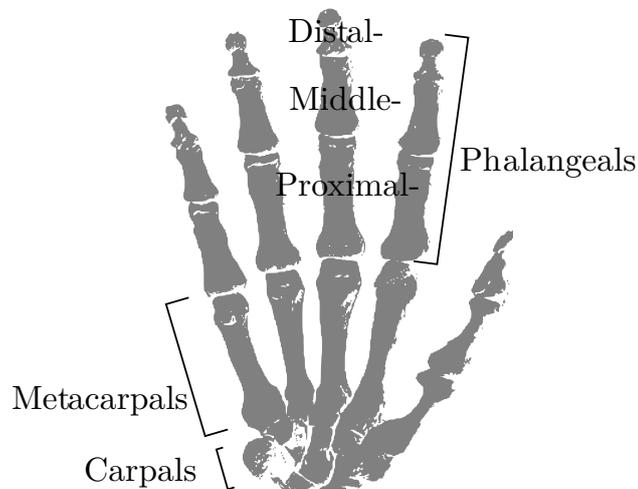


Figure 4.1: Hand Bones [Bun09]

shows also the similarity between a human hand and the hand of the Hand-Arm-System, which is shown without the housing. The hand with housing can be seen in figure 1.1.

Joint types:

- **DIP** - Distal Interphalangeal joints
- **PIP** - Proximal Interphalangeal joints
- **MCP** - Metacarpophalangeal joints
- **CMC** - Carpometacarpal joint
- **IP** - Interphalangeal joint

4.1.3 Finger motions

In addition to names of the joints, there also exist names for each direction of movement, which are depicted in figure 4.2(b). Closing the fingers is named flexion and opening is extension. For the side motion it depends if the finger move towards the middle hand axes or away from it.

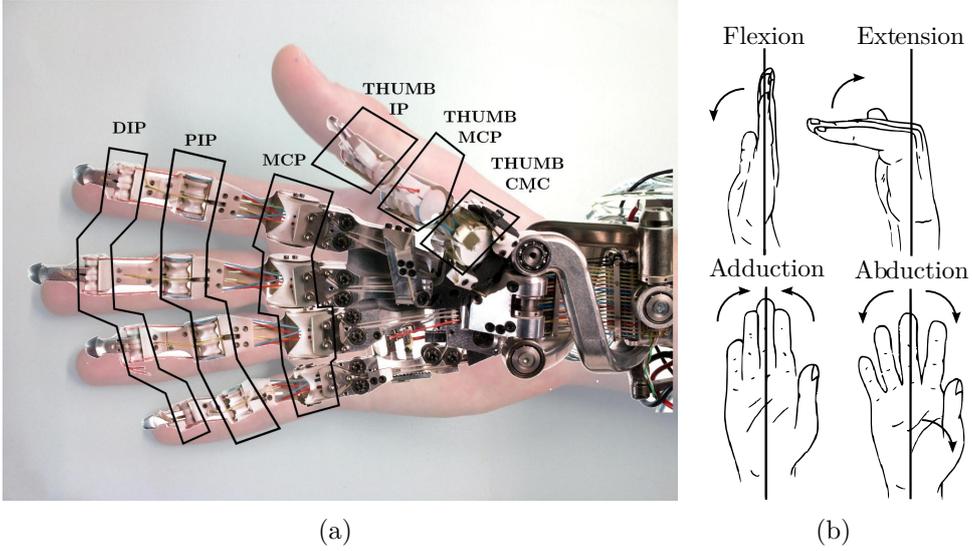


Figure 4.2: Left: Joints shown on human and robot hand; Right: Finger motion terms [Bun09]

4.2 System model of a tendon driven robotic hand

The system is a “multi DOF tendon-driven variable stiffness robot”, as described in [Wim+08]. First of all, the mappings are required to obtain the kinematics of the tendon forces \mathbf{f}_t and the joint motion \mathbf{q} . Later the general equations of motions are shown.

4.2.1 Tendon kinematics

The mapping between the motor angle $\boldsymbol{\theta}$ and the joint angle \mathbf{q} is obtained with two coupling matrices: the link side coupling (4.1) and the motor side coupling (4.4). The coupling of four fingers is independent of \mathbf{q} and thus the coupling matrix is also valid (equation (4.1)) for the tendon elongation \mathbf{h}_q and joint position \mathbf{q} . However, the thumb is dependent on \mathbf{q} , which increases the complexity of the computation. The special thumb kinematics are discussed in [Cha+11] in which parts of this work were used.

$$\dot{\mathbf{h}}_q = \mathbf{P}^T \dot{\mathbf{q}} \quad \text{for constant } \mathbf{P} : \quad \mathbf{h}_q = \mathbf{P}^T \mathbf{q} \quad (4.1)$$

For the coupling on link side each finger is divided in the base MCP joints (figure 4.4) and the finger joints DIP and PIP (figure 4.3). Indeed, the tendons are going through the base center of rotation and therefore, are not influenced by the base motion.

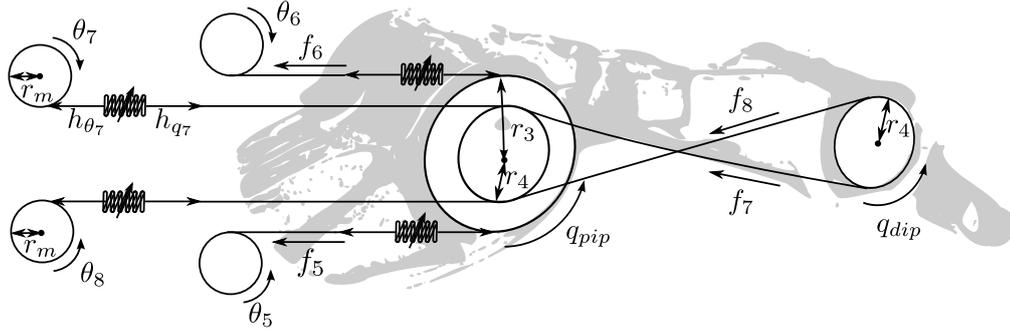


Figure 4.3: PIP and DIP joint coupling

$$\begin{pmatrix} \dot{h}_{5q} \\ \dot{h}_{6q} \\ \dot{h}_{7q} \\ \dot{h}_{8q} \end{pmatrix} = \begin{pmatrix} r_3 & 0 \\ -r_3 & 0 \\ -r_4 & r_4 \\ r_4 & -r_4 \end{pmatrix} \begin{pmatrix} \dot{q}_{pip} \\ \dot{q}_{dip} \end{pmatrix} \quad (4.2)$$

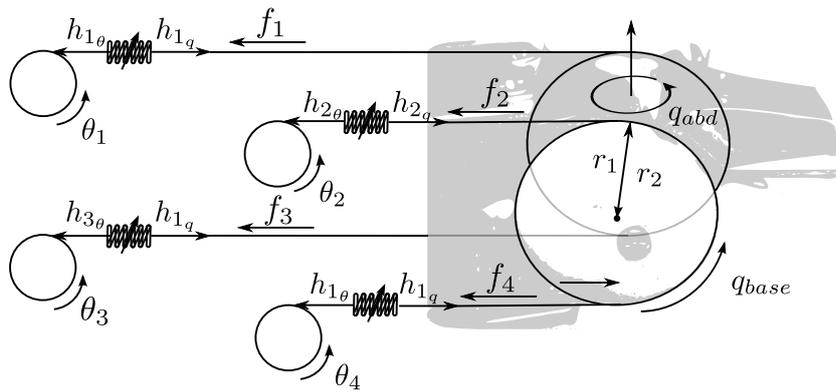


Figure 4.4: Base joint coupling

$$\begin{pmatrix} \dot{h}_{1q} \\ \dot{h}_{2q} \\ \dot{h}_{3q} \\ \dot{h}_{4q} \end{pmatrix} = \begin{pmatrix} r_1 & r_2 \\ -r_1 & r_2 \\ r_1 & -r_2 \\ -r_1 & -r_2 \end{pmatrix} \begin{pmatrix} \dot{q}_{base} \\ \dot{q}_{abd} \end{pmatrix} \quad (4.3)$$

The motor side coupling (Base: (4.5) and PIP/DIP: (4.6)) is a simple diagonal matrix with the motor pulley radius r_m , which can also be used for the tendon elongation and motor position mapping because they are not dependent on the motor position:

$$\dot{\mathbf{h}}_\theta = \mathbf{E}\dot{\boldsymbol{\theta}} \quad \text{with} \quad \mathbf{E} = r_m \mathbf{I}. \quad (4.4)$$

$$\begin{pmatrix} \dot{h}_{\theta_1} \\ \dot{h}_{\theta_2} \\ \dot{h}_{\theta_3} \\ \dot{h}_{\theta_4} \end{pmatrix} = \begin{pmatrix} r_m & 0 & 0 & 0 \\ 0 & r_m & 0 & 0 \\ 0 & 0 & r_m & 0 \\ 0 & 0 & 0 & r_m \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{pmatrix} \quad (4.5)$$

$$\begin{pmatrix} \dot{h}_{\theta_5} \\ \dot{h}_{\theta_6} \\ \dot{h}_{\theta_7} \\ \dot{h}_{\theta_8} \end{pmatrix} = \begin{pmatrix} r_m & 0 & 0 & 0 \\ 0 & r_m & 0 & 0 \\ 0 & 0 & r_m & 0 \\ 0 & 0 & 0 & r_m \end{pmatrix} \begin{pmatrix} \dot{\theta}_5 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \\ \dot{\theta}_8 \end{pmatrix} \quad (4.6)$$

4.2.2 Derivation of the equations of motion

From the conservation of power and the inverse kinematics, one can derive the mapping between tendon forces \mathbf{f}_t and joint torque $\boldsymbol{\tau}_q$. The tendon forces are nonlinearly depending on joint \mathbf{q} and motor angles $\boldsymbol{\theta}$. An example of such a nonlinear curve is shown in figure 6.3. In the next step, the model equations for the model are presented. Conservation of power:

$$\dot{\mathbf{h}}_q^T \mathbf{f}_t = \dot{\mathbf{q}}^T \boldsymbol{\tau}_q \quad (4.7)$$

Tendon kinematics $\mathbf{h}_q(\mathbf{q})$ give:

$$\dot{\mathbf{h}}_q = \mathbf{P}^T \dot{\mathbf{q}} \quad (4.8)$$

Inserting (4.8) in (4.7) gives:

$$\boldsymbol{\tau}_q = \mathbf{P}\mathbf{f}_t \quad (4.9)$$

The equations of motions are formulated according to [MLS94]. The following assumptions are made: No inertial coupling between the motors and the joints, since the motors are fixed in the forearm; the coupling matrix \mathbf{P} has full row rank and the pulling constraint for tendon forces \mathbf{f}_t are fulfilled.

The link side system equation is:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_{q,fric} + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau}_q + \boldsymbol{\tau}_{ext}, \quad (4.10)$$

where $\mathbf{M}(\mathbf{q})$ is the link inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$ are Coriolis and centrifugal terms, $\boldsymbol{\tau}_{q,fric}$ are joint friction torques and $\mathbf{g}(\mathbf{q})$ is a vector of gravity torques. On the right side the joint torque vector is $\boldsymbol{\tau}_q$ and the external torques are $\boldsymbol{\tau}_{ext}$, which are exhibited by the environment. The equation can be extended by inserting $\boldsymbol{\tau}_q$ of (4.9):

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_{q,fric} + \mathbf{g}(\mathbf{q}) = \mathbf{P}\mathbf{f}_t + \boldsymbol{\tau}_{ext}, \quad (4.11)$$

The motor side equations are:

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau}_{\theta,fric} + \mathbf{E}\mathbf{f}_t = \boldsymbol{\tau}_m \quad (4.12)$$

where \mathbf{B} is the motor inertia matrix, $\boldsymbol{\tau}_{\theta,fric}$ are motor and tendon friction, $\mathbf{E}\mathbf{f}_t$ are torques produced by the tendon forces and $\boldsymbol{\tau}_m$ are the motor torques.

4.3 System model validation

To verify the system model correctness the real system and a simulation with the theoretical model were analyzed using a frequency analysis method described in [Jan09]. Both systems were tested in open loop and the transfer functions of the outer system $G_{outer} = \frac{q(j\omega)}{\theta(j\omega)}$ were obtained (figure 4.5). For the outer system an external hall sensor, which was applied for verification only, was used to measure the joint motion on the link side.

The nonlinear system has adjustable stiffness and therefore different system behaviors can be obtained by adjusting the system stiffness. To minimize the influence of the nonlinearity of the stiffness, a small amplitude was chosen in the vicinity of the working point. For the actuation a sine with rising frequency and constant amplitude was used (equation (4.13)). The input and

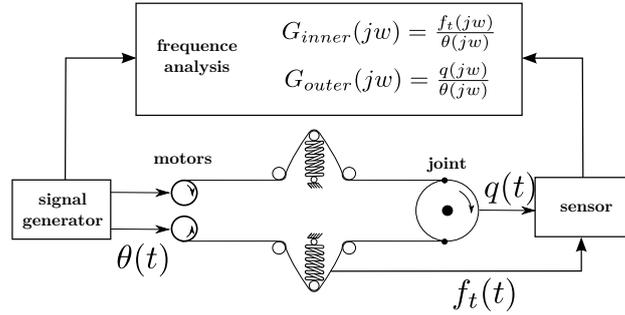


Figure 4.5: Applied method of [Jan09]

output signals were recorded and analyzed by a script, which generated the bode plot.

$$\theta(t) = \theta_0 \sin(\omega t) \quad \text{with} \quad \omega \in [\omega_{min}, \omega_{max}] \quad (4.13)$$

Results If the viscous friction in the joint $\tau_{q,fric}$ is described by a stickslip term, the simulation bode plot fits closely to the real model (figure 4.6). It gives evidence that the mathematical model behaves similar. The real model therefore has a PT_4 behavior around the working point. Variations appear from uncertainties in the mathematical model, e.g. unmodeled tendon behavior or unmodeled sensor behavior. Additionally, the simulation applies a unique nonlinear spring element curve for all tendons, which could be improved by using the calibrations curve of the real system. If the measurements are conducted for other working points, e.g. different stiffness or different start positions, the friction has to be tuned to fit the real system. That means, if complete modeling in the full working range is desired, the joint friction and tendon friction has to be measured in the stiffness range and joint position range for all 38 tendons and 19 joints.

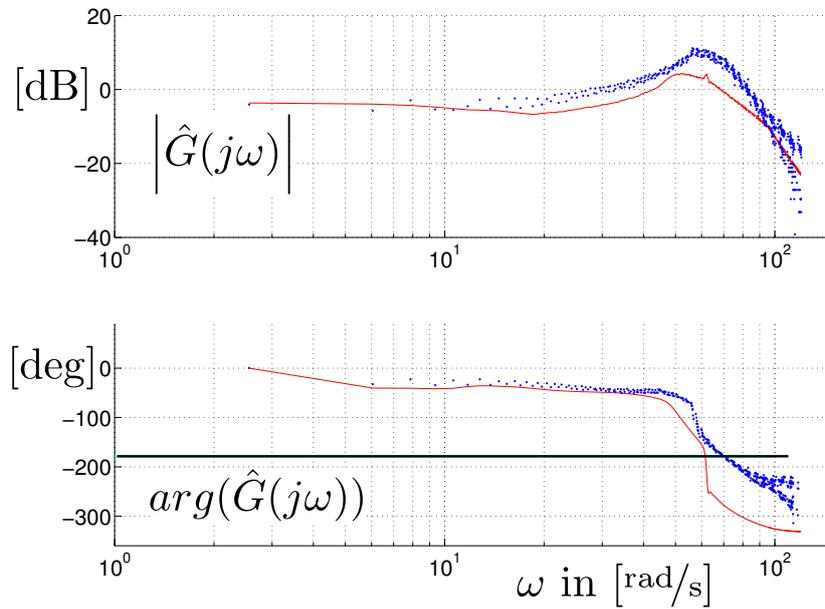


Figure 4.6: Comparison of the simulation (red line) and the real system (blue dots)

4.4 Cascaded system validation

The frequency analysis (figure 4.5) can also be applied to show that a cascaded two loop system can be used to control the motor and link side system separately. Therefore, the transfer functions of the real inner $G_{inner} = \frac{f_t(j\omega)}{\theta(j\omega)}$ and real outer system $G_{outer} = \frac{q(j\omega)}{\theta(j\omega)}$ are compared. The bode plots in figure 4.7 show that the outer system phase lack is at frequency $\omega_{-180} = 60 \text{ rad/s}$ and the inner system phase lack is at $\omega_{-180} = 120 \text{ rad/s}$. Thus the difference is high enough to control the two systems separately and a cascaded controller design is justified for this working point. Note, that by consideration of larger joint stiffnesses the bandwidth of the outer loop controller will come closer to the one of the force controller..

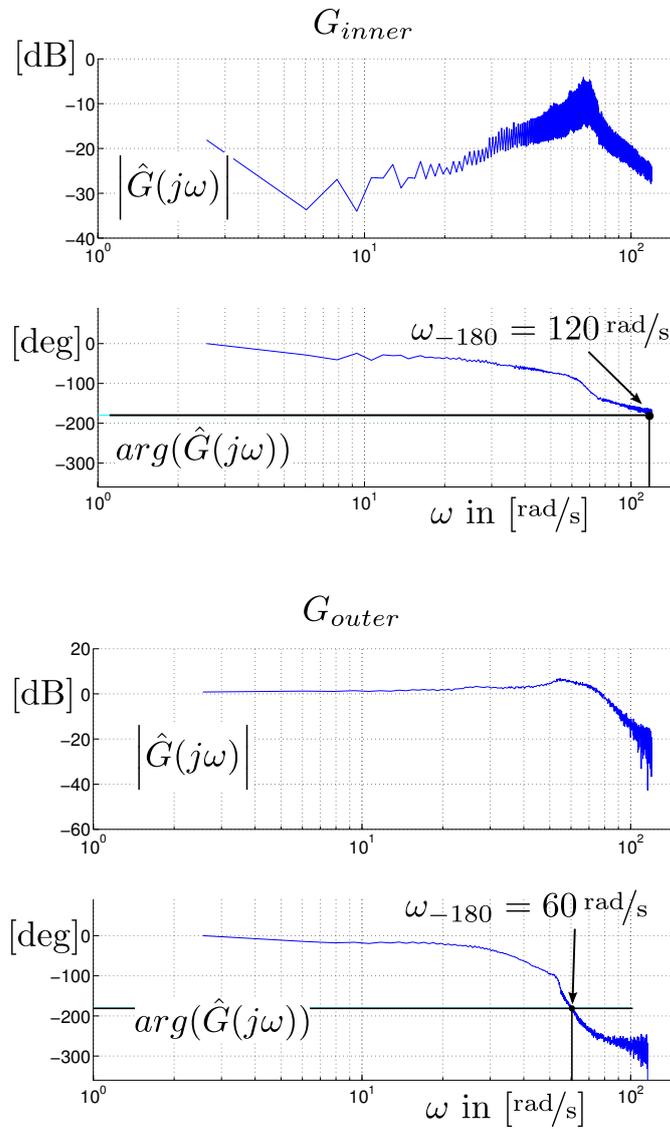


Figure 4.7: Frequency analysis of inner (top) and outer real system (bottom)

5 Design and discussion of compliant controllers

In this chapter different control approaches are presented. The controller designs are developed based on the system model from chapter 4.2. The model was compared to the real system for validation. Computing analytically the gains from the model, requires an extensive parameter identification. This thesis concentrates on the general design and therefore the gains are tuned manually for the best performance. The discussion is divided into the design of a compliant joint position controller and how the stiffness is distributed and controlled in the system. All methods are systematically tested on the real system and their behavior is compared. Finally, the controller is chosen which suits most for the user specification.

5.1 Compliance controller

In this section two compliance control methods and their derivatives are discussed. The advantages and disadvantages are outlined and one solution is chosen. All designs are tested on the real system. The Matlab Simulink implementation of the control design is detailed in chapter 6. In figure 5.1, an overview of the controller designs is given and how the separate controllers are related to each other. The discussed controllers are cascaded controllers based on the two-timescale-characteristic of the general model in section 4.2.2. In section 4.4 the inner and outer system are compared using a frequency analysis. The different phase lack shows that the two system can be controlled separately.

One important aspect of the system is that all quantities for control are not directly measured. Figure 5.2 shows the measurable quantities and the ones needed for control.

5.1.1 Admittance control

The first controller (figure 5.3) is a common approach which is used to achieve compliance behavior on industrial robots that have a stiff low level position controller, prohibiting direct access for the user. Unlike an impedance controller, the admittance describes a ratio of position to force or torque.

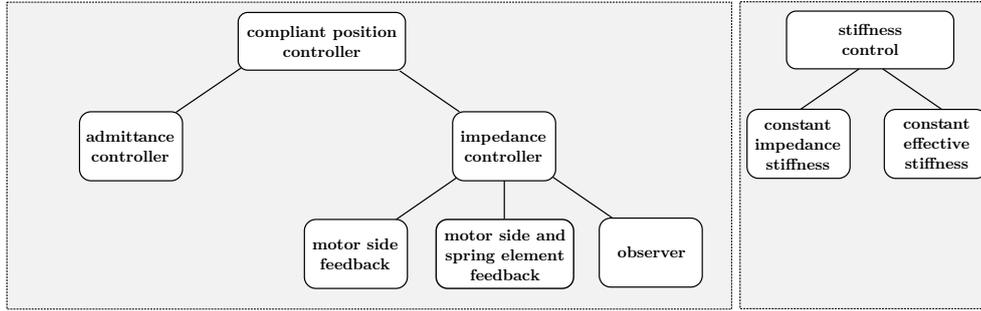


Figure 5.1: Discussion overview

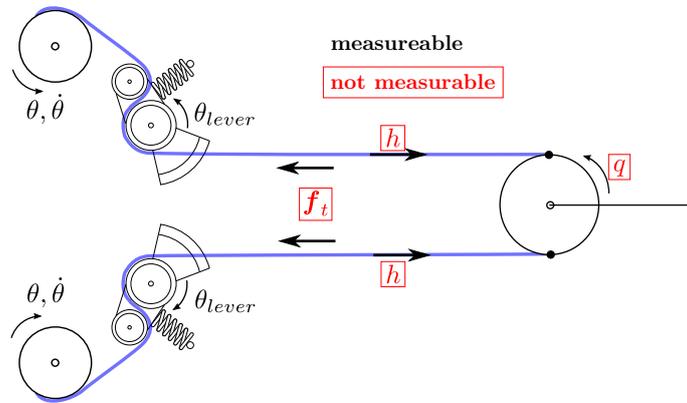


Figure 5.2: Measurements of the system; measurable quantities are in normal font; non measured quantities are written red and in a box

A torque is given and the admittance controller computes a change of the motor position such that the link behaves like a spring-mass-damper-system. The motor is moving away from its current position θ_0 to minimize the torque in the joints, this is emulating a compliant behavior. If the gain of the admittance controller is set to zero, only the inner position loop is active and the joint feels as stiff as the mechanics. If the gain would be infinite the joint would be in zero torque mode¹. The controller equation (5.1) of the outer admittance loop shows that the desired joint positions are obtained from the torque error. The inner position PD controller (5.3) uses the derived θ_{des} of the admittance loop and the desired motor positions θ_0 computed from the desired q_{des} joint position using the coupling matrices. For the admittance controller a solely compliance behavior is used instead of

¹In zero torque mode the joints can be moved freely, e.g. for position teaching tasks.

a spring-mass-damper behavior:

$$\mathbf{q}_{des} = \mathbf{K}_{ad}^{-1}(\boldsymbol{\tau}_{des} - \boldsymbol{\tau}) \quad (5.1)$$

The desired joint position is transformed into desired motor positions:

$$\boldsymbol{\theta}_{des} = \mathbf{E}^{-1} \mathbf{P}^T \mathbf{q}_{des} \quad (5.2)$$

The computed desired motor positions are utilized for the underlying position controller equation, which delivers the motor torque:

$$\boldsymbol{\tau}_m = \mathbf{K}_p(\boldsymbol{\theta}_0 + \boldsymbol{\theta}_{des} - \boldsymbol{\theta}) + \mathbf{D}_p(\dot{\boldsymbol{\theta}}_{des} - \dot{\boldsymbol{\theta}}) \quad (5.3)$$

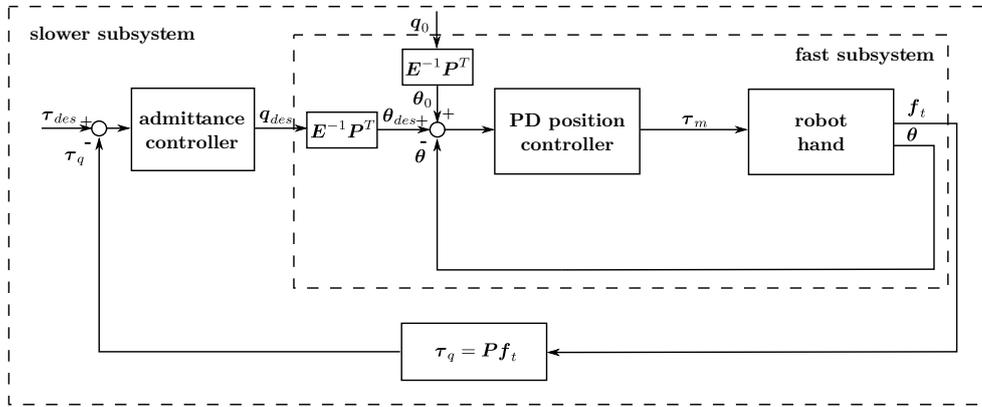


Figure 5.3: Admittance control

Results This admittance control concept shows disadvantages in the case of a tendon driven mechanism like the hand of the Hand-Arm-System hand. The tendons of each finger have to maintain a pulling force on the tendon. Otherwise the tendon could fall of the guiding. That constraint can be realized in the null-space of the coupling matrix, but has to be transformed into a motor position. Consequently, it is complex to keep a pulling force and stay at the desired position. Nevertheless, the simple admittance controller exhibits good compliant behavior but the tendons fall down from the guiding in some disadvantageous configuration. Another issue arises due to the significant stick-slip friction. The motors are approaching to the desired position, but due to the joint stick slip friction, the tendon forces increase and the spring

elements move. That leads to a dead-zone in the finger motion. In conclusion the stiff inner position controller results in almost infinite stiffness of the admittance controller, when the gain is zero. If the gain is increased the compliance rises. The compliance is limited to a specific gain due to sampling, quantification and modeling errors, while it is easier to realize high stiffness.

5.1.2 Impedance control

The impedance control is a well established strategy for providing compliant behavior. Its main idea is to span a virtual spring between the real finger position and the desired position. The impedance stiffness gain \mathbf{K}_{imp} represents the stiffness of the virtual spring, such that the equation $\boldsymbol{\tau}_{des} = \mathbf{K}_{imp}(\mathbf{q}_{des} - \mathbf{q})$ generates a desired joint torque proportional to the joint position error, which is the simplified impedance control, termed stiffness control. That creates a compliance in the joints, which depends on the gain. This behavior can be extended by injecting an additional damping term such that a desired mass-spring-damper relation is realized. In addition, an impedance controller is especially useful for grasping tasks, because a user can define a position inside the object, which is not reachable in real and when the fingers reach the outside of the object, the position error will produce a torque in the joints and apply pressure on the object. The torque $\boldsymbol{\tau}$ on the joints is created by pulling the two tendons such that the tendon forces transformed in the joint space apply such a torque. Therefore, an inner-loop force controller is needed. It has also the advantage of maintaining the force constraints (positive tension). The possibility to set a torque and maintain positive tendon force is given by the antagonistic setup explained in section 2.2. To realize the impedance controller, the joint position \mathbf{q} is required, which is not measurable in the Hand-Arm-System directly. The only sensors available are the position sensors of the motors and the position sensors of the spring elements. In the following, several methods are tested to compute the feedback of the joint position. For all approaches a constant mechanical stiffness is used to reduce disturbing side effects.

Feedback of motor side measurement

In the first case (figure 5.4) the impedance controller is using the joint position solely based on the motor position (as computed in (5.4)). Which allows the system passivity, because the controller can be designed such that no

additional energy can be brought in the system. The outer impedance loop computes a desired joint torque (5.5). This torque is transformed in the tendon space using the coupling matrix \mathbf{P} (5.6). The force controller is using a simple stiffness controller (P-gain) and a feed-forward term of the desired force (5.7) to reduce steady state error. To maintain the tendon constraints ($[\mathbf{f}_{t,min}, \mathbf{f}_{t,max}]$), the impedance controller output is saturated.

The joint position is computed from the motor position:

$$\mathbf{q} = \mathbf{P}^{+T} \mathbf{E} \boldsymbol{\theta} \quad (5.4)$$

This joint position is used for the impedance controller, which has a proportional and a derivative element:

$$\boldsymbol{\tau}_{des} = \mathbf{K}_{imp}(\mathbf{q}_{des} - \mathbf{q}) + \mathbf{D}_{imp}(\dot{\mathbf{q}}_{des} - \dot{\mathbf{q}}) \quad (5.5)$$

The output of the impedance controller is transformed into desired tendon forces:

$$\mathbf{f}_{t,des} = \mathbf{P}^+ \boldsymbol{\tau}_{des} + \mathbf{f}_{int}, \quad (5.6)$$

where the internal forces \mathbf{f}_{int} are chosen high enough to fulfill the pulling constraint of the tendons. For the inner loop a force controller with a proportional and a feed-forward term is used.

$$\boldsymbol{\tau}_m = \mathbf{E}(\mathbf{K}_{force}(\mathbf{f}_{t,des} - \mathbf{f}_t) + \mathbf{f}_{t,des}) \quad (5.7)$$

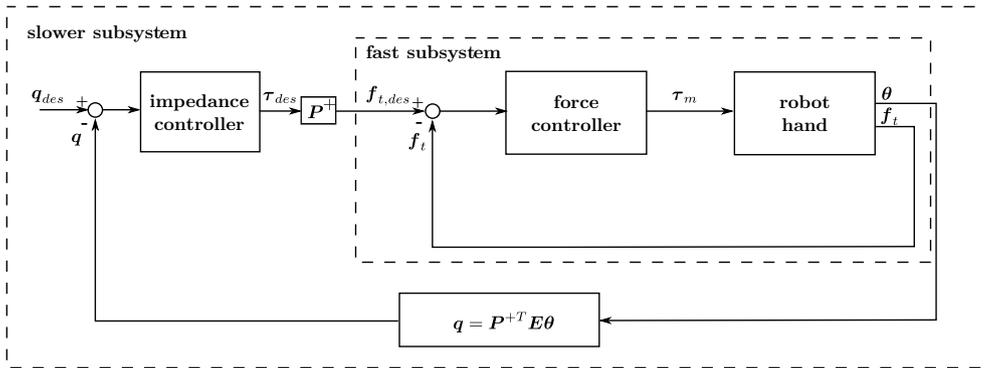


Figure 5.4: Impedance controller on motor side

Results The impedance controller shows good compliant behavior. By setting the impedance gain \mathbf{K}_{imp} to zero, the force controller performs a zero torque control: if the finger is moved the force controller releases one tendon and pulls the other one to maintain the desired force in each tendon. If there were no friction effects in the tendon guiding and in the joints, the finger would be infinitely compliant. If the impedance gain is increased the compliance is reduced. The maximum of the gain is related to the unmodeled dynamics and the discrete control. Similar to the admittance case, the computation of \mathbf{q} solely based on motor position results in a steady state error, because the elongation of the tendons in the spring element is not included by this computation. Due to friction the tendon forces did not return to their original state. Indeed, the spring elements have turned due to the changed tendon forces, which changes the current tendon elongation. This is not recognized by this controller and therefore it is not reacting if no position change is measured in the motors.

Feedback of motor and spring element motion

The tendon elongation due to spring element motion can be used (5.8), to improve the estimated joint angle, transformed from motor side measurement. A spring element is basically a lever that is moving around a point. When pulling a tendon, the spring element moves and modifies the tendon length of the system as shown in figure 5.5. This length is calibrated and accessible via a calibration function $\mathbf{h}_{lever}(\boldsymbol{\theta}_{lever})$. If the motor tendon elongation is added to this displacement (equation (5.9)), a more precise joint position can be computed. The controller equations and structure (figure 5.6) stay the same as in the very first impedance controller. In figure 5.2 it is also shown that the forces, which can not be measured directly, are calibrated from the spring element angle $\boldsymbol{\theta}_{lever}$.

$$\mathbf{q} = \mathbf{P}^{+T} \mathbf{h} \quad (5.8)$$

$$\mathbf{h} = \mathbf{E}\boldsymbol{\theta} + \mathbf{h}_{lever}(\boldsymbol{\theta}_{lever}) \quad (5.9)$$

Results The experiments show that the improved joint position on link side measurement, reduced the steady error, while using the same

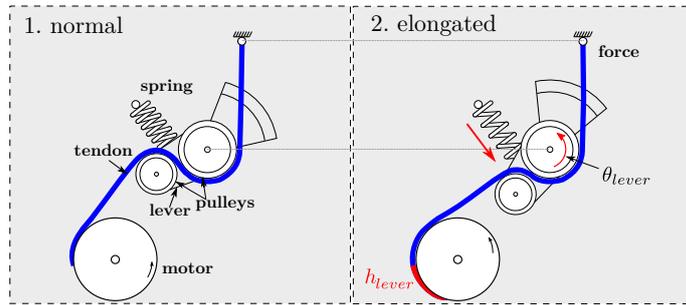


Figure 5.5: Spring element tendon elongation

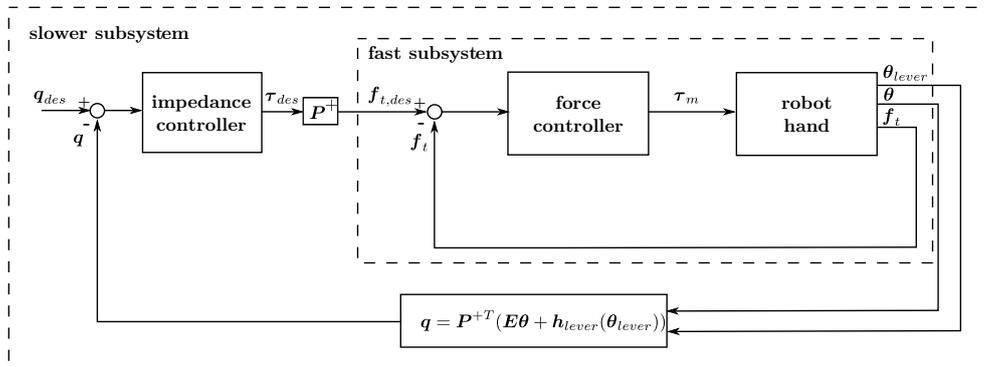


Figure 5.6: Impedance controller with extended joint angle feedback

impedance stiffness. The stiffness gain can be increased significantly before getting unstable. During larger steps the joint showed an underdamped behavior, which is due to the underlying stiff force controller, which is tuned to overshoots once before reaching the desired force. This overshoot causes additional tendon displacements resulting from the spring element position jump.

Feedback of motor side measurement with observer

Since the first approach 5.1.2 has a lack of precision in the joint angle measurement, it is extended by an observer. To preserve passivity only the position of the motor θ_m and the motor torque τ_m are used to estimate a more precise joint position q . The observer is basically the system model of chapter 4, in which the measured torque and motor position is inserted. The joint angle is extracted from the mathematical equations of 4.2.2. In figure

5.7 the extended design is shown. For the impedance controller the equation changes to:

$$\boldsymbol{\tau}_{des} = \mathbf{K}_{imp}(\mathbf{q}_{des} - \mathbf{q}_{obs}) + \mathbf{D}_{imp}(\dot{\mathbf{q}}_{des} - \dot{\mathbf{q}}_{obs}) \quad (5.10)$$

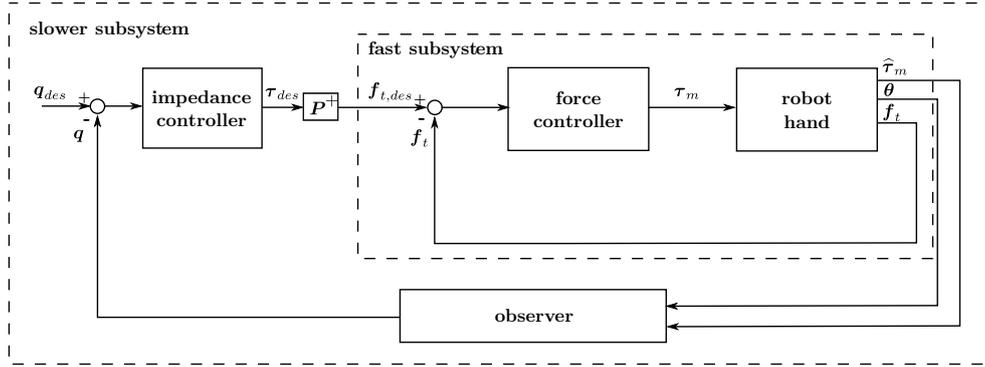


Figure 5.7: Impedance controller with observer

Results During the testing the observed joint angle \mathbf{q}_{obs} was compared to the real joint angle \mathbf{q}_{real} measured by an external magnetic hall sensor at the joint. This external sensor is not a regular part of the hand, but was used to validate the observer. Due to unknown friction components the damping of the observer model was tuned manually. As expected, the estimated position \mathbf{q}_{obs} in the steady state is an improvement over the motor based estimation of the first approach. Nonetheless the very fast changing motor torques produce high errors during dynamic motions. The observer was not able to model the real system precisely enough and consequently this approach was not followed anymore.

5.1.3 Compliant controller choice

Both the concepts of admittance and impedance are able to emulate compliant joints. If the admittance gain is zero the controller is infinitely stiff. The inner position controller allows no position error at all. However, in the case of flexible tendons, the stiffness is given by the mechanics. The admittance gain can be increased but the maximum gain is limited due to the control frequency. The impedance controller is the counterpart of the admittance controller. It starts with infinite compliance for a zero gain and its highest

stiffness is limited to the maximum impedance gain, which also is related to the control frequency, noise and unmodeled dynamics. In the case of the Hand-Arm-System it is essential to have zero torque mode available (infinite compliance). Therefore the impedance controller was chosen, which is infinite compliant in idle mode. It matches the assumed behavior of a human joint which is very compliant if it is not used. In addition to that the force constraints are easily fulfilled with the inner tendon force controller. For the admittance controller the constraints have to be transformed into motor positions for the inner position controller, which is more complex and requires more computation time. From the different impedance controllers the one using the motor and spring element feedback approach was chosen. It showed the best results which are reported in detail in chapter 7. In the other case, the spring element motion is not considered in the computation of the joint position, which is leading to an unacceptable steady state error. Finally, the observer precision could not be used due to the high sensitivity of the measured motor torque.

5.2 Control of the joint stiffness

As explained in section 2.2, the antagonistic set up provides an additional degree of freedom for the stiffness. It enables the system to set both a joint position and a mechanical stiffness at the same time. The impedance stiffness on software side and the mechanical stiffness on hardware side can be combined to one effective stiffness as shown in figure 5.8. The following sections discuss two possible approaches are discussed to control the different stiffnesses in the joints.

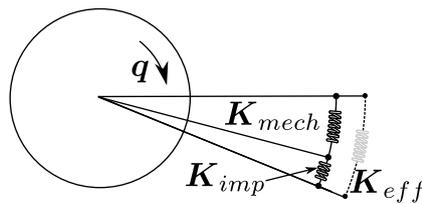


Figure 5.8: Stiffnesses serial connection overview

5.2.1 Constant impedance gain and constant pretension

A intuitive method to set the mechanical stiffness is to add a pretension force $f_{t,pre}$ on each tendon. The impedance gain of the controller can be set

separately. This way is very sparse in resources and the implementation can be easily done for all joints. Figure 5.9 shows the adjusted controller design of the chosen impedance controller in section 5.1.3. The force controller equation is extended by the pretension force:

$$\boldsymbol{\tau}_m = \mathbf{E}(\mathbf{K}_{force}(\mathbf{f}_{t,des} + \mathbf{f}_{t,pre} - \mathbf{f}_t) + \mathbf{f}_t) \quad (5.11)$$

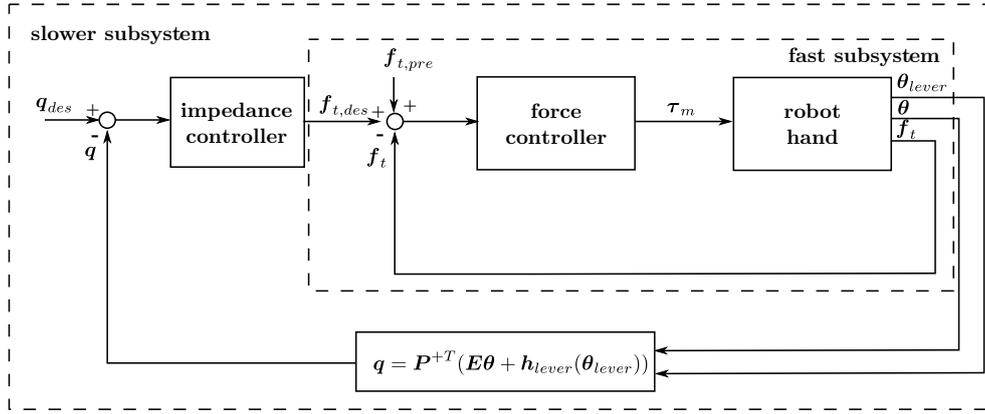


Figure 5.9: Impedance controller with pretension

Results The constant tendon pretension provides a simple way to set the stiffness manually. However, the pretension forces can create a joint torque if they are not selected in the coupling matrix null-space. Therefore, it is difficult for a user to select a set of pretension forces that does not influence the desired joint torque of the impedance controller. Moreover, the mechanical stiffness is only valid locally, because the tendon stiffness is not linear. It follows that the mechanical stiffness changes, if a user touches the joint and therefore he will not feel a constant stiffness. In fact the impedance gain is constant but due to the mechanical stiffness change the effective stiffness in the joints also changes. Similarly, a constant effective stiffness is not achieved during motions.

5.2.2 Controlling constant effective stiffness

A more sophisticated approach consists of computing the impedance stiffness and the mechanical stiffness in order to achieve the desired effective stiffness

(5.12), which is set by the user. Taking the constraint (5.14) into account the mechanical stiffness can be computed. To achieve the effective stiffness, the measured mechanical stiffness is used to compute the impedance gain (shown in equation (5.13)). The mechanical stiffness is achieved by tensioning the tendons in the null-space of the coupling matrix (section 2.2). The impedance controller of section 5.1.3 is used. The damping matrix \mathbf{D}_{imp} is computed proportionally to \mathbf{K}_{imp} to obtain a well damped behavior for any gain.

$$\mathbf{K}_{eff}^{-1} = \mathbf{K}_{imp}^{-1} + \mathbf{K}_{m,achieved}^{-1} \quad (5.12)$$

$$\mathbf{K}_{imp}^{-1} = \mathbf{K}_{eff}^{-1} - \mathbf{K}_{m,achieved}^{-1} \quad (5.13)$$

$$\mathbf{K}_{eff}^{-1} > \mathbf{K}_{m,desired}^{-1} \quad (5.14)$$

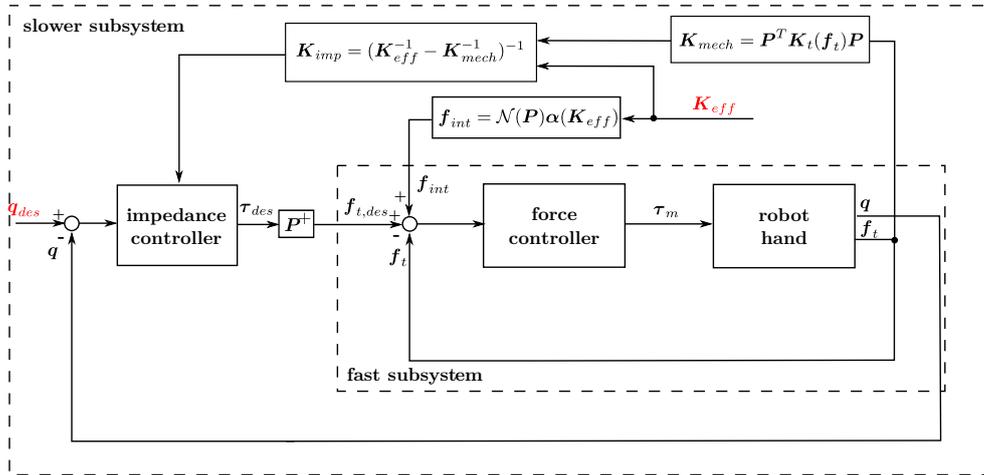


Figure 5.10: Impedance controller with effective stiffness adjustment

Results The controller in figure 5.10 is distributing the stiffness by selecting the pretension forces and the impedance gain. The finger compliance is maintained close to the effective stiffness. Using the measured mechanical stiffness $\mathbf{K}_{m,achieved}$ led to an unstable behavior due to the force sensor noise. Therefore the desired tendon forces $\mathbf{f}_{t,des}$ are used to compute the mechanical stiffness and the resulting impedance gain. This solution is stable and the effective stiffness is correct. The tracking of the desired tendon forces in the force controller is performed at such a high frequency that the force error does not influence the correctness of the effective stiffness.

5.2.3 Stiffness distribution controller choice

The manual stiffness distribution of the first controller, provides a simple solution to set the mechanical stiffness. However no constant effective stiffness can be achieved during motions or interactions. A more convenient interface consists of letting the user set only the desired stiffness (\mathbf{K}_{eff}) having the distribution done by the controller. This second approach is also providing an anthropomorphic behavior: A low effective stiffness is obtained with using low mechanical and impedance stiffness, which results in a high precision error. A higher effective stiffness improves the precision by reducing the influence of the joint and tendon friction. For the control module the second solution is retained due to the intuitive setting.

Verification of the control design

To verify the chosen control design, the closed loop equations are computed and analyzed at steady state under the assumption that the cascaded approach is valid. The closed loop tendon force controller equation is obtained by inserting the tendon controller equation (5.7) in the motor model equation (4.12), which yields:

$$\mathbf{B}\ddot{\boldsymbol{\theta}} + \boldsymbol{\tau}_{\theta,fric} + \mathbf{E}\mathbf{f}_t + \mathbf{E}(\mathbf{K}_{force}(\mathbf{f}_t - \mathbf{f}_{t,des}) - \mathbf{f}_{t,des}) = 0 \quad (5.15)$$

At steady state the velocity and the acceleration are canceled. The steady state error is:

$$\mathbf{f}_t - \mathbf{f}_{t,des} = \frac{\boldsymbol{\tau}_{\theta,fric}}{\mathbf{E}(1 + \mathbf{K}_{force})} \quad (5.16)$$

From (5.16) one can conclude that if \mathbf{K}_{force} is increased the steady state decreases. As previously mentioned the maximum of the gain is related to the unmodeled dynamics and the discrete control.

The outer control loop equation is obtained by inserting equations (5.6) in the dynamic model (4.11). The assumption of the different timescales yields $\mathbf{f}_t = \mathbf{f}_{t,des}$:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_{q,fric} + \mathbf{g}(\mathbf{q}) = \mathbf{P}(\mathbf{P}^+\boldsymbol{\tau}_{des} + \mathbf{f}_{int}) + \boldsymbol{\tau}_{ext} \quad (5.17)$$

Since the internal forces \mathbf{f}_{int} are in the nullspace of the coupling matrix \mathbf{P} , they cancel out. Furthermore, the equation (5.5) inserted in equation (5.17) yields:

$$\begin{aligned} M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \boldsymbol{\tau}_{q,fric} + \mathbf{g}(\mathbf{q}) = & \mathbf{K}_{imp}(\mathbf{q}_{des} - \mathbf{q}) \\ & + \mathbf{D}_{imp}(\dot{\mathbf{q}}_{des} - \dot{\mathbf{q}}) + \boldsymbol{\tau}_{ext} \end{aligned} \quad (5.18)$$

At steady state the equation (5.18) simplifies to:

$$\mathbf{q} - \mathbf{q}_{des} = \frac{\boldsymbol{\tau}_{q,fric} + \mathbf{g}(\mathbf{q}) - \boldsymbol{\tau}_{ext}}{\mathbf{K}_{imp}} \quad (5.19)$$

The result shows that the steady state error of the joint position depends on the friction torque, the gravity and the external torques, as well as the impedance stiffness. Because of the light weight finger design, the effects of gravity are significantly smaller than the joint friction. Thus, the controller does not compensate for the gravity. Similarly as mentioned in section 4.3, neither the joint friction modeling nor the joint friction compensation is discussed in this thesis. As expected the steady state error can be reduced, if the impedance gain \mathbf{K}_{imp} is increased. Increasing the effective stiffness increases the impedance stiffness and therefore reduces the steady state error. From the user perspective the stiffness adjustment results in the expected improved accuracy.

6 Implementation

In this chapter the setup of the hardware and the required changes for the control software are explained. In the following the implementation of the control structure and the components is explained.

6.1 Experimental setup

That section includes basic hardware structure as well as the steps required to run the system. The first part explains the parts required for the Hand-Arm-System and how they interact in software manner. In the second part the adaption of the control software is explained.

6.1.1 Hardware setup

To run the system three elements are needed (figure 6.1). The Linux system is running a Matlab Simulink program where the controller structure is designed. In the next step the realtime workshop generates C-code that is compiled afterwards. The executable file is started on a realtime system QNX which is connected to the Hand-Arm-System via SpaceWire (European Cooperation for Space Standardization standard ECSS-E-ST-50-12C). To be able to change the settings in the running model on the realtime system, the Simulink model on the Linux machine can be connected in external mode via TCP to the model running on the realtime system (QNX). Using the external mode connection the user is able to set constants and view scopes.

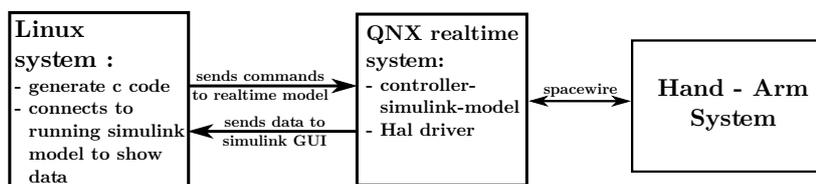


Figure 6.1: Hardware overview

The following connections depicted in figure 6.2 are required to enable the communication. The spacewire transformation box and the Hand-Arm-

System are plugged to a 24V source. The Linux machine with Simulink is connected to the realtime machine via TCP/Ethernet. The realtime machine is connected via optical spacewire to the transformation box, which sends an electrical signal to the hardware. The measurements of all 38 motors are exchanged over this connection.

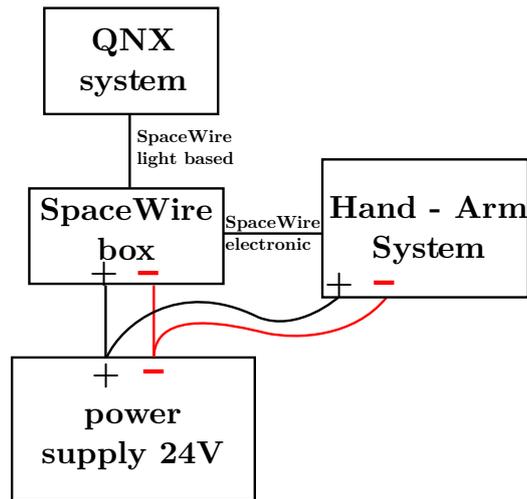


Figure 6.2: Connection overview

6.1.2 Adaption of the control software

The Simulink model has to be compiled to create executable files for the realtime system. Therefore several settings in the simulink model options have to be made: To use the Hand-Arm-System block a link to the simulink library in the network is needed. The paths are shown in table A. To load that before the model starts, a script can be used which loads the path and is executed in the preload callback of the Simulink model. This setting can be accessed in the model via the 'Model Properties'. After the compilation the driver of the Hand-Arm-System (HAL) and the model executable can be started on the QNX using 'ctrl+1'. To connect the model from the Linux machine to the running model, the model has to be set to external mode and connected with 'ctrl+t'. If the user intends to use scopes or displays they must be enabled them in the 'External control panel' in the 'Tools' section. However, too many displays can create a delay between the realtime model and the GUI model on the Linux machine.

Creating the bus system

All blocks are generally connected by the busses to have a clear view over the models. This allow the user to choose the required signals out of an organized tree of data. It also lowers the number of wires used in the model, thus keeping the visual complexity low. To be able to use this structured data in the embedded functions, the busses have to be created in the current workspace. That can be done by loading the saved Simulink bus objects using a '.mat' file. The mat file can be created manually by the user in the "bus manager" of Simulink. For the model a script was used which creates the busses out of a configuration file. So if the user changes the bus structure in the Simulink model, matching changes should be performed in the configuration file. The script has to be placed in the preload callback of the model (6.1.2). In addition to the bus object creation. The inputs of an embedded Matlab function have to be set to 'bus inputs'.

6.2 Implementing of the control model

In this section the structure and design of all important components of the model are detailed. The model is designed along the structure presented in chapter 3. Blocks used for calibration and conversions are also explained. To test the different types of controllers enumerated in chapter 5, all additional components are implemented in the same model and can be switched at runtime.

6.2.1 General control architecture

The model is designed to have the same structure as defined in chapter 3, which is shown in figure 6.4. The peripheral components are described in this section. It describes the implementation and explains the subsystems functionality. This knowledge is required for the advanced users that needs to access more than the control module possibilities. The controller implementation itself is shown separately in section 6.2.2.

Hand-Arm-System block

This block represents mainly the Hand-Arm-System terminator of the context analysis in 3.2.1. It contains a s-function which organizes the communication between the Hand-Arm-System driver (HAL) and the hardware hand. The

s-function version has to be the same as the HAL driver in the start script, which is started by 'ctrl+1'. The inputs that do not run at the 3 kHz of the realtime system have to be interpolated using rate transitions. Which output a signal at the desired frequency. The outputs of the s-function are organized in a named bus structure, thus only one output is needed.

Calibration block

The calibration block is related to function F4 of the data analysis (3.2.2) and receives the raw measurements of the hardware hand. As already explained in section 5, the force is measured via the deflection of the spring elements. The deflection of each spring element is measured via a off axis hall sensor. The sensors give increments which are nonlinearly mapped via a calibration look up table to the forces and the tendon length used for the deflection. One example of the calibration is shown in figure 6.3. The force f_t and the tendon elongation h_{lever} plotted over the increments (spring element angle) are both fitted via a seventh degree of freedom polynomial. Since the hall sensor is starting again at "0" increments, when reaching '4096' increments, the value can over/underflow depending on the arrangement of the hall sensor with respect to the spring element. Therefore the jump of the increments is canceled in a spring correction block and the result is given to the calibration look up table. The bus of measurements is then extended by the force of the tendons f_t and the tendon elongation of the spring element deflection h_{lever} .

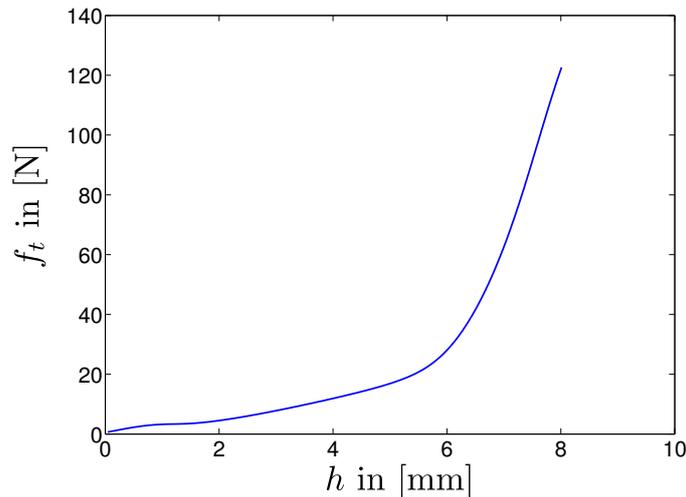


Figure 6.3: Force to tendon elongation relation of a spring element

Observer block

The observer block, which is the second part of function F4 of the data analysis (3.2.2), creates additional information from the system, which can be calculated from the model of the real system. For example, the joint position is computed via the coupling matrix \mathbf{P} and the motor position $\boldsymbol{\theta}$. This is an imprecise observed value due to the fact that the tendon is also elongated by the spring elements. Therefore a more precise \mathbf{q}_{close} is computed taking the spring lever motion into account. A third method is to calculate the joint position \mathbf{q}_{obs} based on the motor position and motor torque. These three approaches are presented in chapter 5 in section 5.1.2 and implemented in this block. All model variables can be set during runtime to tune the observer to the real system variables. The computed joint position are used as input for the impedance controller.

Source handling block

Since the user terminator of the structural analysis (section 3.2.1) can be a human, a block is needed to convert all incoming desired values to the proper units and organize the signals in a bus. To take over the control of the running model at any time a human operator can switch the input mode directly and choose between four modes (manual, GUIInput, data glove). The source block contains the databases defined in figure 3.4, which provides a set of controller gains for each controller.

Command checking block

This blocks is providing the functionality of function F2 in section 3.2.2 also shown in (figure 3.5). The system borders (joint limits, maximum and minimum forces) and warning codes are stored in the workspace via a script (HandData.m) which is also run in the preload callback of the Simulink model. The block checks that the desired values belong to a suitable range and prints out an error if not. Each joint is checked for its joint limits separately because they depend on the special design of each finger.

Zeroing block

In contrast to all other blocks this block is a Stateflow block, which performs the zeroing in several states. Stateflow is a build in tool in Simulink to design a chart with states. The block fulfills the constraint of the described function

F5 of the data analysis in section 3.2.2. Indeed, the finger joint position have to be zeroed to be used in the impedance controller. If a finger shall be enabled, the zeroing can be started via the zeroing button. The state machine checks if all force sensors work in an appropriate range (5 N-200 N) and starts to rise the desired force from the current system forces to a desired pretension. This is necessary to avoid loose tendons, which might fall of the guiding. If the user has selected “real zeroing” the fingers are driven in the end stops using an unbalanced force set on each joint. From this end stops the difference to zero position of each joint is known from the hardware design. The state machine switches the impedance controller on, which contains the needed offset on the \mathbf{q}_{des} . To avoid large discontinuities the impedance gains are rate-limited. The joints drive carefully to the zero position and the system is reinitialized again. The zeroing is completed and the system can be used. The command checking only works when using the ‘real zeroing’ mode, because the borders are calculated from the true zero position.

Controller block

The controller block related to function F3 of the structural analysis (3.3) contains the inputs of the source handling, command checking block and the calibration block. It outputs a desired current for the actuators. The detailed controller structure implementation inside the block is explained in the section 6.2.2.

6.2.2 Implementation of the controller elements

In the discussion chapter 5 some alternatives were chosen to be implemented on the real system. Thus, several blocks are needed to realize the different controller. The important parts are presented in the next sections.

Force controller

The force controller implements the inner loop of a general impedance controller as shown in section 5.1.2 in figure 5.6. The desired and measured forces are the main inputs. Finally the motor torque checking block checks that the desired variables are within the range of the electronic capabilities.

Impedance controller

The impedance controller receives a desired joint position and the measured joint position. The measured input can be switched between a motor based joint position, a link side joint position or an observed joint position. The block is splitted in to the wrist impedance controller, which is not discussed in this thesis and the finger impedance controller. The finger controller contains an embedded function for each finger separately. The finger blocks are able to perform both stiffness adjustments as discussed in chapter 5 in section 5.2. The mode can be switched separately for each finger. The computation of the desired force is executed in the embedded Matlab functions, depending on stiffness distribution algorithm of section 5.2. The simple pretension offset needs no additional computation and can be set directly via a constant. In case of the effective stiffness controller, the forces for the desired mechanical stiffness are needed. This is done by solving a nonlinear optimization problem using a gradient search, which finds the combined forces (internal and desired torque forces) to achieve that effective stiffness, while obtaining the desired joint torque. The desired force is used for the computation of the mechanical stiffness and is used in combination with the user effective stiffness to compute the new impedance gain.

Gradient Search

The gradient search (e.g. in [Sal98]) finds internal forces and the forces, which create the desired torque, such that the desired mechanical joint stiffness is achieved. The internal forces \mathbf{f}_{int} are linear combinations α in the nullspace of the coupling matrix. Since the mechanical stiffness is unique for each spring element the search algorithm is required to achieve the resulting mechanical joint stiffness. The gradient search uses the cost function (6.1) to compute the goodness of the following step. The cost function is computing the Euclidean distance of the mechanical joint stiffness of the current step and the desired mechanical stiffness:

$$C_i = \sum_{k=1}^n \left(\sum_{j=1}^n \left(\mathbf{K}_{m,i}(k,j) - \mathbf{K}_{m,des}(k,j) \right)^2 \right). \quad (6.1)$$

If the cost of the current step is smaller than the cost of the last step, the current step is valid and the direction and amplitude for the following step is computed using the gradient of the cost function. Thus, the search trends to

6 Implementation

the smallest cost and finds the forces which provide the closest mechanical stiffness to the desired stiffness as well as the desired torque.

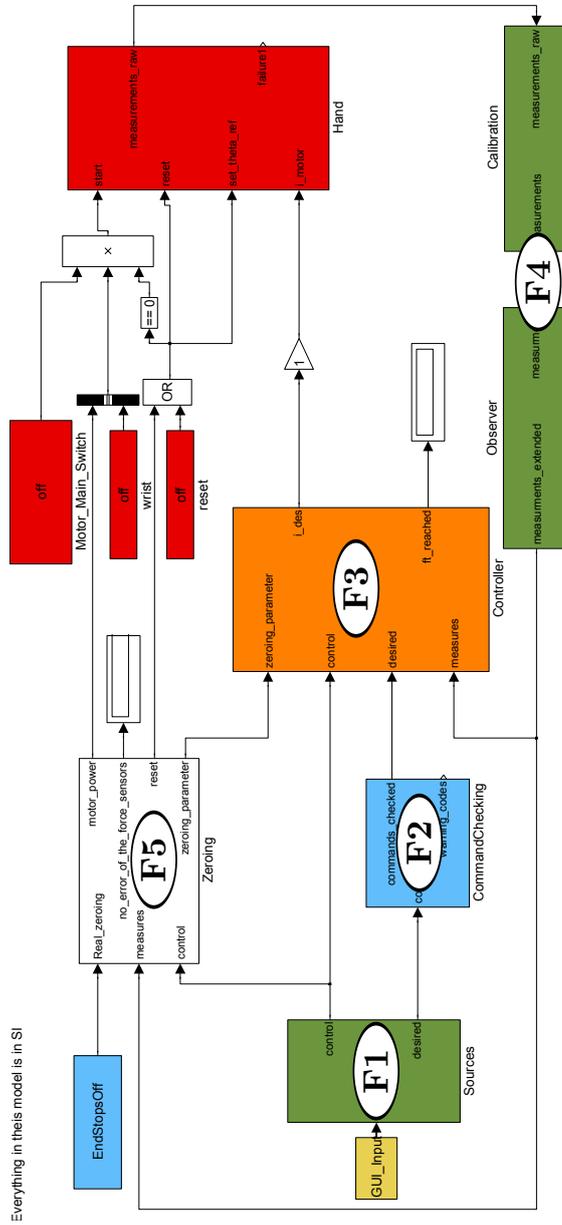


Figure 6.4: Simulink model

7 Validation

The purpose of this chapter is to validate the behavior of the controllers from the design and discussion chapter 5. Although the implementation is done for all fingers, the tests were made on one joint of a single finger testbed. This finger was optimized to reduce effects of friction in order to evaluate the controllers without the stick-slip effects. The performance on the full hand is lower, but the designers are working on reducing the stick-slip. At the time of the validation phase of the thesis only the testbed was available. The validation is separated into the step responses of different amplitudes at different effective stiffness, a frequency analysis similar to section 4.3 and the validation of the achieved effective stiffness.

7.1 Step response

To validate the precision of the chosen controller in figure 5.10, five steps on \mathbf{q}_{des} with different amplitudes and effective stiffnesses were performed on the joint (plots in figure B.1). The results showed the desired behavior (figure 7.1). If the joint is set to be very soft (small \mathbf{K}_{eff}), it is not able to follow a trajectory and the relative error is 100%. This mode can be used for interaction and teaching of trajectories and can be interpreted as zero force mode. If it is set stiffer it starts following the trajectory and it gets precise very quickly. Already at $K_{eff} = 0.2 \text{ Nm/rad}$ the steady error is around 10% and at $K_{eff} = 1.3 \text{ Nm/rad}$ it is in average 2%. The amplitude of the step has only a small influence. It is interesting to note that although it is a friction optimized joint, the stick-slip friction results in the same position error for each amplitude. That means the relative error for small steps is higher.

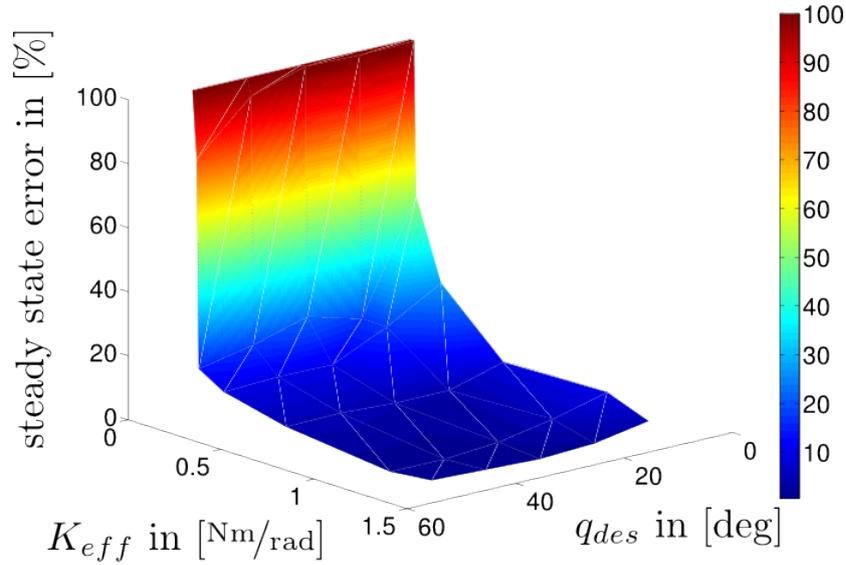


Figure 7.1: Error due to step amplitude and effective stiffness

7.2 Frequency analysis

The frequency analysis of the closed loop system shows in comparison with the open loop system that the controller steady state error is very small (0.01 dB). The resonance frequency of the open loop system $\omega_{0_{open}} = 68 \text{ rad/s}$ is also shifted to higher angular frequency in the closed loop system $\omega_{0_{closed}} = 93 \text{ rad/s}$. The phase reserve of the closed loop system at the crossover angular frequency ω_C is $\varphi_m = 155 \text{ deg}$, that validates a stable behavior for this working point. To get a full picture of the controller behavior this analysis can be performed in the full range of the stiffness setup. The local linearization is not anymore valid if the frequency analysis is done for large steps, because in this case the tendon forces are changing significantly and the non-linearities of the spring elements would have a large influence. In [Wan+03] a method is described which fits the nonlinear system at any time to a linear model with changing coefficients, which would provide the possibility to check the phase reserve over the full range of motion.

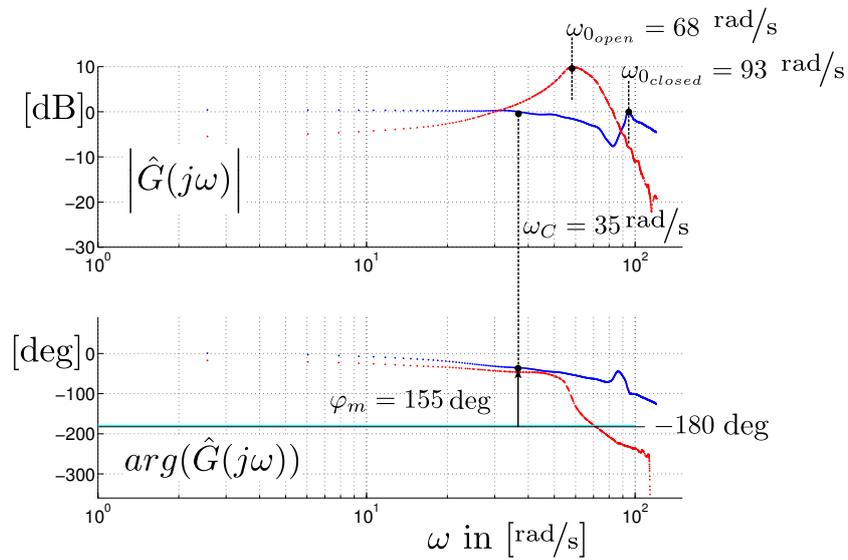


Figure 7.2: Comparison of the open (red) and the closed loop system (blue)

7.3 Stiffness control

To validate the stiffness control the joint was disturbed externally by random amplitudes and therefore the mechanical stiffness (red curve) changed. The impedance gain is computed online and therefore the effective stiffness stays constant (figure 7.3).

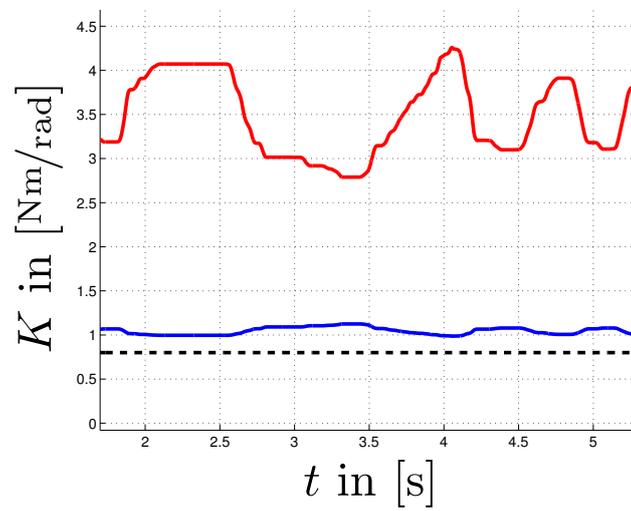


Figure 7.3: Red(top): Mechanical stiffness; Blue(center): Impedance stiffness;
Black(bottom): Effective stiffness

8 Conclusion

In this thesis the related work to tendon driven system with variable stiffness were outlined. To get a detailed view of the task, the requirement definitions were performed in a context and data flow analysis as well as the user specifications. In the next step the system was modeled and the model was compared to the real system behavior using a frequency analysis. Afterwards several approaches of compliant controllers and stiffness distribution methods were applied to the real system and compared, which led to a choice of a controller. In the following part the implementation of all controllers in Simulink were described by fulfilling the user specification and requirements definition. The implemented control software was tested on the chosen solution of the discussion part in different manners (position precision, dynamic behavior). For the control module the controller with only one tunable stiffness was chosen. The validation showed that the hand of the Hand-Arm-System is capable of performing its desired functionality of moving the joints and controlling the stiffness at the same time. The precision on a friction optimized one-finger-testbed showed the best performance. On the full hand the performance is restricted to the friction which is very complex due to the complex tendon guiding. This was expected by the designers. Therefore their desired performance of the system is reached. At the moment the designers redesign parts of the hand to decrease the friction effects in the full hand, such that a similar performance as in the one-finger-testbed can be achieved.

Future work

The model of the system could be extended by precise models of the tendon creeping behavior, sensor behaviors and the stick-slip effects. Additionally, the gradient search of the internal forces could be optimized to take less computation time. That would allow to run the outer loop also faster and therefore much higher gains. The wrist impedance controller has to be implemented, such that the fingers do not move, although the tendon lengths are changing because they are guided inside the wrist. For the user a GUI could be written to get a fast access to the functionality of the system. For

8 Conclusion

the cooperation with the arm of the Hand-Arm-System a coordination module could be implemented. Thus higher-level applications could be tested and programed.

Appendix

A Simulink required paths

Logging	/home/chal_ma/nopublic/SimulinkExtra/sfunctions_schm_fl /home/chal_ma/nopublic/SimulinkExtra/qnx_logger /home/rein_je/foreign_packages/sfunctions /home/rein_je/foreign_packages/make_mex
Gradient Search	/home/rein_je/Hasy/modeling/matlab/gradient_search
Links & Nodes	/home/chal_ma/workspace/robot_suite/library/simulink/ /home/chal_ma/workspace/robot_suite/library/simulink/.. /bus_tools /home/chal_ma/workspace/robot_suite/library/simulink/.. /bus_tools/tree_tools
External Mode Target	/home/odsw/matlab/2007b/0.8.0/library/dlrrm.. /home/odsw/matlab/2007b/0.8.0/rtw/c/dlrrm /home/chal_ma/workspace/robot_suite/library/simulink/
Hasy Hal	/home/joerg/projekte/hasy/0.3/simulink/units

Table A.1: Required paths

B Step response

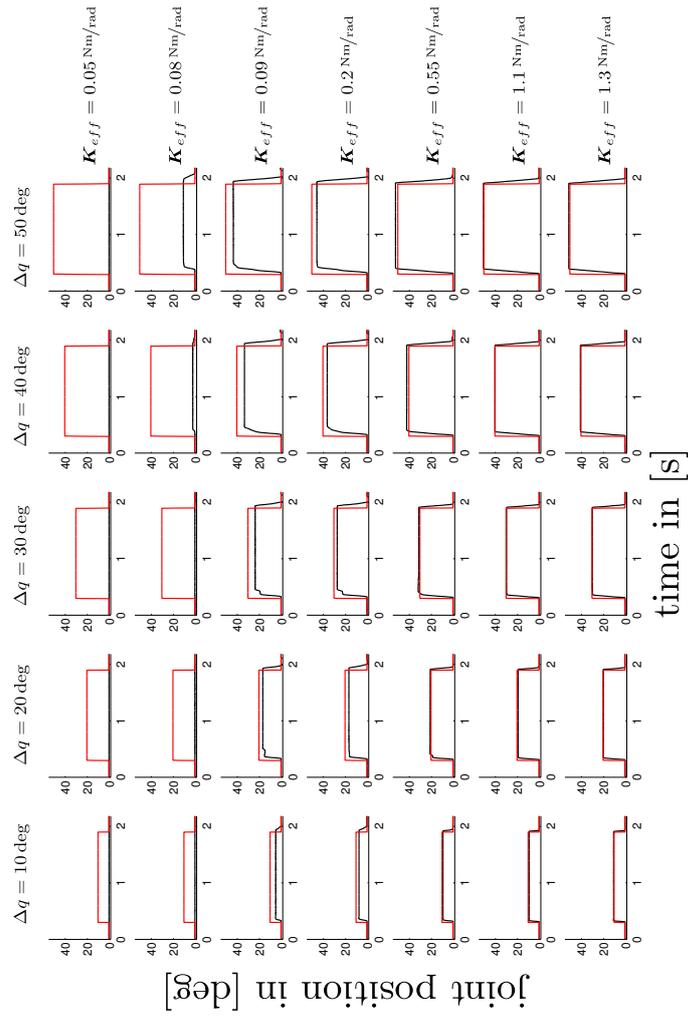


Figure B.1: Step response measurements

Bibliography

- [AO93] A. Ailon and R. Ortega. “An observer-based set-point controller for robot manipulators with flexible joints”. In: *Systems & Control Letters* 21.4 (1993), pp. 329–335. ISSN: 0167-6911. DOI: DOI : 10 . 1016 / 0167 - 6911(93) 90076 - I. URL: <http://www.sciencedirect.com/science/article/pii/016769119390076I>. (Cit. on p. 9).
- [ASH00] A. Albu-Schäffer and G. Hirzinger. “State feedback controller for flexible joint robots: A globally stable approach implemented on DLR’s light-weight robots”. In: *IEEE International Conference on Intelligent Robots and Systems*. 2000, pp. 1087–1094. (Cit. on p. 8).
- [Abd+10] M. E. Abdallah et al. “Applied joint-space torque and stiffness control of tendon-driven fingers”. In: *Proc. 10th IEEE-RAS Int Humanoid Robots (Humanoids) Conf.* 2010, pp. 74–79. DOI: 10.1109/ICHR.2010.5686289. (Cit. on p. 7).
- [BP00] A. Bicchi and D. Prattichizzo. “Analysis and Optimization of Tendinous Actuation for Biomorphically Designed Robotic Systems”. In: *Robotica* 18 (2000), pp. 23–31. (Cit. on p. 5).
- [Bun09] V. Bundhoo. “Design and evaluation of a shape memory alloy-based tendon-driven actuation system for biomimetic artificial fingers”. MA thesis. University of Victoria, 2009. (Cit. on pp. xv, 21–23).
- [CA09] M.T. Ciocarlie and P.K. Allen. “Hand posture subspaces for dexterous robotic grasping”. In: *The International Journal of Robotics Research* 28.7 (2009), p. 851. (Cit. on p. 2).
- [CPK03] R. Cortesao, J. Park, and O. Khatib. “Real-Time Adaptive Control for Haptic Manipulation with Active Observers”. In: *IEEE International Conference on Intelligent Robots and Systems*. 2003, pp. 2938–1943. (Cit. on p. 9).

- [Cha+11] M. Chalon et al. “Impedance control of a non-linearly coupled tendon driven thumb”. In: *IROS*. 2011. (Cit. on p. 23).
- [Cor+01] R. Cortesao et al. “Compliant Motion Control with Stochastic Active Observers”. In: *IEEE International Conference on Intelligent Robots and Systems*. 2001, pp. 1876–1881. (Cit. on p. 9).
- [DL00] A. De Luca. “Feedforward/Feedback Laws for the Control of Flexible Robots”. In: *IEEE International Conference on Robotics and Automation*. 2000, pp. 233–240. (Cit. on p. 8).
- [DLL98] A. De Luca and P. Lucibello. “A general algorithm for dynamic feedback linearization of robots with elastic joints”. In: *IEEE International Conference on Robotics and Automation*. 1998, pp. 504–510. (Cit. on p. 8).
- [DLR] DLR. *Internal documents*. Institute of Robotics and Mechatronics. (Cit. on pp. xv, 2–4).
- [Jan09] K. Janschek. “Systementwurf mechatronischer Systeme: Methoden-Modelle-Konzepte”. In: Springer, 2009. Chap. 2.7, pp. 168–173. (Cit. on pp. xv, 26, 27).
- [KB02] K. Koganezawa and S. Ban. “Stiffness control of antagonistically driven redundant D.O.F. manipulator”. In: vol. 3. cited By (since 1996) 4. 2002, pp. 2280–2285. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0036448892&partnerID=40>. (Cit. on p. 7).
- [KHO98] H. Kobayashi, K. Hyodo, and D. Ogane. “On tendon-driven robotic mechanisms with redundant tendons”. In: *International Journal of Robotics Research* 17.5 (1998), pp. 561–571. (Cit. on p. 5).
- [KOY04] Hitoshi Kino, Naruyasu Okamura, and Syuichi Yabe. “Basic Characteristics of Tendon-Driven Manipulator Using Belt Pulleys”. In: *Proceedings of 2004 IEEE International Conference on Intelligent Robots and Systems*. Vol. 2. 2004, pp. 1287–1292. (Cit. on p. 7).

-
- [LG95] T. Lin and A.A. Goldenberg. “Robust Adaptive Control of Flexible Joint Robots with Joint Torque Feedback”. In: *IEEE International Conference on Robotics and Automation*. 1995, pp. 1229–1234. (Cit. on p. 8).
- [LK+91] Laurin-Kovitz et al. “Design of components for programmable passive impedance”. In: vol. 2. cited By (since 1996) 41. 1991, pp. 1476–1481. URL: <http://www.scopus.com/inward/record.url?eid=2-s2.0-0025749428&partnerID=40>. (Cit. on p. 7).
- [MLS94] R.M. Murray, Z. Li, and S.S. Sastry. “A Mathematical Introduction to Robotic Manipulation”. In: CRC Press, 1994. Chap. 6.4, pp. 293–298. (Cit. on pp. 5, 26).
- [OL97] J.H. Oh and J.S. Lee. “Control of Flexible Joint Robot System by Backstepping Design Approach”. In: *IEEE International Conference on Robotics and Automation*. 1997, pp. 3435–3440. (Cit. on p. 8).
- [Ott08] Ch. Ott. “Cartesian Impedance Control of Redundant and Flexible-Joint Robots”. In: vol. 49. Springer Verlag, 2008. Chap. 5, pp. 59–68. ISBN: ISBN: 978-3-540-69253-9. (Cit. on p. 8).
- [PJ+11] R. Platt Jr et al. “Joint-Space Torque and Stiffness Control of Tendon-Driven Fingers”. In: *IEEE International Conference on Robotics and Automation*. 2011. (Cit. on p. 7).
- [Pal07] G. Palli. “Model and Control of Tendon Actuated Robots”. PhD thesis. DEIS, University of Bologna, 2007. (Cit. on p. 5).
- [Sal98] R. Salomon. “Evolutionary algorithms and gradient search: similarities and differences”. In: *Evolutionary Computation, IEEE Transactions on* 2.2 (1998), pp. 45–55. (Cit. on p. 51).
- [Tah+06] Kenji Tahara et al. “Bio-Mimetic Study on Pinching Motions of A Dual-Finger Model with Synergistic Actuation of Antagonist Muscles”. In: *IEEE International Conference on Robotics and Automation*. 2006, pp. 994–999. (Cit. on p. 5).

- [Tom91] P. Tomei. “A Simple PD Controller for Robots with Elastic Joints”. In: *IEEE Transactions on Automatic Control* 35 (1991), pp. 1208–1213. (Cit. on p. 8).
- [Wan+03] L. Wang et al. “Time-frequency analysis of nonlinear systems: the skeleton linear model and the skeleton curves”. In: *Journal of vibration and acoustics* 125 (2003), p. 170. (Cit. on p. 56).
- [Wim+08] T. Wimböck et al. “Impedance Control for Variable Stiffness Mechanisms with Nonlinear Joint Coupling”. In: *IEEE International Conference on Intelligent Robots and Systems*. 2008, pp. 3796–3803. (Cit. on pp. xv, 7–9, 23).
- [Wim09] T. Wimmer. *The European Proximity Operations Simulator (EPOS)*. DLR-German Space Operations Center. 2009. URL: <http://www.weblab.dlr.de/rbrt/OOS/EPOS/EPOS.html>. (Cit. on p. 4).

Selbstständigkeitserklärung

Hiermit versichere ich, Jens Reinecke, geboren am 13.07.1987 in Karl-Marx-Stadt, dass ich die vorliegende Diplomarbeit zum Thema

Joint impedance control of a stiffness adjustable tendon driven anthropomorphic hand

ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Bei der Auswahl und Auswertung des Materials sowie bei der Herstellung des Manuskripts habe ich Unterstützungsleistungen von folgenden Personen erhalten:

Maxime Chalon, Thomas Wimböck, Werner Friedl und Alexander Dietrich

Weitere Personen waren an der geistigen Herstellung der vorliegenden Diplomarbeit nicht beteiligt. Mit ist bekannt, dass die Nichteinhaltung dieser Erklärung zum nachträglichen Entzug des Diplomabschlusses (Masterabschlusses) führen kann.

Dresden, den 27.09.2011

.....
Unterschrift