NTNU
Norwegian University of
Science and Technology

Design and Analysis of a Hybrid Position/Force Controller for Robotic Rehabilitation of Upper Limb after Stroke

Kristine Blokkum

Master of Science in Cybernetics and Robotics
Submission date: June 2016
Supervisor: Øyvind Stavdahl, ITK

Norwegian University of Science and Technology
Department of Engineering Cybernetics

Abstract

There is an increasing number of stroke victims, leading to an increasing number of motor deficits as a result of stroke. Rehabilitation is an important factor in trying to improve the life quality of stroke victims. General physical therapy is a common approach, but the field of robotic rehabilitation is an expanding field of study.

The motivation for this thesis is to design a rehabilitation robot to enhance the rehabilitation offered to patients suffering from motor deficits after stroke.

The main objectives are to first present a literature review on the matter of end-effector based rehabilitation robots followed by a suggestion to an approach believed to be suitable. There are several robotic rehabilitation systems available. This thesis chooses and implements a control strategy based on the literature and suggest a functional design for verification of the control strategy.

The result of this work is a proposal and analysis of a hybrid position/force controller for robotic rehabilitation of an upper limb after stroke. The controller has been verified by simulations on a 2 degree-of-freedom robot manipulator. Two approaches to force control were tested, and a conclusion to which approach is best suited for this particular purpose is presented.

The thesis also discusses how to design a model and a controller for a robot manipulator. The model in this thesis is based on a 6 degree-of-freedom UR5 robot manipulator as well as a simple 2 degree-of-freedom planar robot manipulator.

Sammendrag

Antallet som hvert år blir rammet av slag er økende, noe som fører til at antallet med bevegelseshemninger som følge av slag øker. For å øke livskvaliteten til slagrammede pasienter er det vikig med rehabilitering. Robotrehabilitering er et fagfelt i stadig utvikling og det er flere robotrehabiliteringssystemer i utvikling i dag.

Motivasjonen bak denne avhandlingen er å designe en rehabiliteringsrobot for å forbedre rehabiliteringstilbudet til pasienter med bevegelsesvansker som følge av slag.

Hovedmålene med denne masteravhandlingen er å først presentere et litteraturstudie om end-effector baserte robotrehabiliteringssystemer, etterfulgt av et forslag til en egnet strategi. Det er flere robotrehabiliteringssystemer presentert i litteraturen. Denne avhandlingen velger og implementerer en reguleringsstrategi basert på litteraturstudiet og foreslår et funkjsonsdesign for å bekrefte strategien.

Resultatene i denne avhandlingen analyserer en hybrid posisjon/kraft regulator for robotrehabilitering av armer etter slag. Regulatoren er bekreftet av simuleringer av en robot med to frihetsgrader. Det er også testet to ulike strategier til kraftregulering med en konklusjon til hvilken som er vurdert som best egnet.

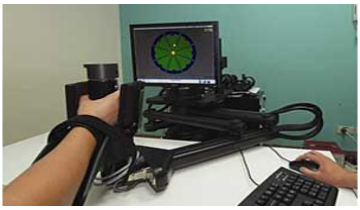
Denne avhandlingen diskuterer også en fremgangsmåte til utledning av en modell av en robotmanipulator. Modellen i denne avhandlingen baserer seg på en UR5 robot med 6 frihetsgrader, men en modell for en robotmanipulator med kun 2 frihetsgrader er også presentert.

PROJECT ASSIGNEMENT

Robotic rehabilitation of upper-limb after stroke

BACKGROUND

Approximately 15 000 people in Norway experience a stroke each year, and the incidence is believed to increase by up to 50 per cent in the next 20 years due to an ageing population. Stroke patients require extensive rehabilitation training to recover. Intense repetitions of coordinated motor activities in rehabilitation therapy and training constitute a significant burden for therapists, and due to economic reasons the duration of primary rehabilitation is getting shorter and shorter.



Recent systematic reviews of trials comparing conventional rehabilitation therapy by physiotherapists (CT) with robot-assisted therapy (RT), suggests that RT gives similar upper-limb rehabilitation in terms of motor function recovery, activities of daily living (ADL) and motor control. Also, extra sessions of RT in addition to regular CT have been shown to be more beneficial than regular CT alone in motor recovery of stroke patients [1]. Robot rehabilitation of upper-limb after stroke have thus gained increasing interest in recent years, and new robotic devices have been designed to assist in physical rehabilitation of stroke patients [2].

These robot rehabilitation devices may significantly improve on existing practice through increased effectivity in rehabilitation training with less man-power, closer monitoring of patient progress, higher quality and more objective therapeutic sessions, and more personalized follow-up for each individual patient.

TASK

A number of different control strategies have been employed for robot rehabilitation ranging from high-level strategies – such as assistive, challenge-based, haptic or couching strategies designed to provoke motor plasticity and thus improve motor recovery – to low-level strategies to control force, position, impedance and admittance factors of the high-level strategies [2].

Recent advancements in light-weight and affordable robot manipulators that complies with safety regulations can operate without safety guards between robots and humans, and have a potential for decreasing the cost of custom built robotic rehabilitation devices significantly.

The project shall propose a robot control strategy for rehabilitation of upper-limb after stroke using a standard industrial robot manipulator. The proposed robot manipulator is the UR5 robot from Universal Robots. The proposed strategy should be verified in simulation and experiments.

Task description

1. Present a literature review of end-effector based robotic rehabilitation devices and their control strategies, and propose a robot rehabilitation control system for a standard industrial robot manipulator
2. Set up a robotic rehabilitation simulation model of the UR5 robot manipulator from Universal Robots

3. Choose and implement a control strategy for robotic rehabilitation in the simulation model, and analyze the simulation performance
4. Suggest a functional design for experimental verification of the results
5. If time permits: Implement and analyze the robot rehabilitation control strategy in experiments

Useful resources may be the Robotics Toolbox and the V-REP robot simulator.

Objective and purpose
The objective of the project is to design a robot rehabilitation control system for a standard industrial robot manipulator for upper-limb stroke patients.

Subtasks and research questions
- Is robotic rehabilitation with standard robot manipulators feasible?
- What are the main challenges in robotic rehabilitation using standard robot manipulators?

Main supervisor: Associate professor Øyvind Stavdahl, ITK

Co-supervisor: Erik Kyrkjebø, Sogn og Fjordane University College

Bibliography

1. Norouzi-Gheidari, N., P.S. Archambault, and J. Fung, *Effects of robot-assisted therapy on stroke rehabilitation in upper limbs: Systematic review and meta-analysis of the literature.* J. of Rehab. R&D, 2012. **49**(4): p. 479-495.
2. Maciejasz, P., et al., *A survey on robotic devices for upper limb rehabilitation.* J Neuroeng Rehab, 2014. **11**(1).

Project Description

Norwegian University of Science and Technology - NTNU
Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Engineering Cybernetics

PROJECT ASSIGNMENT
Student name: Kristine Blokkum
Course: TTK4900 - Engineering Cybernetics, Master Project

Title: **Design and Analysis of a Hybrid Position/Force Controller for Robotic Rehabilitation of Upper Limb after Stroke**

Supervisor: Øyvind Stavdahl
Co-supervisor: Erik Kyrkjebø

Assignment delivered 13.06.2016
Trondheim, June 2016

Weakness and disability in upper limb is a common problem among stroke victims [10]. Several studies have been made, focusing on the effect of robotic rehabilitation after stroke. Two strategies have been developed: Exoskeleton robots, and end-effector robots. The main focus in this assignment has been the use of an end-effector robotic system for use in rehabilitation of upper limb.

Robotic rehabilitation has been a subject of research for many years, resulting in a large number of different systems being tested out. Therefore, the main objectives of this project has been to present a literature research and set up a model for a possible rehabilitation robot, and choose and implement a control strategy for testing. One objective has also been to suggest a functional design intended for testing, and if time permits preform experiments on a robot manipulator to verify the functional design.

Assignment consists of the following parts:

1. Present a literature review of end-effector based robotic rehabilitation devices and their control strategies, and propose a robot rehabilitation control system for a standard industrial robot manipulator

2. Set up a robotic rehabilitation simulation model of the UR5 robot manipulator from Universal Robots

3. Choose and implement a control strategy for robotic rehabilitation in the simulation model, and analyze the simulation performance

4. Suggest a functional design for experimental verification of the results
5. If time permits: Implement and analyze the robot rehabilitation control strategy in experiments

This report is a part of a pre project in creating a competitive grant application for a project on Rehabilitation of Upper-Limb after Stroke. The pre project is finished in 2016, and is led by Erik Kyrkjebø at Sogn og Fjordane University College (SFUC).

AUDIENCE
To benefit from this report the reader is advised to have some background in control engineering. There is however no requirement for the audience to have any medical background, or background in physical therapy.

Preface

First of all I would like to thank Erik Kyrkjebø and Øyvind Stavdahl for being great supervisors during this project. A big thank you to Kjetil Uhlen and Linne Shields for proofreading. And last but not least thank you to my fellow students at the office who always made me stay a little longer and work a little harder.

This master thesis is extending previous work done during the fall of 2015 in a project thesis presented by the same author. Some of the background theory on the use of robotic systems in stroke rehabilitation is also presented in the project thesis [2]. However, this thesis goes further in the controller design of the robotic rehabilitation system. The intention is that this thesis presents a full description of a possible rehabilitation robot. More information is presented in [2], but it will not act as a prerequisite for understanding this master thesis.

During the work of this thesis extensive time has been used trying to understand V-REP. Since the software was new to all parties involved a lot of time have gone by trying to get communication between V-REP and Matlab to work. Even though the controller was not successful for the UR5 robot in V-REP it is believed that a 2 link manipulator will provide enough valuable results to conclude on a suitable control strategy.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Stroke is a leading cause of neurological injury among the adult population [28], and every year approximately 15 000 people experience stroke in Norway [9]. Stroke rehabilitation is important for the quality of life for the stroke patients, and there is evidence that early stroke rehabilitation is beneficial for motor function [7]. The goal is that robot rehabilitation can be a useful tool in achieving the goal of providing necessary rehabilitation to all stroke patients.

Robotic rehabilitation can be a part of an early rehabilitation strategy. The idea is that one physical therapist can attend to more patients if there is access to a rehabilitation robot. This will lead to an expanded offer for patients in need of physical therapy. In [9] the authors describe the probable benefits of immediate rehabilitation after the acute phase of a stroke is over. That way, some of the chronic neurological deficits may be prevented.

Robotic rehabilitation can also be a part of the long term physical therapy after a stroke. In [9] they also present the importance of having long term rehabilitation stations spread around the districts, so the possibility of staying at home and still attend to physical therapy is available to the majority of the population. Also here, robotic rehabilitation can make a difference. Each physical therapist can attend to more patients, expanding the rehabilitation services.

There are several strategies developed in robotic rehabilitation of upper limb. This report will focus on end-effector based rehabilitation. An end-effector system will have a single attachment point to the patients arm, preferably close to the wrist, in order to provide motor training for a large part of the limb [17]. The end-effector system in this project is based on the UR5 robot from Universal Robots, see figure 1.1 (figure taken from http://www.roboticsbusinessreview.com). The UR5 robot is a robotic system with 6 degrees of freedom (from here on: DOF).

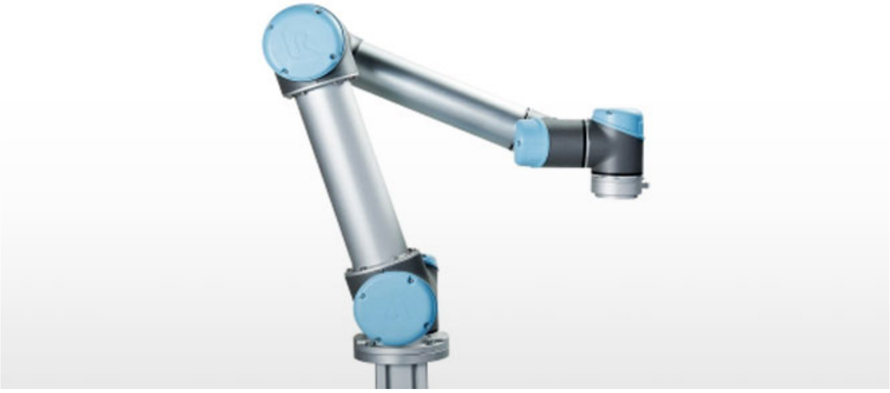


Figure 1.1: UR5 robot from Universal Robots

An important part of designing a rehabilitation robot for the upper limb is to decide which strategy of physical therapy to use. There are several strategies to this, and several opinions to which strategy gives the optimal effect of the rehabilitation. Each patient and therapist may have opinions to which approach is optimal, and it is important that the involved parties all believe in the particular approach. One possibility is that one single approach is not the optimal solution, but a combination of different approaches will have a better effect in increasing motor function.

After deciding on rehabilitation strategy, a control strategy needs to be designed. This was discussed in [2], and the work in this thesis is extending the results from [2].

The economic aspect of the project have not been well considered in this report. However, there have been made assumptions that an end-effector robot will be more economic than an exoskeleton one. This is due to the effectiveness of being able to use the same robot for more than one patient without having to make changes to software or hardware. Beyond that point the economic cost of the development have not been considered.

One of the goals of the rehabilitation robot has been that it should be effective for patients in different phases of the rehabilitation. In the early phase the robot should be a guiding device reestablishing movement in the affected arm. After some motor function in the limb has been restored, the intention is that the robot should act as training robot. This is the main reason behind the decision to use a combination of rehabilitation strategies. How the different strategies are designed was discussed in [2], and aspects necessary for the implemented approach of this paper will be discussed in chapter 5. The ultimate intention is to have a switch to decide the rehabilitation strategy which can easily be changed by a physical therapist or the patient himself. This however is outside the scope of this thesis.

There are several strategies to robotic rehabilitation [21]. Chapter 2 presents the different strategies found in the literature.

1.2 Thesis goals

The goals of this thesis are to establish a control strategy for the rehabilitation robot and verify the controller through simulations. The ultimate goal of the project led by Erik Kyrkejebø is to design a complete rehabilitation robot for rehabilitation use. This thesis has the goal to contribute to that project by establishing the controller design, come up with a functional design for further testing, and analyze the controller design through simulations.

This thesis will establish a suitable controller for the purpose of robot rehabilitation. It will provide examples of different approaches with testing through simulations of the strategy believed to be best suited.

The intention is also to create a model based controller of an UR5 robot to make the system more conventionally available and easy to set up by physical therapists. The belief is that this approach will create a good frame for what is optimal for patient recovery.

1.3 Contributions

In a large part of the literature already developed systems have been tested in controlled trials. This assignment has a different approach. The contributions of this report along with [2] are to map the systems already developed and then decide on a rehabilitation strategy believed to have the most improvement of motor function.

The work from [2] has the focus of mapping controller designs already in use. This thesis extends the work from [2] by presenting a functional design based on what is believed to be the most suited strategy based on the literature review.

This thesis provides good results from simulations based on the presented rehabilitation strategy. It is reasonable to believe that the robot traits will provide a good base for muscle training. The results are based on a multivariable model based hybrid position/force controller designed for a 2 link robot manipulator.

Based on the positive results from the 2 link robot manipulator simulations this thesis provides a model for the UR5 robot. Even though the simulation results are not optimal it is believed that the created model is a good base for future development of a controller for the UR5.

1.4 Report outline

In chapter 2 the background literature for this paper is presented. This includes rehabilitation strategies and some design features of different rehabilitation robots. Most of this chapter is a copy of results presented in the project thesis of fall 2015, written by the same author ([2]).

Chapter 3 gives insight to the main control strategies found in the literature review that are considered suitable for the purpose of this thesis.

In order to set up a controller for a robot manipulator there is a need for knowledge about the robot model. This information is found in chapter 4. This chapter also presents how to develop a model for a 6 DOF robot manipulator with revolute joints which is intended to use as the rehabilitation robot.

Chapter 5 presents the simulation scenario of this project. The first sections presents a functional design for the rehabilitation robot which is intended to simulate. The last part contains some description of the simulation software intended to use in this project.

The results of the simulation work are presented in chapter 6. In this section two different strategies for force control are tested. There are also simulation results from the hybrid controller for the 2 dof rehabilitation robot.

Chapter 7 and chapter 8 discusses the results from the previous chapters and concludes on the strategy believed to be best suited for a robotic device intended for stroke victims.

The last part of this paper is a description of work the author believes will benefit future work. This information is found in chapter 9.

In the appendix information used in simulations of this project can be found as well as information in order to complete the model derived in chapter 4.

Chapter 2

Theoretical Background

In stroke rehabilitation there have been evidence that early rehabilitation might reduce the number of patients with chronic motor deficits. In this section different rehabilitation strategies are presented. First the most common approach today which is general physical therapy, before the focus turns to robotic rehabilitation. Large parts of this chapter are directly taken from [2] with a few modifications.

2.1 Introduction to general physical therapy after stroke

In most institutions with focus on motor function rehabilitation the approach does not involve the use of rehabilitation robots. In [14] they looked at the two most common methods of conventional physical therapy after stroke. The Bobath method and Motor Relearning Programme (MRP).

Figure 2.1: Example of conventional physical therapy, picture taken from www.forskning.no

The Bobath method is based on experience and understanding of the nervous system. It consists of manual stimulation, ”guiding” exercises, and exercises to improve mobility.

Physical therapists practising the Bobath method try to avoid the use of tools to compensate for the movement the patient is trying to perform. They try to create exercises to increase the mobility and the ability to perform daily life activities by not only working on partly function of the limb.

Unlike for example the Bobath method, the MRP method is based on science in education, psychology and bio-mechanics. The treatment with MRP is based on functional targeted strategies, focusing on tasks that are going to improve the patient's ability to engage in the daily life. In MRP, the physical therapist acts as a consultant. He provides motivation and creates a program for the patient to follow. One focus with MRP is to work on regaining the independence for the patient, and therefore the training often consists of training aids. The method consists of four steps [3]. The first step will be to identify the problem, and the missing ability to interact with daily activities. The next step will be to train using specific exercises to improve the problem. The third step is training using functional components, and the last step is to transfer the skill learned by exercise into tasks for the daily life.

In [14] they performed a clinical trial of 61 patients and concluded that MRP has the best effect on patients past stroke. The patients treated with the MRP tended to have a reduced time of hospitalization. They also seemed to have greater improvement of motor function than the patients treated with Bobath's method.

2.2 Strategies for robot design

There are different phases of rehabilitation after a stroke. In an early phase the focus of the rehabilitation will be to regain sensation in the limb, reduce stiffness in the joints, and increase the ability to move the limb without support. In later stages, the goal of the physical therapy will be to regain muscle strength and retain the ability to perform daily tasks, like picking up a cup of coffee. These two different stages of the rehabilitation require different control, and a rehabilitation robot should include a function that lets it switch between the two rehabilitation strategies.

The first question to consider in robotic rehabilitation is the design of the robot. Depending on the application, the robot design might be very different, both in structure and size. Below is a presentation of some robot designs to consider.

2.2.1 Exoskeleton vs end-effector system

In robotic rehabilitation of upper limb, there are two very different strategies. One is end-effector robotic system, the other is an exoskeleton system. In an end-effector system, the patient is connected to the robot in a contact point at the end of the robot. The exoskeleton is a device connected to the patient at several points on the limb. It is a device that needs to be worn, and it has to be fitted to every user separately.

In [18] several upper limb rehabilitation robots are presented. Figure 2.2 shows a picture of the RUPERT exoskeleton rehabilitation robot. RUPERT is designed to support movements in the shoulder, elbow, forearm and wrist. It has 5 degrees of freedom, and the exercises the patients are set to do is based repetitive therapy tasks designed to help train the patient to do activities in the daily life [32].

Figure 2.2: RUPERT - Exoskeleton Robot, picture taken from [32]

Figure 2.3 shows the already commercially available InMotion ARM, which is an end-effector system designed to help rehabilitation of the shoulder and the elbow. It is designed by Interactive Motion Technologies, Inc. The InMotion ARM is a robot designed for several stages of the rehabilitation, and it can be used in several stages of the rehabilitation. It also has add-on modules for hand and wrist training.

From a control standpoint the two different approaches has major differences. The exoskeleton robot approach leaves the robot with much more control of the limb as the whole arm is connected to the device. The end-effector approach leaves the joints in the arm much more free to move. From a control point of view one can say that the end-effector robot leaves the system without the controllability and the observability of the limb, which the exoskeleton robot provides.

This project focuses on the end-effector system. It is easier to implement, and can be used by many patients without much personal calibration and modification of the robot. It will be easier for the physician to use, which as discussed earlier is considered a require-

Figure 2.3: InMotion ARM - End-effector robot, picture taken from [18]

ment for this purpose.

One can discuss that the exoskeleton system gives a more customized rehabilitation for each individual, but it requires more complicated systems and will be more complicated to use.

2.2.2 Home based or institutionally based system

The question on where the rehabilitation robot should operate is important. A robot placed in a gym or in the office of a physical therapist, from here on referred to as an institutionally based system, can be more complex and larger than a system placed at a patient's home, from here on called a home based system. Home based robots need to be more compact and more user friendly than one placed where the physical therapist is present. The ten patient trial in [23] looked at ten patients post stroke and their goal of recovery, to come up with a good strategy for home stationed systems.

A home stationed robot will need something to motivate the patient. If the patient is expected to exercise on his own, a robot equipped with a game to keep track of progress might be a good idea. That way the patient can keep track of progress, both in strength, stability and in range of motion. Another thing pointed out by the patients in [23] was that the robot needs to be so simple it can be maneuvered by the patient himself. The patients expressed they were feeling like a burden to friends and family, and saw the rehabilitation robot as a chance of gaining some independence.

In [15] they suggest that home-based systems supported by a team of experts increase the independence of patients. There is however evidence that such systems will result in improvements essentially on patients without severe disabilities. Severely impaired patients will most likely have a need for a longer stay in an institution.

One advantage of a robot stationed at a physical therapy office is the presence of experts on rehabilitation. A physical therapist has experience in rehabilitation of stroke patients and can keep track of progress. A physical therapist can also make sure the exercise is done the right way to ensure optimal rehabilitation. A home stationed rehabilitation robot might compensate for that by having some sort of communication platform between the patient and the physical therapist. One solution is to have the possibility to send text or video messages. Another possibility for communication might be a live video chat. A video chat is probably the best solution for the patient, but it requires that the physical therapist sets a time to video chat with all his patients. If the physical therapist has to set a time to talk to all his patients, having the patients come to him might be a better idea. That however depends on the patient's ability to travel between home and the physical therapist.

In developing a rehabilitation robot it is important to remember that a robot can never take the place of a physician. The rehabilitation robot can only act as a tool to help optimize the training. It can also help the physician to treat more patients. Personal contact is however a very important part of the rehabilitation, and personalized exercises need to be decided by a physician. The robot can only be a tool to help the patient with exercises given by the physician.

2.3 Rehabilitation strategies

Rehabilitation strategies can be divided into two main categories. Active physical therapy and passive physical therapy. In order to meet the desired stages of the rehabilitation, the robotic system needs some sort of switch in order to change between different rehabilitation strategies. One control strategy might be to guide the limb through a predefined path, where another is designed to give resistance to a predefined path. There are several strategies for robotic rehabilitation within the two main categories. [21] and [18] presents four different control strategies: Assistive control, challenge-based control, haptic simulations and non-contact coaching.

2.3.1 Active or Passive robotic therapy

For the two main categories of rehabilitation one is described as passive training, the other as active training [35]. In passive training, the patient is not required to apply force in any way. The patient is expected to let the robot move the limb. The thought is that passive training is suppose to reduce plasticity in the impaired limb and in that way give the arm a better basis for the start of active therapy.

The scientists who support the passive robotic control therapy suggest that by leading a limb though a path, the patient will learn to recognise the path and follow the path without help. The thought is that if a robot can induce a motion in the limb the patient is unable to perform independently, the brain might recognise the movement. When the brain recognises the movement the motion might be easier to perform in the future [21]. In [25] they value the idea of passive training. They support their hypothesis by a throwing experiment both in a regular gravity space and in a space station with a micro gravity environment. The study showed no indication that the performance of the subjects in the gravitational field they describe as the assistive training approach was impaired compared to the control group.

There is not a unanimously vote that passive therapy is the best approach for neurological injuries. Some claim that the guidance through a path makes the patient less prepared for the activities the assist is simulating [19]. Their assertion is that timing errors could increase with patients being treated with passive training.

In [19] Marchal-Crespo and Reinkensmeyer performed a trial in which they claim to prove that the effect of active therapy is the preferred strategy. The study claims that the passive training will make the patient's ability to understand the movement weaker. They propose active therapy as the best approach for motor learning.

Active therapy requires more participation from the patient. The patient has to actively be a part of the movement by using the limb to apply force for training. Active training can increase the patient's ability to apply force and the goal is that the muscle strength of the patient is supposed to be increasing with active therapy.

One goal for the robot in this project is the ability to adapt to several patients in several states of the rehabilitation. A switch that can be manipulated to change between assistive control and challenge-based control would be helpful in a rehabilitation perspective. If the robot also has the ability to change the assistance or the degree of challenge based on real time measurements from the patient's abilities to apply force, the need of a physical therapist present at all times would be unnecessary and the patients would end up more independent.

2.3.2 Assistive Rehabilitation

The first strategy to robotic therapy is assistive therapy [21]. Assistive therapy is based on the principle that the robot is assisting the patient in performing a task. Some studies shows that this is helpful in getting movement back in the limb. The robot will then guide the arm through a trajectory with an assistance allowing the movement to be more natural and also possibly with a higher speed. That way the patient can have a more intense training and a more rapid improvement.

Assistive therapy might also help with motivation. If the patient cannot move the arm at all, assistive therapy might be more motivating than trying to accomplish a task that seems impossible. The assistive therapy can also act to help the patient stay inside a desired path. If the patient moves the limb outside the predefined path, the robot can apply a force to get the limb back on track.

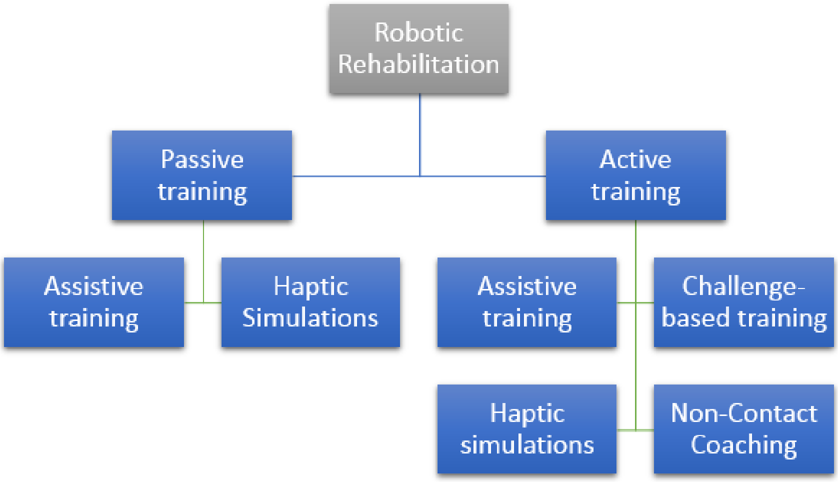


Figure 2.4: Robotic rehabilitation strategies

One problem with assistive therapy is that there is a possibility of misuse. If it is used wrong, the patient might rely too much on help from the robotic device to get any training. The movement might not require the patient to pay attention to the training, causing it to be just a robot lifting the impaired limb.

One solution presented in [21] that might reduce the problems of assistive therapy is triggered assistance. Triggered assistance is based on the principle of assistive therapy, but the assistance is not triggered until a certain variable is reached. The variable might be time, force, joint velocity or muscle activity. If the patient has weak muscle activity in the arm, the triggering can be based on electromyography (EMG) signals. That way, the patient is required to make an effort, but there does not have to be any measurable force on the robot device. This however creates the problem that the patient might give the robot enough to trigger the assistance, and then relying on the robot to do all the work.

Triggered assistance can also be used if the patient is able to move faster than a predefined speed, or apply more force than initially intended. Then the robot can act as resistance, making it a muscle workout device.

One example of triggered assistance therapy is as mentioned EMG-triggered assistance [21]. In EMG-triggered assistance, EMG signals are measured on the skin and when the signals increases above a threshold the assistance is triggered. The assistive triggering might also be designed to move in a direction based on which muscles are activated. The EMG signals measures the amount of work the muscle is doing, and the robot can be triggered even though the muscles are not strong enough to move the arm on its own. There is however a chance that the patient is unable to control the muscle activity well enough to control the direction of the robot, causing the robot to move in a direction not intended by the patient. This might be an issue regarding motivation and safety.

In [4] an assistive control system is presented. The idea is using a predefined path, and a score according to how well the patient is able to follow this path. If the patient is unable to complete the movement, the robot will guide the patient through the path without the patient's score increasing. This way, the patient and the physician can easily follow the progress, and the motivation to increase the score from time to time might help the patient to focus.

Another example of assistive control strategy for rehabilitation is counterbalancing assistance [21]. Counterbalancing assistance involves having the ability to reduce or increase the support to the limb. In that way, the robot can zero out the weight of the arm, or it can completely guide the limb training.

2.3.3 Challenge-Based Rehabilitation

Challenge-based control is in a way the opposite of assistive control, and is also presented in [21]. One can say that an assistive controller makes the movement easier for the patient, while challenge-based controllers act as a resistance. This strategy can be applied later in the rehabilitation, when the patient no longer needs assistive therapy.

There are few arguments against the challenge-based control strategy. There is an agreement that training the limb with a challenge-based activating strategy will increase the motor function of the limb [21]. Physical therapy using weights is a form of challenge-based training. The robot in this case needs to act as a weight for the weak arm to move.

There are a few different challenge-based strategies discussed in [21]. One is the resistive strategy. This strategy focuses on giving resistance to the limb that the patient is trying to move. Even not canceling the gravitational force can be considered a resistance. As can providing a counteracting force to resemble lifting imaginary objects.

In [30] they report good outcomes after a trial with constraint-induced treatment. This is a challenge-based strategy where the focus is that the patient will not depend too much on the healthy limb. Constraint-induced therapy makes sure that the impaired limb is used to perform the task. This is a strategy in focus for training sessions intended for several limbs, not just the affected one. It can be used for exercises like turning a wheel, and walking for the patients whose legs are affected from a neurological injury. This is not a focus area for the 6 degree-of-freedom robot in this project, which is intended to train only the weak upper limb. It is however a control strategy to have in mind for other training robots.

A third approach to challenge-based control is what they in [21] call the error-amplification therapy. This approach is the opposite of the assistive control trying to cancel the patient's error. This approach increases errors made by the patient not being able to stay inside the predefined path. If a patient makes a motion pushing the limb outside the path, the robot will increase the error forcing the patient to work to get the limb back on the trajectory. This will according to studies increase the patient's desire to work to stay in the task provided by the robot. This is also a strategy to combine with a visual screen showing the patient which path to follow.

2.3.4 Haptic Simulations

Haptic simulations are a tool to motivate the patient by relating the workout to real life situations. This can for example be a video screen showing the lifting of a coffee cup while the patient is doing training. The haptic control will then guide the patient through a task shown on a screen [24]. Haptic simulations can be a form of both active and passive training. In [20] a system which sets up a passive training situation is presented. They conclude that haptic simulations have a positive effect on passive training. As motivation is an important factor for all training, haptic simulations can simulate real life situations giving the patient a more visual goal to work for.

Figure 2.5: Example of the possibilities of haptic simulations, picture taken from www.instantreality.org

As technology is progressing the possibilities for haptic simulations increase. At the time of writing the most advanced scenario would be the use of virtual reality glasses combined with training activities of daily life.

2.3.5 Non-Contact Coaching

The non-contact coaching strategy is not a rehabilitation strategy like the assistive or challenge-based control. Non-contact coaching robots are robots designed to motivate and guide the patient through the rehabilitation, which is usually the task of the physical therapist.

Non-contact coaching is not considered in this thesis. There is a possibility of using screens for communication if necessary. It is important to remember the need of human contact. In [23] one person highlights the importance of human interaction to keep the motivation up. The intention of this project is not to replace physical therapists, but instead to create a tool to make the work of the physical therapist easier and more effective for the patient.

2.4 Defining rehabilitation paths

An important part of robot rehabilitation is deciding different trajectories [21]. There are several ways to do this, and the most common way is to create the trajectories based on activities of daily life. This can be done in several ways.

Mathematical models of trajectories are one way of generating trajectories. In [34] they studied the movements of eight people without motor deficits while they performed 24 different daily life activities. The thought is that from analyzing pattern from the performance of activities it is possible to develop a mathematical model to the trajectory.

Another approach to create trajectories is to base the trajectory on direct movement from people without motor deficits. The authors in [1] present an example of this with walking on a treadmill. In a rehabilitation of the upper limb where only one of two limb are impaired, an example of this approach could be mirroring of movements of the healthy limb.

In [12] the authors present another approach to the development of trajectories. They point out the fact that one cannot take for granted that one limb remains without motor deficits after a stroke. In such a case, the mirroring of the healthy limb is not ideal. They focus on tasks that are pre-recorded and saved in a database. With access to the database, the therapist designs a training program for the patient to follow.

2.5 Safety

The most important part of any rehabilitation robot is to make sure the safety of the patient is cared for. The system will need to have predefined safety features that ensure safe handling in case of a power outage or a system failure of any kind. The safety features in this report are based on the UR5 from Universal Robot, as that is defined in the project assignment.

In case of a power outage it is important to make sure that the robot stays in the position it had before the loss of power. If the robot arm were to drop, it would most likely bring the patient's arm in a rapid movement which could lead to injury. If the robot arm just froze in position when the power outage happened the patient would most likely be able to disconnect from the robot without injury until the power is back and the training can restart.

There should also be an emergency brake in close reach of the patient and the physical therapist. If the patient in an early stage feels uncomfortable with the movement of the robot, the patient should be able to abort the program and stop the robot.

Another important safety aspect of the robot would be to secure the patients from electrical charges. For the use of this robot the chance of microshock will probably not be a problem. Microshock requires electrodes placed inside the body, on the heart or close to the heart. There is a low chance that a patient will still be in a condition to where he could experience microshock at the time this robot will be in use.

The real hazard of this device will be a macroshock. However, on the website of Universal Robots they present a scenario where the UR5 robot intended to use in this project is used in a gyno-urological medical use. This indicates that there is a possibility for securing the robot enough to use it in a rehabilitation situation.

The user manual for the UR5 robot sets out a warning that with the use of the robot it is important to provide secure and proper grounding. This is a factor assumed to be taken care of in an institution hosting a rehabilitation robot. There is no more research on the matter in this thesis, but it is an important aspect to consider before the robot is taken into use.

2.6 Patient performance evaluation

In order for the patients to monitor their own improvements some sort of evaluation is necessary. In [4] a score based on performance evaluation is tested out. They divided each path into ten segments, and the patients were given a score according to their ability to follow each segment without the interaction of the robot. If the patient was unable to complete the segment the robot would interact, and the score did not increase. In this approach the patient can continuously monitor his own recovery and the belief is that the system also works as a motivation [4].

All the above situations is thought to have some screen showing the patient the path and the performance while the patient is using the rehabilitation robot. This will not be included in the simulations in this thesis. The thought is that there will be a screen showing the patient the progress of the rehabilitation. It will be a part of future work, and will not be discussed further in this thesis.

Chapter 3

Control Strategies

The ultimate goal of this thesis is to come up with a control strategy that can be utilized by patients at several stages of the rehabilitation. Therefore, there is a wish to come up with a control strategy that can be modified to suit both active and passive rehabilitation.

Because the patient is applying a force to the end-effector a controller based on only position is not considered by the author to be well suited. The controller will need to take into account the force the patient is applying to the manipulator, and will need to provide a counteracting force to the patient to provide exercise. There will be proposed two different ways to achieve this: Impedance control or hybrid position/force control.

Even though some scientists focus on the difference between admittance and impedance control, several articles uses the terms interchangeably. Some even use the term impedance control on control systems others refer to as admittance control. In this paper the term impedance control is the term used to describe these controllers, even if some articles refers to the controller as an admittance controller.

Recent controllers for robotic rehabilitation devices have focused more on the control of the dynamics of the system, when the patient interacts with the robot. The benefits of the dynamic controllers is that they are easier to stabilize, and in order to have a real-time control of the force of the robot a dynamic solution would be preferable. Impedance and hybrid controllers are examples of controllers which control the dynamics of the patient and robot.

3.1 Impedance Control

Impedance control is a way to control the dynamic relationship between the environment and the end-effector. It is a way to control both position and force, which is beneficial for the situation where external forces affects the robot. The impedance controller is usually modeled as a mass-spring-damper environment. This is to show how the dynamics of the controller works [6].

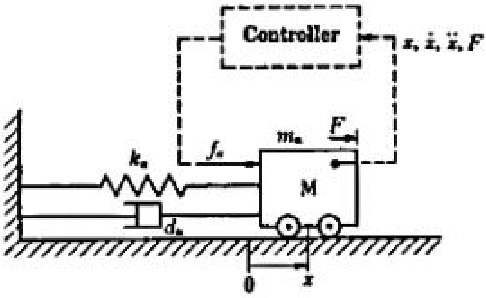


Figure 3.1: Mass-Spring-Damper Impedance Controller. Figure taken from [6].

The mass-spring-damper system in figure 3.1 is an example of a one degree-of-freedom impedance controller. The equation of motion for the system can be expressed as:

$$M\ddot{x} + d\dot{x} + kx = u + F \tag{3.1}$$

Where M is the mass of the cart, d is the damping coefficient, k is the spring coefficient, u is the control input and F is the Force of the cart.

Bu using $u = u(x, \dot{x}, \ddot{x}, F)$ one can control the impedance of the system. If the wish is to cancel the damping effect and reduce the mass an example of a input can be:

$$u = 2F + d\dot{x} - 2kx \tag{3.2}$$

The closed-loop system ends up being:

$$\frac{1}{3}M\ddot{x} + kx = F \tag{3.3}$$

Looking at the closed loop system one can see that the damping is eliminated and the mass is only a third of what is was originally. The controller changed the mechanical impedance of the system to fit the wanted system. One can see that by changing u, the design of the closed loop system can be modified to fit the system one want to create.

Some literature focus on different ways to control the dynamic relationship between the manipulator and the environment, that is impedance control and admittance control.

Impedance and admittance control systems are inverse systems. While impedance control is based on resistance, admittance control is based on conductivity [21], [4]. In a dynamic relationship, if one part of the relationship is seen as an impedance, the other has to be an admittance [8].

In [8] the author defines the difference between the two systems. They present impedance control as a system where the equations will have the form of motion in and force out. In admittance control however, it will be the opposite. With an admittance controller, the model will have a form of force in and motion out.

This means that in an impedance controller any movement outside the desired trajectory will be considered an impedance, or an error. Then conventional robot controllers can be applied to the system with the desire of reducing the impedance by applying a force in order to stay in the desired path. Admittance control on the other hand runs on force from the patient. The patients applies a force on the device which in turn produces movement in the robot.

In both impedance and admittance controllers conventional controller strategies can be used, for example PID-controllers. In [36] and [29] they present PID controllers for the impedance and admittance controllers respectively. In [29] the controller is an adaptive PID controller. The reason they claim that an adaptive PID controller is preferred over a regular one is that a regular one would not consider the fact that the dynamic stiffness in the model will be variable.

3.2 Hybrid Position/Force Control

Hybrid position/force controllers were first presented by Raibert and Craig in 1981 [27]. The controller is based on the fact that the manipulator and the external force is not coupled together like they are in the impedance controller. It also uses the fact that every task the manipulator is set to preform can be defined with a set of constraints. The constraints can both be natural and artificial. The constraints apply to both position and force. Natural constraints can be the fact that a manipulator can not move through a surface, so there is a position constraint keeping the manipulator from exiting the workspace. An example of a force constraint may be that the manipulator can not apply an arbitrary force tangent to a frictionless surface.

Both examples presented above are examples of natural constraints. The hybrid controller can also be defined with artificial constraints. These are specified to give the controller a desired behavior.

The hybrid position/force controller proposed in [27] is not a control law. It is merely an architecture designed where different control laws might be applied. Figure 3.2 shows a block diagram of the architecture of the hybrid controller. The idea is to measure both force exerted by the environment and position of the joints and apply two different control laws and merge the system together into one hybrid controller. The postion controller will not be responsible for controlling any of the forces applied to the robot manipulator, it will only make sure the robot does not leave the desired position. The force controller will be responsible for applying the desired force to the end-effector.

Controlling force and postion will most likely not have the same transfer function. Therefore the possibility of using the exact same control law will most likely not be optimal. There is however likely that the two approaches can be based on be the same control law with modifications to make it suitable for the exact purpose.

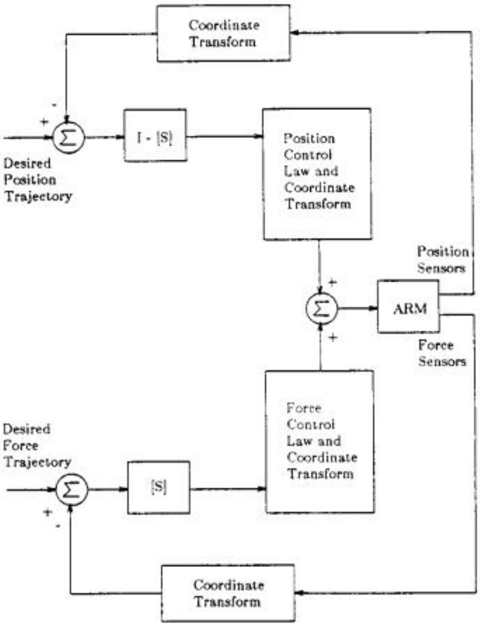


Figure 3.2: Hybrid position/force controller as described in [27]. Figure taken from [33].

The hybrid position/force controller in [27] presents a control scheme where the artificial constraints in position is normal to the surface of the robot motion, while the artificial force constraints is along the tangents of the motion surface. The coordinate system for which the constraints are developed is based the base frame of the robot, not the coordinate system of the robot end-effector. This will be further discussed later in section 5.2.2.

In [27], [33] and [6] a simple example is used to describe the hybrid position/force controller. The controller is explained through a "peg in the hole" problem. The problem is illustrated in figure 3.3. The concept is that the peg is to be inserted into the hole by a controller. Once the peg is partly in the hole the movement in the x and y direction stops. There is a natural constraint keeping the peg from moving in neither of these two directions. There are also constraints keeping the peg from rotating around the x and y axis, which implies that there can be no moment in neither of these directions. These are the natural constraints of the system.

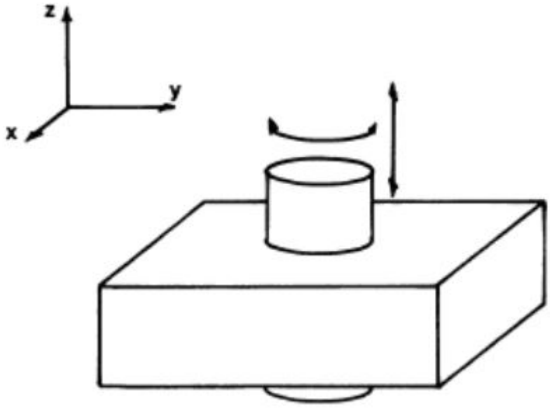


Figure 3.3: Illustration of the "peg in the hole" example. Figure from [6].

The arbitrary constraints can be found by saying that there should be no moment around the z-axis, which is not an absolute necessity but nevertheless a possible arbitrary constraint.

The idea is to create a force selection matrix, which in the case of the peg in the hole would end up being:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.4}$$

Where the force vector is $F = \begin{bmatrix} F_x \ F_y \ F_z \ M_x \ M_y \ M_z \end{bmatrix}$.

If F is the force and moment vector represented at the end-effector at a given time, and τ is the joint torque vector the following relationship can be used to convert from forces to torques, given that Jg is the geometric jacobian matrix of the robot manipulator (derived in section 4.1.2):

$$F^T * J_g = \tau^T \tag{3.5}$$

The part responsible for position control uses the position selection matrix, $P = I - S$, to calculate the position errors in the constraint frame. This eliminates position control in the direction where the force control is active. After the error has been calculated Raibert and Craig multiplies the error with a stiffness matrix to achieve the desired stiffness in the position controller. To control the different joints of the manipulator [27] uses the inverse jacobian matrix is used to find the joint errors, and a control law can be designed to control each joint.

After both the the position controller and the force controller of the system has calculated the desired torques they are summed and sent to the robot manipulator, as shown in figure 3.2.

In [27], Raibert and Craig does not introduce the manipulator dynamics in their control scheme. This however is presented in [37]. Yoshikawa claims that it is not possible to prove that the hybrid controller presented by Raibert and Craig in [27] can be used to control force and position independently. He presents a dynamic hybrid position/force controller in [37].

3.2.1 Weder and Saridis simplification of the Hybrid controller

Weder and Saridis proposed a simplification to Raibert and Craigs hybrid position/force controller in 1988. In [33] they present their findings. They claim that by simplifying the algorithm the algorithm will require less calculations, and will be easier to include into an real-time application.

For a rehabilitation purpose the position of the end-effector is important, but there is not a requirement that the position is 100 % correct at all times. Regarding that the patient should be able to control the position, and the idea of allowing human errors makes in unnecessary to introduce an error free position controller. It is considered more important for the rehabilitation purpose to introduce the correct force.

With allowing some position errors, one can eliminate the selection matrix. The other advantage of this method is that there is no longer necessary to preform the transformations to get the position of the end-effector in the constraint frame of the selection matrix. This reduces the amount of calculations necessary, and the algorithm will be more efficient.

As for the simplifications in the force controller, one can define the desired forces and moments in the frame of the measured forces and moments. This eliminates the need of a transformation of the force reference. For the sake of a rehabilitation robot, this is possible and an efficient way of defining the force reference.

The block diagram of the modified hybrid controller can be seen in figure 3.4. The Figure is taken from [33]. For the position controller Weder and Saridis has a simple P-controller. The force controller is a PD-controller.

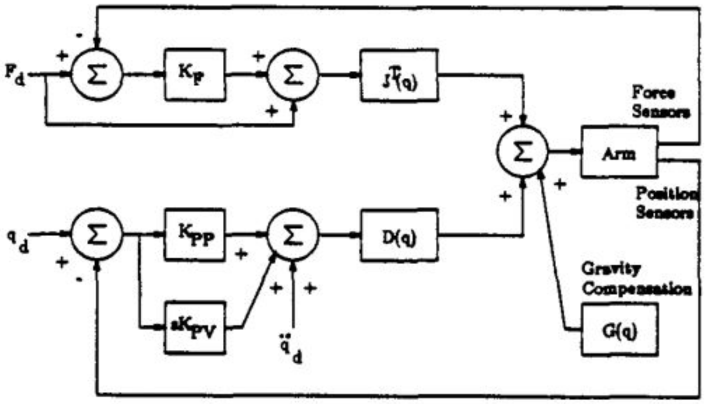


Figure 3.4: Hybrid controller from [33]. Figure from [33].

The controller is easy to implement, and it is based on well known control laws. This is from the authors standpoint an advantage, which is discussed later in this paper. For both the position and force control there is a possibility of introducing more complicated control algorithms. The control laws for this thesis will be discussed in section 5.2.1.

$$\tau = \tau_F + \tau_P + G \tag{3.6}$$

$$\tau_P = D[\,\ddot{q}_D + K_{PP}(q_d - q) + K_{PV}(\dot{q}_d - \dot{q})] \tag{3.7}$$

$$\tau_F = J^T[\,F_D + K_F(F_D - F)] \tag{3.8}$$

where τ is the total torque working on the manipulator. τ_F is the torque from the force controller, τ_P is the torque from the position controller and G is the gravity working on the robot. D is the inertia matrix discussed in chapter 4, the K matrices are the controller gain matrices, F is the measured force, F_D is the desired force, q is the joint angle, and q_d is the desired joint angle.

Equation 3.6-3.7 shows the equations of the hybrid controller proposed in [33]. Equation 3.6 shows the total torque applied to the robot manipulator. The torque of the position controller is showed in equation 3.7, while equation 3.8 showed the torque resulting from the force controller.

One argument in [33] is that the amount of calculations needed for the Raibert and Craig hybrid controller is not feasible for a real-time application. This is most likely not as issue at the present time. Computer technology has come a long way since 1988, and even though this has not been tested, the author assumes that the capacity of the computers today will be able to work the hybrid controller in real-time. However, one argument for introducing the simplifications is that all requirements are met for simplifying the controller as discussed in [33]. This leads to believe that the controller will not end up with reduced stability or performance by this applications.

3.3 Hybrid Controller in a Rehabilitation Robot

Above two controller designs are presented, which both have been implemented on rehabilitation robots ([16], [36], [5]). Both of these approaches can be applied both to active and passive rehabilitation. This is advantageous for making a rehabilitation robot that is suitable for patients in different stages of the rehabilitation.

In this project the focus will be the use of a hybrid position/force controller, described in section 3.2. This is because with an impedance controller one can only control the impedance of the system. With the hybrid position/force controller there is a possibility to control the forces exerted by the environment [6]. This is what will give the rehabilitation robot the chance to customize the rehabilitation. The structure of the hybrid controller is shown in figure 3.2.

Since the coordinate system for the force reference is in Cartesian coordinates of the base frame and the position reference is in the same base frame there is no need for a selection matrix to make sure the two controllers work in opposite directions. This makes the design of the control law a little different that what Raibert and Craig proposed but the controller is based on the same principle. The controller in this thesis will resemble the simplified controller from [33]. The controller will be presented in the next chapter.

Chapter 4

UR5 Robot manipulator modeling

One of the main goals of this thesis is to develop a controller for a rehabilitation robot. In [31] there are two different approaches presented in designing a controller for a robot maipulator. The first one is point-to-point control, or independent joint control, which is the simplest controller design. The other approach is multivariable control.

In a point-to-point controller for a robot manipulator one have to assume that the coupling effects from other joints in the manipulator is a disturbance. This reduces the need of complex modeling and controller design.

In multivariable control the control input is based on equations of motion derived from the dynamic relationship in the robot. The dynamic equations describes a complex, nonlinear and multivariable system. In this thesis the attempt will be to create a multivariable controller. This is to assure that the closed loop system is stable and that the robot is able to follow a trajectory.

This chapter presents robot manipulator modeling and multivariable control. Most of this chapter is inspired by [31]. It is an attempt to combine the most important aspects of robot manipulator modeling and complex robot control. In [31] the approach is described in more depth but it is to the authors belief that this chapter provides enough information to get familiar with robot manipulator modeling and control.

The derivation of robot kinematics and dynamics will in this chapter be presented. In order to create a controller based on the dynamic model of a robot it is important to know how each link of a robot manipulator affects the joints of the robot. This will be derived throughout this chapter.

The intention for this thesis is to develop a controller design based on the UR5 robot manipulator with 6 rotational joints $\boldsymbol{q} = [q_1, ..., q_n]^T$. For the conclusive simulations in this thesis a model for a robot with 2 planar rotational joints is used. The model derivations will be similar, but this chapter presents the model for the 6 DOF UR5 robot. Information to complete the model for the UR5 and the 2 DOF robot manipulator is included in the appendix.

4.1 Kinematics

When deriving a robot manipulator controller it is important to have knowledge about the motions of the robot. When describing the kinematics of a system one describes the motion of the robot without looking at the causes of motion. With analyzing the robot kinematics the torques and forces making the robot manipulator move is not considered, only the motion itself [31].

There are two ways to describe the kinematics of a robot manipulator. One is forward kinematics, the other is inverse kinematics. Forward kinematics is used to find the end-effector position and orientation of a robot manipulator given the joint angles. Inverse kinematics is used to calculate the joint angles based on the end-effector position and orientation.

4.1.1 Denavit-Hartenberg

The Denavit-Hartenberg (from here on DH) convention is used for frame selecting in robotic applications [31]. The DH convention is based on finding a homogeneous transformation for each link in the robot manipulator. These homogeneous transformation matrices can easily be multiplied together to find the transformation matrix for the combined robot manipulator. The transformation matrix can give the forward kinematics of the robot.

Each link in the robot manipulator is assigned to a coordinate frame. The four DH parameters θ_i (angle of rotation around z_{i-1}), d_i (distance from the origin o_{i-1} to the interaction with x_i measured along z_{i-1}), a_i (distance between z_{i-1} and z_i measured along x_i), and α_i (rotation around x_i) are used to transform the reference frame from $i-1$ into the reference frame of i. This is done by the homogeneous transformation A_i, shown in equation 4.1. When the transformation matrix has been defined it is easy to derive the forward kinematics of the robot manipulator.

$$\begin{aligned} \boldsymbol{A}_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \\ &= \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{4.1}$$

After each homogeneous transformation matrix is found, they can be multiplied together to create the transformation matrix T_n^0, equation 4.2.

$$\boldsymbol{T}_n^0 = \boldsymbol{A}_1 \dots \boldsymbol{A}_n \tag{4.2}$$

$$\mathbf{T}_n^0(\mathbf{q}) = \begin{bmatrix} \mathbf{R}_n^0(\mathbf{q}) & \mathbf{o}_n^0(\mathbf{q}) \\ \mathbf{0} & 1 \end{bmatrix} \tag{4.3}$$

The UR5 robot has 6 degrees of freedom, but the transformation matrix T_6^0 from the first link attached to the floor to the end-effector will be a 4x4 matrix on the form as in equation 4.3. $\boldsymbol{R}_n^0$ is the rotation matrix which gives the orientation of the end-effector, and $\boldsymbol{o}_n^0$ is the translation vector which gives the end-effector position in Cartesian coordinates.

The coordinate frame for the UR5 robot is shown in figure 4.1. From this the DH parameters can be found. The UR5 is a well known robot manipulator, so the DH parameters have been presented in several pieces of the literature ([26], [13]). The DH parameters for the UR5 is shown in table 9.1 and table 9.2.

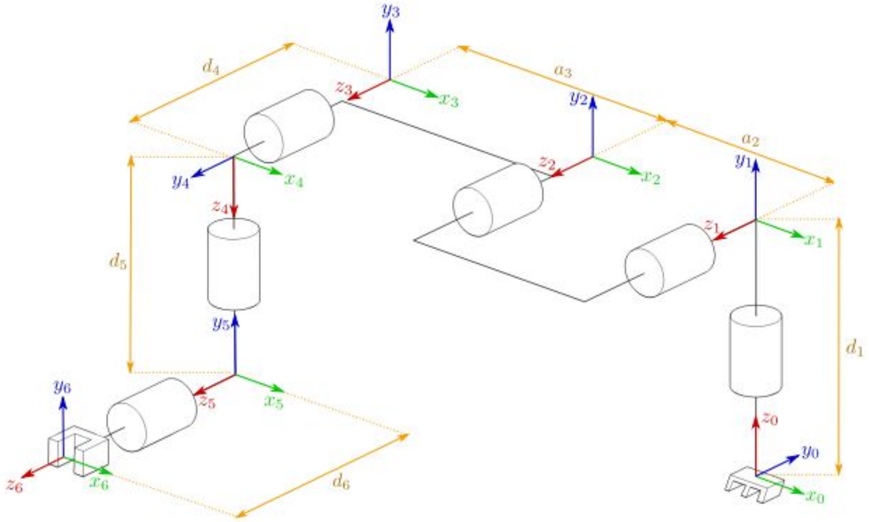


Figure 4.1: Denavit-Hartenberg coordinate frame. Figure taken from [26]

The derivation of the forward kinematics is important if the robot manipulator is only able to measure the joint position. From the transformation matrix it is easy to find the end-effector position if the joint positions are known, as it is given from the translation vector (equation 4.3).

Link	θ_i	d_i [m]	a_i [m]	α_i [rad]
1	θ_1^*	d_1	0	$\frac{\pi}{2}$
2	θ_2^*	0	$-a_2$	0
3	θ_3^*	0	$-a_3$	0
4	θ_4^*	d_4	0	$\frac{\pi}{2}$
5	θ_5^*	d_5	0	-$\frac{\pi}{2}$
6	θ_6^*	d_6	0	0

Table 4.1: Denavit-Hartenberg parameters for the UR5

Link	θ_i	d_i [m]	a_i [m]	α_i [rad]
1	q_1	0.089	0	$\frac{\pi}{2}$
2	q_2	0	-0.425	0
3	q_3	0	-0.392	0
4	q_4	0.109	0	$\frac{\pi}{2}$
5	q_5	0.095	0	$-\frac{\pi}{2}$
6	q_6	0.082	0	0

Table 4.2: Denavit-Hartenberg parameter values for the UR5. Values taken from [22]

4.1.2 Velocity kinematics

To create a model for a robot manipulator it is necessary to know about the angular and linear velocities of the robot manipulator. The relationships between the angular and linear velocities at the end-effector and in each robot joint is presented through the Jacobian matrix.

The Jacobian matrix is a useful tool in robot control. It can be used for areas like planning and execution of smooth trajectories, determination of singular configurations, in the execution of coordinated anthropomorphic motion, the derivation of the dynamic equations of the robot motions, and in the transformations of force from the end-effector to the manipulator joints [31].

It is possible to split the Jacobian matrix to consider linear and angular velocity separately. If $\boldsymbol{v}_n^0$ is the linear velocity and $\boldsymbol{\omega}_n^0$ is the angular velocity the relationship can be written like this:

$$\boldsymbol{v}_n^0 = \boldsymbol{J_v}\dot{\boldsymbol{q}} \tag{4.4}$$

$$\boldsymbol{\omega}_n^0 = \boldsymbol{J_\omega}\dot{\boldsymbol{q}} \tag{4.5}$$

where $\boldsymbol{q}$ is the joint position. The full manipulator Jacobian matrix is then:

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J_v} \\ \boldsymbol{J_\omega} \end{bmatrix} \tag{4.6}$$

There are two different representations of the manipulator velocity kinematics. One is the geometric Jacobian, the other is the analytic Jacobian. For the purpose of this thesis there will be introduced a third Jacobian matrix, which is the Jacobian for an arbitrary point on a manipulator link.

The following sections explains the derivations of the Jacobian matrices. The approach is further explained in [31].

Geometric Jacobian

The Geometric Jacobian is used to relate the end-effector forces and moments to joint torques, which will be done in the force controller. It is derived using the homogeneous transformation matrices $\boldsymbol{A_1}, ..., \boldsymbol{A_n}$.

For the linear velocity the following equation is used to derive the Jacobian matrix:

$$\boldsymbol{J}_{g,\boldsymbol{v}_i} = \frac{\partial \boldsymbol{o}_n^0}{\partial q_i} = \begin{cases} z_{i-1} \times (o_n - o_{i-1}) & \text{for revolute joints} \\ z_{i-1} & \text{for prismatic joints} \end{cases} \tag{4.7}$$

where $\boldsymbol{o}_n^0$ is the translation vector from the base to end-effector link. The angular velocity Jacobian is derived by the following equation:

$$\boldsymbol{J_{g,\omega_i}} = [\boldsymbol{z}_0^0, ..., \boldsymbol{z}_{n-1}^0] = \begin{cases} z_{i-1} & \text{for revolute joints} \\ 0 & \text{for prismatic joints} \end{cases} \tag{4.8}$$

$$\boldsymbol{z}_{i-1}^0 = \boldsymbol{R}_{i-1}^0 \boldsymbol{k} \tag{4.9}$$

Where $\boldsymbol{R_{i-1}^0}$ is the rotation matrix from the base to link $i-1$ and $\boldsymbol{k}$ is the z unit vector $\boldsymbol{k} = [0\ 0\ 1]$.

The total geometric Jacobian is given by:

$$J_g = \begin{bmatrix} J_{g,\boldsymbol{v}_i} \\ J_{g,\boldsymbol{\omega}_i} \end{bmatrix} \tag{4.10}$$

Analytic Jacobian

If the desire is to describe the end-effector velocities and accelerations the analytic Jacobian matrix is used. This is useful when the desire of a control loop is to control the position and velocity of the end-effector.

The position of the end-effector is given in Cartesian coordinates and orientation, and is structured like this:

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{o}_n^0 \\ \boldsymbol{\alpha}_n^0 \end{bmatrix} \tag{4.11}$$

As usual $\boldsymbol{o}_n^0$ is the translation from the manipulator base to link n, and $\boldsymbol{\alpha}_n^0$ is a minimal representation of the end-effector frame relative to the base frame. For this thesis the orientation vector will be given in Euler angles, and $\boldsymbol{\alpha} = [\phi \; \theta \; \psi]^T$.

The definition of the analytic Jacobian can be seen through the defining of the end-effector velocity:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \boldsymbol{v}_n^0 \\ \dot{\boldsymbol{\alpha}}_n^0 \end{bmatrix} = \boldsymbol{J}_a \dot{\boldsymbol{q}} \tag{4.12}$$

To describe the relationship between the geometric and the analytic Jacobian matrix one can use the Euler angle transformation $\boldsymbol{R}_{zyz} = \boldsymbol{R}_{z,\psi}\boldsymbol{R}_{y,\theta}\boldsymbol{R}_{z,\phi}$. If $\boldsymbol{S}(\boldsymbol{\omega})$ is a skew symmetric matrix the following relationships holds:

$$\dot{\boldsymbol{R}}_{zyz} = \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{R}_{zyz} \tag{4.13}$$

where $\boldsymbol{\omega}$ is the definition of the angular velocity given by

$$\boldsymbol{\omega} = \begin{bmatrix} c_\psi s_\theta \dot{\phi} - s_\psi \dot{\theta} \\ s_\psi s_\theta \dot{\phi} + c_\psi \dot{\theta} \\ \dot{\psi} + c_\theta \dot{\phi} \end{bmatrix} = \begin{bmatrix} c_\psi s_\theta & -s_\psi & 0 \\ s_\psi s_\theta & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{B}(\boldsymbol{\alpha})\dot{\boldsymbol{\alpha}} \tag{4.14}$$

Using this relationship one can derive the analytic Jacobian from the geometric Jacobian.

$$\boldsymbol{J}_a(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{B}^{-1}(\boldsymbol{\alpha}) \end{bmatrix} \boldsymbol{J}_g(\boldsymbol{q}) \tag{4.15}$$

This approach will fail when $\boldsymbol{B}(\boldsymbol{\alpha})$ is not invertible. This will be the case when all joints are aligned and singularities appear. These cases need to be taken care of by the controller to assure that it does not cause problems.

Jacobian for an Arbitrary Point on a Link

To set up a dynamic model for a robot manipulator it is necessary to not only know the Jacobian for the end effector of each link, but also to know the Jacobian matrix for an arbitrary point on a link [13]. In order to derive this Jacobian matrix one need to define a vector $\boldsymbol{r}^0_{\boldsymbol{P}_j}$ pointing to the desired point $\boldsymbol{P}$ on a link l_j in the Cartesian coordinate frame of the manipulator base. The linear velocity kinematic is then described by:

$$\begin{aligned} \boldsymbol{J}_{\boldsymbol{P}_j,\boldsymbol{v}} &= \frac{\partial \boldsymbol{r}^0_{\boldsymbol{P}_j}}{\partial q_i} = \boldsymbol{z}^0_{i-1} \times (\boldsymbol{r}^0_{\boldsymbol{P}_j} - \boldsymbol{o}^0_{i-1}), \quad \forall i \leq j \\ \boldsymbol{J}_{\boldsymbol{P}_j,\boldsymbol{v}} &= \mathbf{0}, \qquad \forall i > j \end{aligned} \tag{4.16}$$

For the angular velocity part of the Jacobian matrix the equations will be:

$$\begin{aligned} \boldsymbol{J}_{\boldsymbol{P}_j,\boldsymbol{\omega}} &= \boldsymbol{z}^0_{i-1}, \qquad \forall i \leq j \\ \boldsymbol{J}_{\boldsymbol{P}_j,\boldsymbol{\omega}} &= \mathbf{0}, \qquad \forall i > j \end{aligned} \tag{4.17}$$

Which creates the Jacobian matrix (equation 4.18) from the base to the arbitrary point on link l_j, which will later be used to derive the dynamic model of the robot manipulator.

$$\boldsymbol{J}_{\boldsymbol{P}_j}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{z}^0_0 \times (\boldsymbol{r}^0_{\boldsymbol{P}_j} - \boldsymbol{o}^0_0) & \dots & \boldsymbol{z}^0_{j-1}(\boldsymbol{r}^0_{\boldsymbol{P}_j} - \boldsymbol{o}^0_{j-1}) & \mathbf{0} & \dots & \mathbf{0} \\ \boldsymbol{z}^0_0 & \dots & \boldsymbol{z}^0_{j-1} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} \tag{4.18}$$

4.2 Dynamic Model Derivation

The dynamics of a robot manipulator describes the relationship between the motion of a robot manipulator and the forces and torques being applied to the robot by a controller or environmental impact.

In robot modeling and control it is important to know about the robot dynamics. If one is interested in the movements resulting from torques and forces it is necessary to know about the direct dynamics. If the desire is to control the position and velocities of the manipulator joints by applying forces and torques knowledge about inverse dynamics is important.

Thee two approaches in developing robot dynamics that are most commonly used are Euler-Lagrange and the Newton-Euler method. Newton-Euler focuses on building the dynamic equations based on forces and torques between all links in the manipulator. Euler-Lagrange develops the dynamics by looking at the difference in kinetic energy and potential energy [26].

In the model derivation of this thesis the Euler-Lagrange method is used. One standard way of presenting the dynamic model of a n-link robot manipulator is to to use the following dynamic equation of motion 4.19:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{4.19}$$

In equation 4.19 $\mathbf{q} \in \mathbb{R}^n$ is the joint coordinate vector, where n is the number of joints. For the UR5 robot n would be 6. $\mathbf{D}(\mathbf{q})$ is the non-singular inertia matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ is the centrifugal and Coriolis force. The joint gravity vector is $\mathbf{g}(\mathbf{q})$, and $\boldsymbol{\tau}$ is the input joint force and torque vector [31].

Equation 4.19 is also the vector-matrix form of the Euler-Lagrange equations. How to find the matrix elements will be derived in this section. It builds on finding the kinetic and potential energy of the system.

4.2.1 Euler-Lagrange Equations of Motion

It is at least two ways of developing the Euler-Lagrange equations of motion. In this thesis as in [31] the principles of virtual work will be used.

To set up the Euler-Lagrange equation of motion one needs to find equations for the kinetic and the potential energy. The Lagrangian $\mathcal{L}$ is expressed in equation 4.20, where $\mathcal{K}$ is the kinetic energy, and $\mathcal{P}$ is the potential energy. Equation 4.21 is the Euler-Lagrange equation.

$$\mathcal{L} = \mathcal{K} - \mathcal{P} = (\frac{1}{2}m\boldsymbol{v}^T\boldsymbol{v} + \frac{1}{2}\boldsymbol{\omega}^T I\boldsymbol{\omega}) - mg\boldsymbol{x} \tag{4.20}$$

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\boldsymbol{x}}} - \frac{\partial \mathcal{L}}{\partial \boldsymbol{x}_i} = f_i \tag{4.21}$$

where $\boldsymbol{v}$ is the element linear velocity, $\boldsymbol{\omega}$ is the angular velocity and $\boldsymbol{x}$ is the object position.

For robot manipulation it is often interesting to look at the the rotational joint torques, and for that case the Euler-Lagrange of motions will be as equation 4.22.

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = f_i \tag{4.22}$$

where q_i is joint i, when $i \in \{1, ..., n\}$.

4.2.2 Kinetic and Potential Energy

This section presents a way to derive the kinetic and potential energy of the n-link robot manipulator. Section 4.2.3 and how to use the energy functions to help complete the equation of motion.

Kinetic Energy for a robot manipulator

For a robot manipulator with n rotational links the kinetic energy is a sum of translational and rotational kinetic energy. The kinetic energy of a robot manipulator is therefore as follows:

$$\mathcal{K} = \frac{1}{2} m \boldsymbol{v}^T \boldsymbol{v} + \frac{1}{2} \boldsymbol{\omega}^T \boldsymbol{\omega} \tag{4.23}$$

where m is the mass of the entire manipulator, $\boldsymbol{v} \in \mathbb{R}^3$ is the linear velocity in the x, y, and z direction, and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the angular velocity around the x, y and z axis.

The inertia tensor expressed in the frame of the link is constant and not depending on joint motion. Using that information and one can express the inertia tensor in the base fram as:

$$\mathcal{I} = \boldsymbol{R} \boldsymbol{I} \boldsymbol{R}^T \tag{4.24}$$

where R is the rotation matrix from the body attached frame to the base frame. If the desire is to use an industrial robot as in this thesis, the inertia tensor of each joint are usually given. There is also a way to calculate the inertia tensors. More on that in the appendix.

By using equation 4.24 and the equations for the linear and angular velocity in equations 4.4 to 4.5 it is possible to derive the total kinetic energy for the robot manipulator with n joints as:

$$\begin{aligned} \mathcal{K} &= \frac{1}{2} \dot{\boldsymbol{q}} \left[\sum_{i=1}^{n} \{ m_i \boldsymbol{J}_{\boldsymbol{v}_{mi}}(\boldsymbol{q}) + \boldsymbol{J}_{\boldsymbol{\omega}_{mi}}(\boldsymbol{q})^T \boldsymbol{R}_i(\boldsymbol{q}) \boldsymbol{I}_{mi} \boldsymbol{R}_i(\boldsymbol{q})^T \boldsymbol{J}_{\boldsymbol{\omega}_{mi}}(\boldsymbol{q}) \} \right] \dot{\boldsymbol{q}} \\ &= \frac{1}{2} \dot{\boldsymbol{q}} \boldsymbol{D}(\boldsymbol{q}) \dot{\boldsymbol{q}} \end{aligned} \tag{4.25}$$

where the sum iterates over the robot joints, and $m_i \in \mathbb{R}$ is the mass of link i, and $\boldsymbol{J}_{\boldsymbol{v}_{mi}} \in \mathbb{R}^{3 \times n}$ and $\boldsymbol{J}_{\boldsymbol{\omega}_{mi}} \in \mathbb{R}^{3 \times n}$ is the linear and angular Jacobian matrix from the fixed base of the robot to the mass center of link i. This Jacobian matrix is calculated as explained in equation 4.18.

From the equation of kinetic energy we get the inertia matrix of the robot manipulator. This is necessary when designing a controller for a robot manipulator. In the following section the gravitational force vector is derived using the potential energy of the robot manipulator.

Potential Energy for a robot manipulator

For simplicity it is in this thesis assumed that the potential energy is concentrated at the mass center of a link. For the case of a n link robot manipulator the potential energy is the sum of the potential energy in each link.

$$P = \sum_{i=1}^{n} P_i = \sum_{i=1}^{n} m_i \boldsymbol{g}_0^T \boldsymbol{r}_{mi} \tag{4.26}$$

where m_i is the link mass, $\boldsymbol{g}_0$ is the gravity vector in the fixed inertial frame and $\boldsymbol{r}_{mi}$ is the vector from the base to the mass center of link i.

4.2.3 Equations of motion

The intention of this chapter is to derive the unknown parameters of the equation of motion (equation 4.19). This is important for knowing about the robot manipulator and being able to control in in an efficient way.

First, the gravity vector can be derived from the equation of potential energy in equation 4.26. The gravity vector can be derived by a partial derivative of the potential energy:

$$\boldsymbol{g}(\boldsymbol{q}) = [g_1(\boldsymbol{q}), ..., g_2(\boldsymbol{q})] \tag{4.27}$$

$$g_i = \frac{\partial \boldsymbol{P}}{\partial q_i} \tag{4.28}$$

for $i \in \{1, ..., n\}$.

The centrifugal and Coriolis matrix is calculated by entries of the inertia matrix in the following way:

Element number (k, j) of $\boldsymbol{C}(\dot{\boldsymbol{q}}, \boldsymbol{q})$ is defined as:

$$c_{kj} = \sum_{i=1}^{n} c_{ijk}(\boldsymbol{q}) \dot{q}_i \tag{4.29}$$

where

$$c_{ijk} := \frac{1}{2} \left\{ \frac{\partial d_{kj}}{\partial q_i} + \frac{\partial d_{ki}}{\partial q_j} - \frac{\partial d_{ij}}{\partial q_k} \right\} \tag{4.30}$$

Here, c_{ijk} are defined as the Christoffel symbols.

4.3 Multivariable Robot Control

This section will present how to use of the dynamic equations of motion to control a robot manipulator. The equations of motion form a nonlinear and multivariable system, and by using this to derive a control law for the robot manipulator one ends up with a nonlinear, multivariable controller who is guaranteed to be stable [31].

There are two ways of controlling a robot through multivariable control. One is joint space control, the other is task space control. For the joint space control the position reference is given in joint angles. For task space control the position reference is given in Cartesian coordinates.

The goal for this thesis is to establish a position controller based on Cartesian coordinates relative to the base frame of the robot. However for understanding the task space controller the joint space inverse dynamics will be derived first.

4.3.1 Joint Space Inverse Dynamics

When designing the control output we can look at equation 4.19 and rewrite it to look like this:

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{u} \tag{4.31}$$

where $\boldsymbol{u}$ is the control output which can be expressed as:

$$\boldsymbol{u} = f(\boldsymbol{q},\dot{\boldsymbol{q}},t) \tag{4.32}$$

Because of the properties of the inertia matrix $\boldsymbol{D}(\boldsymbol{q})$ the following $\boldsymbol{u}$ can be chosen:

$$\boldsymbol{u} = \boldsymbol{D}(\boldsymbol{q})\boldsymbol{a_q} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{4.33}$$

From combining equation 4.31 and 4.33 it is clear that the closed loop system gives:

$$\ddot{\boldsymbol{q}} = \boldsymbol{a_q} \tag{4.34}$$

By choosing $\boldsymbol{u}$ as in 4.33 the resulting system in equation 4.34 is linear and decoupled. This is an advantage, because then $\boldsymbol{a_q}$ can be chosen to a SISO linear second order system. For position control this can be a PD controller, as the system of position control already contains two integrators. A choice for $\boldsymbol{a_q}$ could then be a simple PD controller with feed forward acceleration:

$$\boldsymbol{a_q} = \ddot{\boldsymbol{q}}_d(t) - \boldsymbol{K}_0(\boldsymbol{q} - \boldsymbol{q}_d) - \boldsymbol{K}_1(\dot{\boldsymbol{q}} - \dot{\boldsymbol{q}}_d) \tag{4.35}$$

where $(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d, \ddot{\boldsymbol{q}}_d)$ is the reference values from the trajectory.

For this thesis the desire is to keep inside a circular path, and both $\ddot{\boldsymbol{q}}_d$ and $\dot{\boldsymbol{q}}_d$ would be 0, and there will be no feed forward of the acceleration, as that will be 0.

The controller gain matrices $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$ should be chosen to be diagonal matrices:

$$\boldsymbol{K}_0 = \begin{bmatrix} \kappa_{0,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \kappa_{0,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \kappa_{0,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \kappa_{0,4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \kappa_{0,5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \kappa_{0,6} \end{bmatrix}, \boldsymbol{K}_1 = \begin{bmatrix} \kappa_{1,1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \kappa_{1,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \kappa_{1,3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \kappa_{1,4} & 0 & 0 \\ 0 & 0 & 0 & 0 & \kappa_{1,5} & 0 \\ 0 & 0 & 0 & 0 & 0 & \kappa_{1,6} \end{bmatrix} \tag{4.36}$$

The natural frequency of the system, ω_i, is the rate at which each joint can reduce the tracking error to zero. Each element of the diagonal gain matrices $\kappa_{k,i}$ can be calculated from the natural frequency the following way to ensure a critically damped system:

$$\kappa_{0,i} = \omega_i^2, \quad \kappa_{1,i} = 2\omega_i, \quad i \in \{1, ..., n\} \tag{4.37}$$

Figur 4.2 shows the information flow of the joint space controller. Each block is built on the equations given in this section.

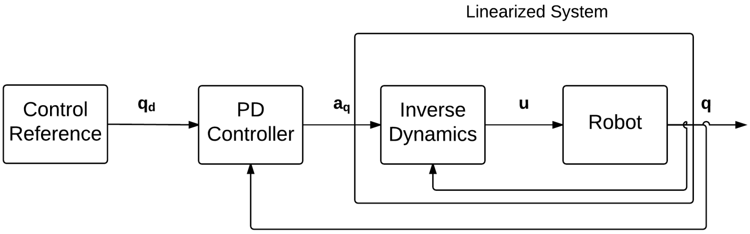


Figure 4.2: Joint space control of a robot manipulator

4.3.2 Task Space Inverse Dynamics

For many control purposes it is more interesting to have a reference trajectory in Cartesian coordinates instead of joint angles and velocities. This is the case for the control problem in this thesis. In that case the derivations above is valid, but there is the need of adding an extra element in order to transform from Cartesian coordinates to joint angles and velocities.

The position and orientation of the end-effector $\boldsymbol{X}$, and the end-effector velocity $\dot{\boldsymbol{X}}$ can be found in two ways. First option is that the Cartesian coordinates and velocities can be read directly from the robot manipulator. This is easy, as there is no need for a forward kinematic transformation, but it is not always possible.

Often only the joint variables can be read from the robot. Then the forward kinematics can be used in order to derive the end-effector position. Either way the Cartesian coordinates of the end-effector is the input to the controller.

If the end-effector position and orientation is not read directly from the robot the position and orientation is easily found using the transformation matrix and equation 4.11. Using the analytical Jacobian matrix it is possible to express the velocity and acceleration at the end-effector:

$$\dot{\boldsymbol{X}} = \boldsymbol{J}_a(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.38}$$

$$\ddot{\boldsymbol{X}} = \boldsymbol{J}_a(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_a(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.39}$$

By rewriting equation 4.39 the input $\boldsymbol{a_q}$ can be found:

$$\ddot{\boldsymbol{q}} = \boldsymbol{J}_a^{-1}\left\{\ddot{\boldsymbol{X}} - \dot{\boldsymbol{J}}_a(\boldsymbol{q})\dot{\boldsymbol{q}}\right\} \tag{4.40}$$

$$\boldsymbol{a_q} = \boldsymbol{J}_a^{-1}\left\{\boldsymbol{a_x} - \dot{\boldsymbol{J}}_a(\boldsymbol{q})\dot{\boldsymbol{q}}\right\} \tag{4.41}$$

Which results in a double integrator system where:

$$\ddot{\boldsymbol{X}} = \boldsymbol{a_x} \tag{4.42}$$

Since the reference model is given in Cartesian coordinates the input $\boldsymbol{a_x}$ can be written as:

$$\boldsymbol{a_x} = \ddot{\boldsymbol{X}}_d - \boldsymbol{K}_0(\boldsymbol{X} - \boldsymbol{X}_d) - \boldsymbol{K}_1(\dot{\boldsymbol{X}} - \dot{\boldsymbol{X}}_d) \tag{4.43}$$

Here, as with the joint space inverse dynamics the $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$ are diagonal matrices and can be chosen using the natural frequency as in equation 4.37.

For control purposes it is likely to believe that for a desired response from the robot the gain matrices $\boldsymbol{K}_0$ and $\boldsymbol{K}_1$ should be tuned to gain optimal behaviour.

Figure 4.3 shows the information flow of the task space controller. The only difference between the joint space and task space controller is that the task space controller uses the forward and inverse kinematics to transform between joint angles and end-effector position and orientation.

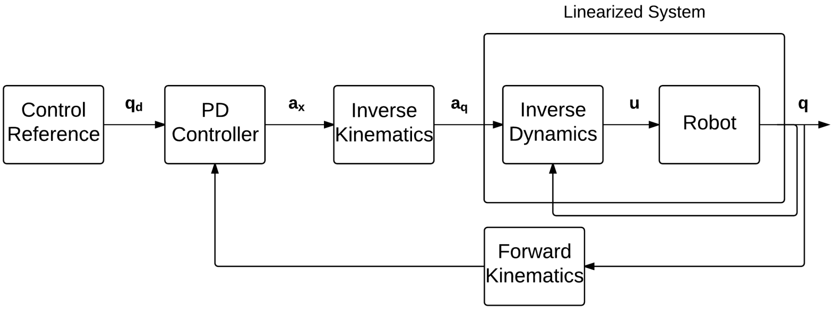


Figure 4.3: Task space control of a robot manipulator

4.4 Force Control

The above section focuses on position control based on the derived robot dynamic model. In section 3.2 there is a brief explanation of the working of a hybrid position/force controller. This section will provide the necessary equations for the force control part of the hybrid controller.

Equation 4.19 shows the dynamic equation of the robot manipulator. When the desire is to add force control to the system one have to assume that there is a force sensor attached to the end-effector. When that assumption is made, the measured force needs to be taken into account when deriving the dynamic equation for the robot manipulator, resulting in equation 4.44.

$$\boldsymbol{D}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_g^T(\boldsymbol{q})\boldsymbol{F}_e = \boldsymbol{\tau} \tag{4.44}$$

where $\boldsymbol{F}_e$ is the force measured at the end-effector, and $\boldsymbol{J}_g$ is the manipulator geometric Jacobian matrix.

From the derivation of the multivariable controller we got a controller output as in equation 4.33. For the controller that is also considering the force applied at the end-effector it is necessary to expand the control output equation to the following:

$$\boldsymbol{u} = \boldsymbol{D}(\boldsymbol{q})\boldsymbol{a_q} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{J}_g^T \boldsymbol{a_f} \tag{4.45}$$

For a simple force controller one could set that $\boldsymbol{a_f} = \boldsymbol{F}_e$. This way the controller would compensate for the force at the end-effector. For the purpose of a rehabilitation robot the desire is not to compensate fully for the force at the end-effector which originates from the patient. The desire is to provide enough resistive force to keep the patient in active training while still allowing the force of the patient to move the robot. If there is a desired force for the robot it is possible to use a standard control law like a PD controller. then $\boldsymbol{a_f}$ would have the form:

$$\boldsymbol{a_f} = \ddot{\boldsymbol{F}}_d(t) - \boldsymbol{K}_0(\boldsymbol{F}_e - \boldsymbol{F}_d) - \boldsymbol{K}_1(\dot{\boldsymbol{F}}_e - \dot{\boldsymbol{F}}_d) \tag{4.46}$$

where $\boldsymbol{F}_e$ is the force measured by the force sensor, and $\boldsymbol{F}_d$ is the desired force reference. The force control law is further discussen in section 5.2.1.

Looking at both equation 4.33 and equation 4.45 it is possible to divide robot control into position control and force control. This results in the following controller outputs position and force:

$$\boldsymbol{u}_{position} = \boldsymbol{D}(\boldsymbol{q})\boldsymbol{a_q} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) \tag{4.47}$$

$$\boldsymbol{u}_{force} = \boldsymbol{J}_g^T \boldsymbol{a_f} \tag{4.48}$$

Chapter 5

Implementation

One intention of this thesis is to verify the control strategy of the rehabilitation robot through simulations. The wish is to establish the behaviour of the controller design and to see if the design is well suited for further testing.

For the first simulations of this project the focus has been to develop the hybrid position/force controller on a planar two link robot manipulator. This is in order to make sure the controller has the desired behaviour to provide good rehabilitation for stroke patients.

The wish is to extend this controller for simulations on a model of the UR5 in the graphic simulation tool V-REP. Hopefully it is possible to use the same control law as with the two link planar manipulator, with modifications to make it control a 6 DOF robot manipulator.

This chapter presents the functional design of the controller as well as a description on how to implement the controller intended to simulate. There are aspects intended for a 6 DOF UR5 robot, but they are also explained with the necessary modifications to make the applicable on a two link robot manipulator in Matlab.

This chapter also gives a brief introduction to the simulation tool V-REP and Remote API. This is found in section 5.4.

5.1 Simulation scenario

From the project description it comes that one rehabilitation strategy is to be chosen and simulated. As presented in chapter 2 there is an agreement that for highest progress and best outcome the patient will benefit most from active rehabilitation. This is the reason that the strategy chosen to simulate in this master thesis is active rehabilitation, focusing on triggered assistance. There are suggestion for assistive training for the weak patients, as well as a challenge based strategy for the strongest patients. If a patient is not strong enough to carry out the task the device will provide help to finish the task. The hope is that there two strategies will be able to work with the same controller. This will be further discussed later in this chapter.

5.1.1 Creating the path

In order to make a simple model the thought is that the path in the first scenario is two dimensional. The idea is to create a shape and have the patient follow the edges of that shape. The very simplest example would be a one dimensional movement in a line in the x or y direction. For this project there will be suggested a path as a geometric shape in the xz plane for the 6 DOF robot. This way it is possible to take the gravitational force into account and simulate lifting objects. For the 2 DOF robot the path will be in the xy plane. The error model in this thesis will be based on a circular path, as in figure 5.1.

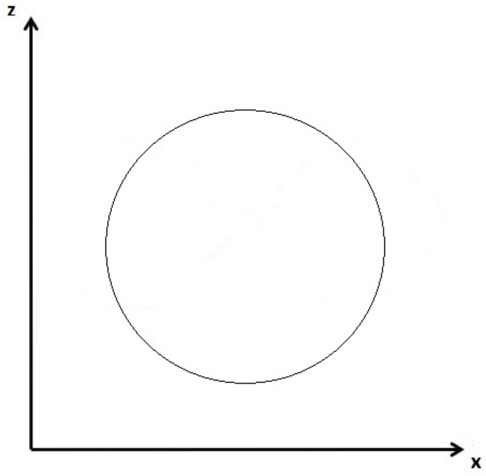


Figure 5.1: Circular path in the xz plane

5.1.2 Patient model

When is comes to rehabilitation one have to assume that there is a wide range of patient function. Some patients will have more strength and will therefore be able to apply more force to the system. Some patients might even have uncontrolled muscle movements, which the controller will need to take into account. Because of the different patient scenarios the idea is to create 4 different patient simulations.

Patient 1 - Assistive training

Patient 1 will be simulating a weak patient in need of help to be guided through the path. The controller needs to sense the force the patient is able to apply in order to make up for the rest of the movement without overcompensating. The patient should apply some force and not relax and be guided through the path.

Patient 2 - Challenge-based training

Patient 2 is a patient who has more strength in the limb. Patient 2 will need less guidance, and the idea is that the controller in this scenario will contribute an opposing force the limb has to counteract, providing muscle exercise.

Patient 3 - Assistive training

Patient 3 has some upper limb strength but is struggling with uncontrolled spastic muscle movements. The controller will in this scenario have to counteract the spastic movements of the limb and stabilize the limb so the patient is able to follow the path. The controller will also need to sense the patients ability to apply force like the scenario for patient 1 and patient 2.

Patient 4 - Too weak

Patient 4 will be a patient who is not able to overcome the trigger level of the controller. This desire is that the controller will not provide any assistance to make the patient move in the right direction until he is able to provide enough force to overcome the trigger value. The solution for this patient will be to either set the trigger value even lower, or to find a different training strategy until the limb has regained enough motor function to work the robot.

5.1.3 Triggered assistance

In order for the training to be active the robot has to make sure the patient is actively using muscles before the controller starts compensating for muscle weakness. This will be done by including a trigger based controller acting from a force sensor. This could be for example that the arm needs too apply a force larger than 0.5 N (simulating barely lifting a mass of 0.05 kg). There should be a possibility for the physical therapist to eliminate the trigger aspect if the patient is too weak. This approach will make the robot provide passive training to the patients.

There is however one scenario that is hard to control. If a patient is lazy the robot will most likely only sense weak motions from the patients and the robot will provide more work than originally needed. One option is to have an adaptive memory on the trigger value. If the patient once was able to apply a force of 1 N the controller could set the trigger value to 1 N. This is outside the scope of this thesis, but it is a possible solution to a problem that is not unlikely to occur.

5.2 Controller

Since the system controller should be able to take into account that patients have different possibilities to perform a task the controller should be adaptive to the measured force. The force measured by the force sensor needs to update the resistive force in order to make the patient gain from the rehabilitation session without it being so challenging the motivation gets lost. In order to still have a customized rehabilitation there should also be parameters to tune manually in the controller that makes the difficulty of the rehabilitation change.

As explained in section 3.2, the coordinate system of a hybrid controller will be based on the task the robot manipulator is performing. In the case of this thesis that will be the position of the end-effector in base frame. The arbitrary force constraints will be in the direction of the actual force the patient needs to apply in order to have the robot move. The arbitrary position constraints will be designed to keep the patient from moving too far away from the path. In the position controller it will be possible to introduce a dead band. This is to avoid having the controller interact when the patient makes small human errors.

For the position controller the path needs to be defined in a fixed base frame. This is so the that the position error can be easily calculated.

5.2.1 Control law

In a hybrid position/force controller there are two control laws implemented. One for position control and one for force control. The intention of this thesis is to implement well known and simple control laws, as presented in section 3.2. The advantage of a simple control law design is that the behaviour of the control law is easily predictable. This is helpful since the interesting part of this system is to investigate the behaviour of the hybrid controller interacting with the environment.

For the force controller in this thesis there will be two different scenarios, both presented later this section. One of the scenarios is a PID controller.

With modifications, the control law for position and a possibly force control is shown in figure 5.2. For discrete measurements it might be a good idea to include a low-pass filter for the derivaritve gain in the PID controller. This can be done by designing the PID controller as in figure 5.3.

The resulting equation for the controller output are shown in the following equations 5.1- 5.4, where i is the discrete value and dT is the time step.

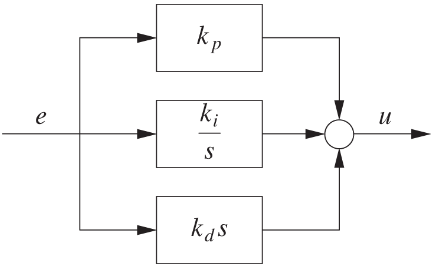


Figure 5.2: PID controller

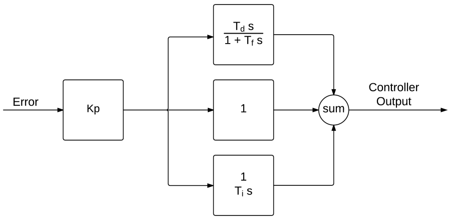


Figure 5.3: PID controller with low-pass filter

$$U(i) = K_p(U_p(i) + U_i(i) + U_d(i)) \tag{5.1}$$

$$U_p(i) = e(i) \tag{5.2}$$

$$U_i(i) = U_i(i-1) + \frac{dT}{T_i}e(i) \tag{5.3}$$

$$U_d(i) = \frac{1}{T_f + dT}(T_f U_d(i-1) + T_d(e(i) - e(i-1))) \tag{5.4}$$

As mentioned in section 3.2 the controller used in the experiment in [33] was a P controller for the position control and a PD controller for the force control. The intention is to expand to a PD controller for position and to test two different approaches to force control.

Position Control Law

Since position control already contains two integrators it is reasonable to believe that a PD controller will be suitable for position control. As mentioned, the control law from figure 5.2 can be modified to fit position control by eliminating the integrator part of the controller.

There will be no further testing on the position controller as it is assumed that the PD controller will be sufficient. If the results comes out with a different conclusion there will need to be further testing.

Force Control Law

Force control is based on controlling $F = ma = m\ddot{r}$ where r is the position. It is likely to believe that applying a force to the robot without any control law or a filter of any kind the acceleration will be too large for a patient to handle. Controlling force does not contain any integrators, which leads to believe that there is a wish to include an integral part in the control law. Figure 5.2 shows a PID controller which the author believes will be suitable for controlling force in the hybrid controller. A simple illustration of this concept is shown in figure 5.4.

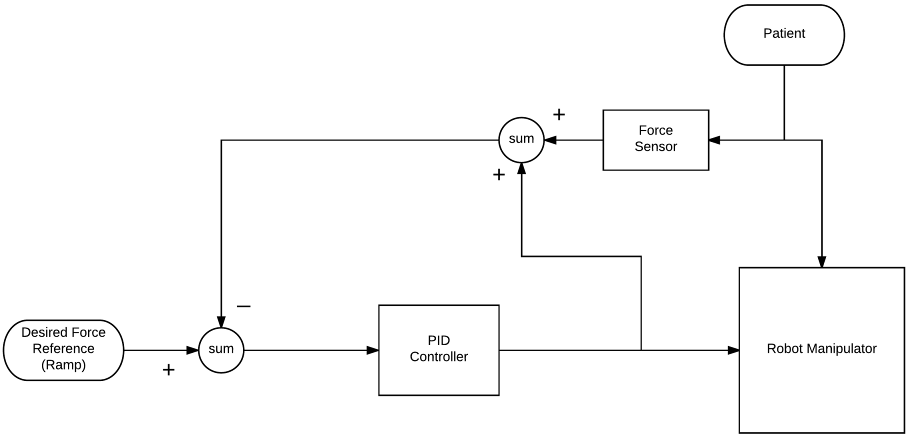


Figure 5.4: Force PID controller

Another option for the force controller could be simply to filter out the noise from the spastic movements from the patient and apply a smooth control action to the robot that brings the sum of the patient impact and the control action to the reference level. A simple illustration of this concept is shown in figure 5.5.

It is reasonable to believe that these two approaches will have different results. An assumption is that a PID controller will most likely smooth the movements of the robot by counteracting the patient's spastic movements. The low pass filter will most likely provide a steady force for the patient to work against, and the spastic movements of the patient will create a non-smooth movement of the robot end-effector. Both these approaches will be tested and simulated. The simulation results are presented in chapter 6.

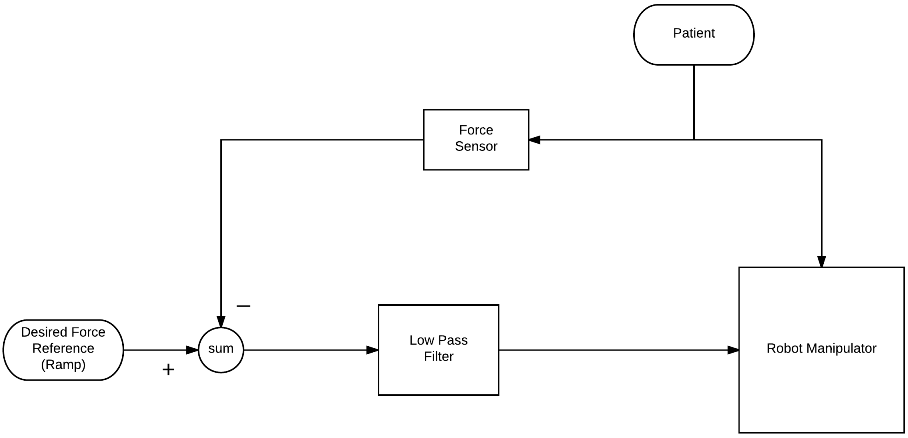


Figure 5.5: Force controller by low pass filter

5.2.2 Reference model

For the active rehabilitation it is important that the controller provides a force for the patient to counteract. Since the model contains both a position controller and a force controller there will be two error models giving rise to the hybrid controller output.

For the controller to be based on the same principle as in [27] the force controller has to drive the robot end-effector in a direction normal to the position controller.

Position Reference

The position controller will provide force to keep the robot end-effector on the path described in section 5.1.1. The idea is that if the patient is able to keep the end-effector inside the path the position controller will not interact.

In the circle reference path of this thesis, the position deviation can be calculated by equation 5.5 to 5.6, where the variables are shown in figure 5.6

$$(x-h)^2+(z-k)^2=r \tag{5.5}$$

$$e_p = r_{ref} - r \tag{5.6}$$

where r_{ref} is the desired circle radius and e_p is the position error.

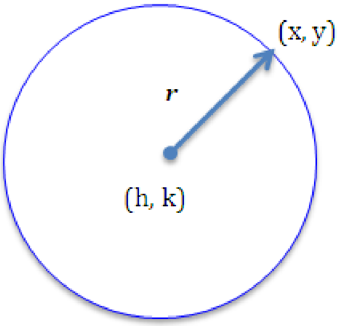


Figure 5.6: Reference circle. Figure from http://www.mathcaptain.com

Here it is possible to introduce the error-amplification approach discussed in section 2.3.3. To do this the proportional gain in the PD-controller could be set to be negative, pushing the patient away from the path if the patient starts to drift away. This approach is discussed more thoroughly in [2]. The error-amplification approach will not be in focus for the simulation of this project because the approach will most likely not be an approach that benefit the majority of the stroke victims in the early stages.

To avoid having the slightest movement outside the trajectory being compensated by the controller there is a possibility of introducing a deadband to the controller. With a deadband there is an area around the mathematical trajectory allowing human mistakes to be made without controller interaction.

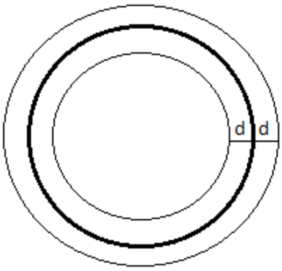


Figure 5.7: Position controller deadband

Figure 5.7 shows how a deadband in a geometric path would work. The idea is that the patient should follow the middle circle, but he is allowed to move a distance d away from the path at any moment without the interaction of the controller.

As discussed in the paper written during the fall of 2015 ([2]) there should be a reward system giving the patient points after how well he is doing. This can be done by measuring the distance away from the ideal path at all times. If the patient is following the optimal path he will get full score. If the controller needs to interact the score will be significantly lower.

Force Reference

For this thesis the force controller is the part that will provide the most most customized behaviour. There is a wish for the controller to provide a resisting force that the patient needs to oppose.

In section 4.4 the layout for the force controller was derived. For this section the focus will be to derive possible solutions to force control.

For this thesis the idea is that the movement should be caused by a constant force. If the patient is able to provide more force than desired the controller will provide a resisting force such that the sum will still be the reference. If the patient is not able to provide the desired force the controller will provide a compensating force such that the arm is still moving and following the path.

There is also a desire to include a triggering threshold in order to avoid the patient to completely ride the robot. This threshold could be a fixed value or a percentage of the desired force. This could be implemented with a switch as shown in figure 5.8.

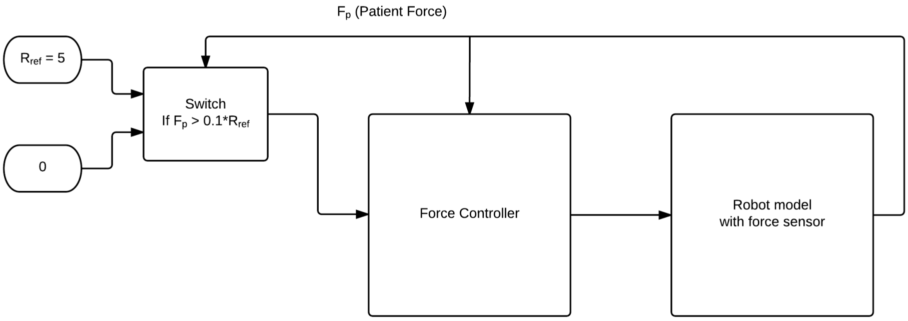


Figure 5.8: Triggered assistance implemented with a switch

This strategy results in more resistance for the patient if he has more strength. By adding the trigger one also introduces a active rehabilitation without the possibility to become passive. If the patient stops exercising the robot stops moving.

If the desire is to also have a passive rehabilitation system for the weakest patients the triggering threshold can simply be removed and the patient no longer needs to apply muscle force to make the robot move.

For a force PID controller the error model would be based on a difference between the force the patient is able to apply and the reference. However, since it is a PID controller the wish has to be that the error should be zero. This leads to the error in equation 5.7, and it can also be seen through figure 5.4.

$$e_f = F_{ref} - (F_p + u_f) \tag{5.7}$$

where F_p is the force the patient is able to apply measured by the force sensor, and u_f is the controller output, the force applied to the robot from the controller.

For the low pass filter approach the signal to be filtered would simply be the difference between the force reference and the force the patient is able to apply:

$$F_{filter} = F_{ref} - F_p \tag{5.8}$$

This is also illustrated in figure 5.5.

For the 6 DOF UR5 robot the space is three dimensional and the gravity has to be taken care of. If the path is a circle the approach explained in the following paragraphs could be implemented.

In order to provide a smooth exercise the controller needs to take into account the position and direction of the arm. If the arm is in a direction where the gravity is providing a helpful force, the controller needs to modify the reference force to make all parts of the path equally hard to pass.

Figure 5.9 shows the reference force for the force controller in the positive and negative z-direction.

The weight of the arm will in a situation that is not in horizontally planar have an impact on the error model. The work of the gravitation will be on the form:

$$F_g = m_{arm} * g * cos(\theta) \tag{5.9}$$

Where θ is the angle between the negative z direction and the tangent of the circle at the point in the trajectory. See figure 5.10.

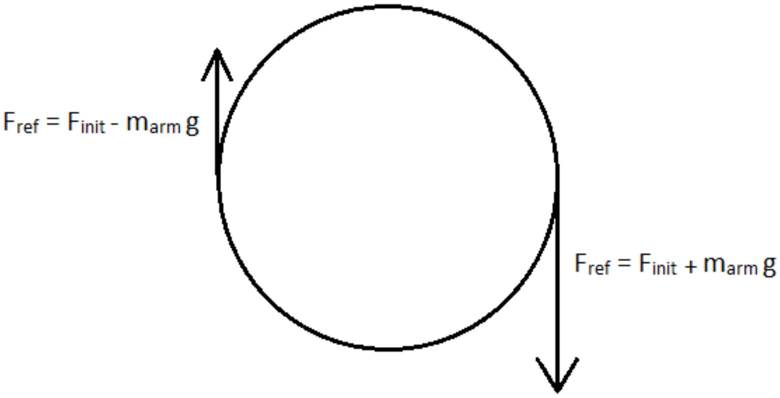


Figure 5.9: Force reference

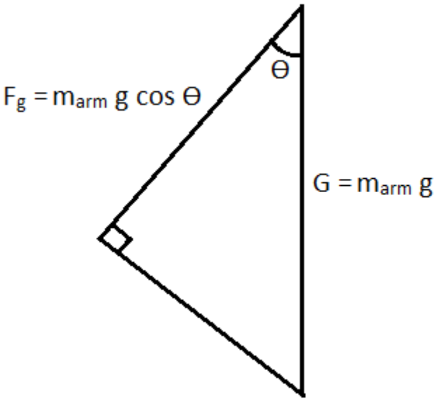


Figure 5.10: Gravitation impact

With this equation we get that the gravitational force is $m_{arm} * g$ in the negative z-direction and $-m_{arm} * g$ in the positive z-direction. The work of the gravitation in the direction of the path is 0 when the tangent of the path points in positive or negative x direction (at the top and the bottom of the circle).

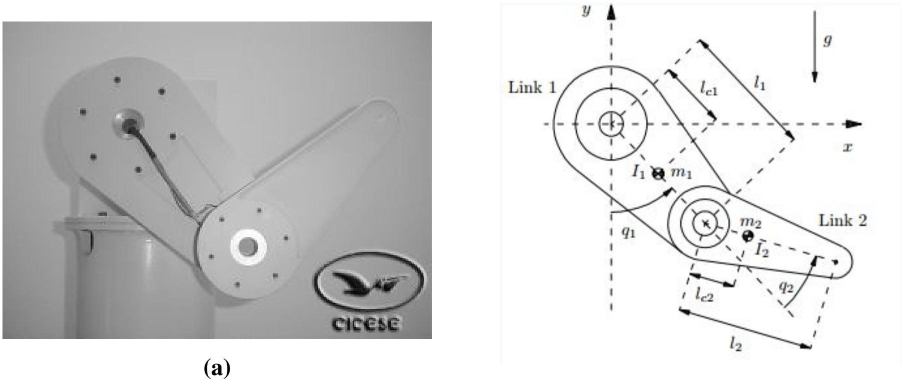
According to Shirley Kindrick, the human arm is approximately 6.5 % of the body weight. Assuming that the human average is 88 kg, the arm of the patient is assumed to be approximately 5.7 kg. There should however be a way to modify the weight of the arm between each patient in order to customize the rehabilitation.

For the two link planar robot in this thesis the force caused by gravity is not an issue in any direction of motion as the motion is defined in the xy plane. Assuming there is a possibility to rest the limb on the end-effector during the rehabilitation, the gravity is not considered for the two link manipulator. The friction caused by the interaction between the arm and a table will also be neglected.

5.3 2-link Planar Manipulator

For the first simulation scenario the focus will be a two link planar robot manipulator with two revolute joints. The hybrid position/force controller will be implemented.

Since this thesis builds on the use of industrial robot manipulators the 2-link planar manipulator model used is based on the Pelican Robot, as presented in [11]. The Pelican robot is shown in figure 5.11.



(a)

(b)

Figure 5.11: The Pelican 2-link robot manipulator, figures from [11]

Since the robot is thought to be planar in the xy plane the gravitation will be eliminated. There has not been done any more research to whether the robot is suited for the purpose. The Pelican robot has only been chosen to base the simulations on an actual robot manipulator.

5.4 Software

For the two-link manipulator the model and controller will all be designed in MathWorks's programming language Matlab. This is software well known to both the writer and the supervisors. It is also assumed known to people with knowledge in cybernetics, and there will be no explanation in this thesis on the working of Matlab.

For the second part of the simulation the software intended to use is new to the writer and the supervisors. V-REP is the simulation software intended to use with communication to a hybrid controller in Matlab. V-REP and communication through a remote API is presented in the sections below.

5.4.1 V-REP

As suggested by the assignment the simulation software to be used is V-REP. V-REP is a 3D robot manipulator simulation tool from Coppelia Robotics. V-REP can be controlled through different applications written in different languages. The idea for this project is to create the controller for the simulator in MATLAB and Simulink and control the robot from there.

V-REP also contains a model of the UR5 robot with a graphic representation which is considered an advantage.

5.4.2 Software communication - Remote API

Remote API is a part of the V-REP API framework that makes it possible for an external program to communicate with the V-REP simulator. The Remote API contains functions that makes it possible to communicate with V-REP through the languages C/C++, Python, Java, MATLAB/Octave, Urbi or Lua. The communication between the server and client using Remote API happens through sockets, making it an efficient way to communicate [38]. When one is using V-REP in a combination with Matlab there are two modes available, asynchronous and synchronous mode.

Asynchronous mode is default when using the Remote API. When using asynchronous mode, the simulation is running on the server side (V-REP) without waiting for instructions from the client side (MATLAB). This is an advantage if the simulation is not dependent on continuous information from the client. The server/client communication in asynchronous mode is shown in figure 5.12.

In synchronous mode the simulator waits for a trigger in order to move on to the next time step. This makes it possible to introduce a controller in MATLAB and control each time step in V-REP. For the controller in this thesis synchronous mode is used. The information flow in synchronous mode is shown in figure 5.13.

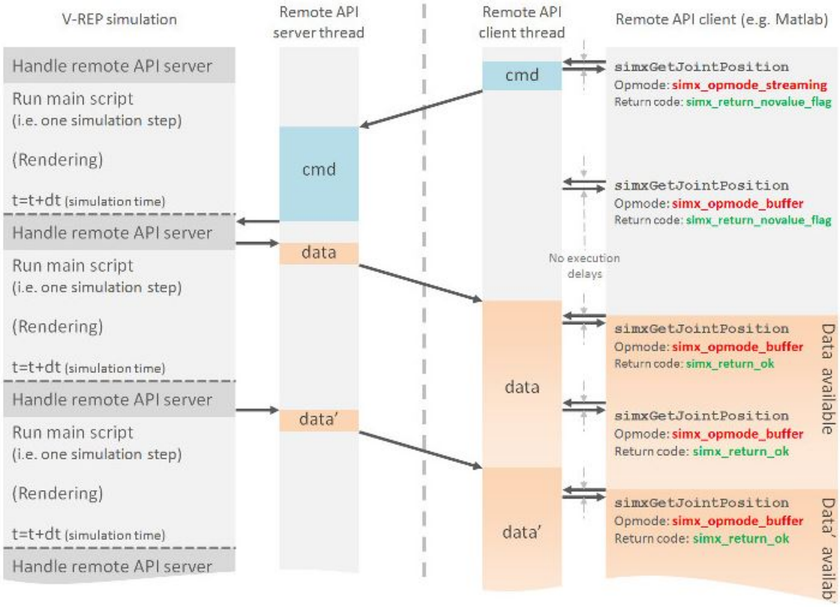


Figure 5.12: Asynchronous communication between server and client. Figure taken from http://www.coppeliarobotics.com

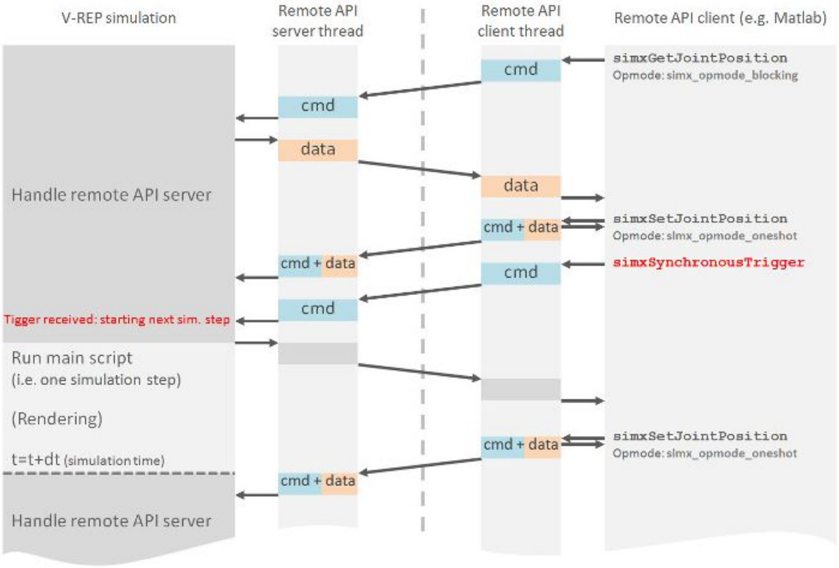


Figure 5.13: Synchronous communication between server and client. Figure taken from http://www.coppeliarobotics.com

5.5 Simulation variables

One challenge with using V-REP is the limitations on the values that can be communicated between MATLAB and V-REP. Regarding this problem the author have come up with the following communication variables for the simulator. The overall flow of variables between V-REP and Matlab is shown in figure 5.14. The variables going from Matlab to V-REP will be joint torque calculated from the hybrid controller. From V-REP the joint positions and velocities will be sent to Matlab. Matlab will then calculate the end-effector variables which will eventially be needed for controller purposes.

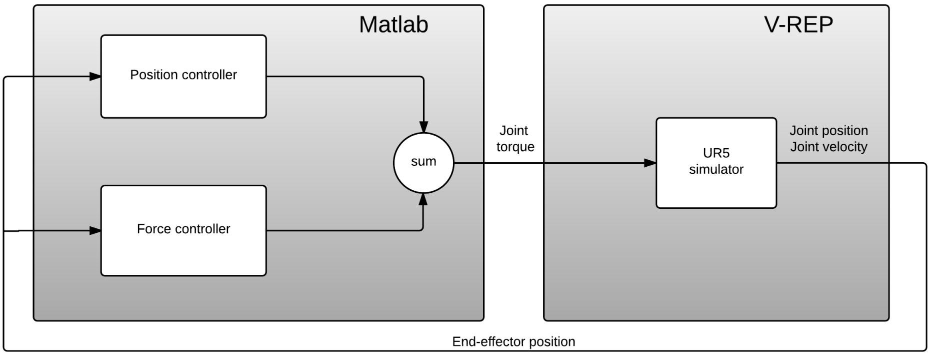


Figure 5.14: Matlab to V-REP communication variables

5.5.1 Position control variables

From V-REP the Matlab script will read the joint positions and velocities. Knowing this Matlab will use the transformation matrix (equation 4.3) to find the end-effector position and use the analytic jacobian (equation 4.15) to find the end-effector velocities.

By knowing the position of the joints and the desired position of the joints the controller can calculate the force needed for the end-effector to move to the desired position. From that the joint torques can be calculated.

5.5.2 Force control variables

The idea is that the force will differ depending on the gravity influence which will vary based on the position in the path as explained in section 5.2.2. This is why the force controller also needs information on the end-effector position which as mentioned can be found through joint positions.

For the simulations the patient model will also be designed in Matlab and applied to V-REP through joint torque calculated using the geometric jacobian (equation 4.10) in the relationship shown in equation 5.10.

$$J_g^T F = \tau \tag{5.10}$$

where F is the patient force in cartesian coordinates.

5.5.3 Hybrid controller output variables

The controller output will be the sum of the torque between the position controller and the force controller. The thought is that this is the controller output that will be sent to the actual robot.

For the simulation purpose the patient force will also be summed and applied to the robot in V-REP along with the controller output.

Chapter 6

Results

This chapter shows the results from the simulations of the control system for the rehabilitation robot. First section presents the testing of different approaches of a force controller. Later on the simulations of the 2 DOF robot is presented, and the last section shows the 6 DOF UR5 robot manipulator with the simulator in V-REP.

6.1 Force Control

The first interesting thing to investigate for the rehabilitation robot is the design for the force controller. Here a PID controller and a low pass filter has been tested and the results are presented below. The designs of the systems are presented in section 5.2.1.

As the force controller part of the hybrid controller provides the most customized behaviour it is the part that is the most interesting to investigate. It is possible to think of several ways to design the force controller and the goal of this thesis is to find the approach to fit the desired behaviour of the robot.

As explained in section 5.1.2 there are four different patient scenarios that is useful to investigate. Below is results from these four patients. For all simulations below the force reference were set to 5 N and the trigger value was set to 0.5 N.

6.1.1 Patient 1

The first patient is too week to apply the force set as a reference. The force the patient is able to apply is shown in figure 6.1. This means that the controller has to compensate by applying force to guide the patient. The wish is that the total force from the patient and the controller should be the reference force.

Figure 6.2 shows the results from the PID controller. The patient is applying a smooth force, and the result is also a smooth force from the controller and the total force is also smooth.

Figure 6.3 is the resulting plots from the simulations with a low-pass filter. One can see that the plots has a slight decrease before they start reaching the desired value. This leads the patient to feel a constraining force before they might feel that the robot is slightly loosing up and letting them provide a force allowing the robot manipulator to move. This might not be a desired behaviour the controller.

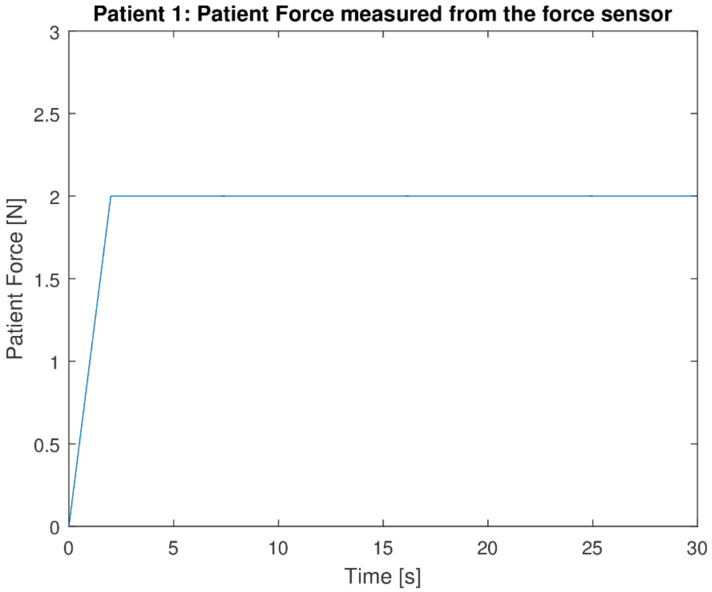
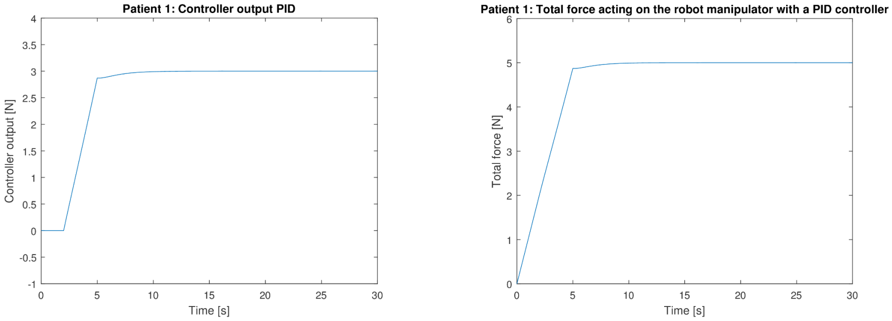
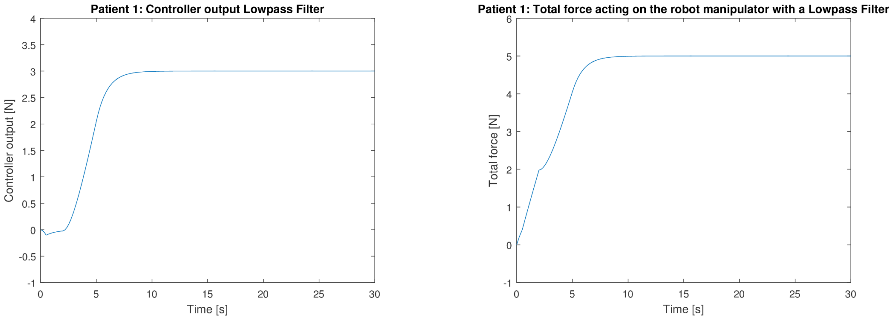


Figure 6.1: Patient 1: Force measured from force sensor



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.2: Patient 1: Control output and total force with PID



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.3: Patient 1: Control output and total force with low pass filter

6.1.2 Patient 2

Patient 2 represents a patient further along the rehabilitation process, as explained in section 5.1.2. Patient 2 is stronger, and the wish is that the controller should apply a restraining force for the patient to counteract. This would show in a plot as a negative force. The patient force measured by the force sensor is shown in figure 6.4.

Results from the PID controller are shown in figure 6.5. The PID is acting the desired way, and providing the necessary constraining force keeping the total force working on the robot manipulator at 5 N.

The low-pass filter simulations are shown in figure 6.6. The total force will have an overshoot before it stabilizes at the desired value. This is not ideal. It causes the robot to have a higher acceleration than initially desired. There might however be a possibility to tune the low pass filter in order to avoid this issue. This is something to consider if the low pass filter ends up being the chosen strategy.

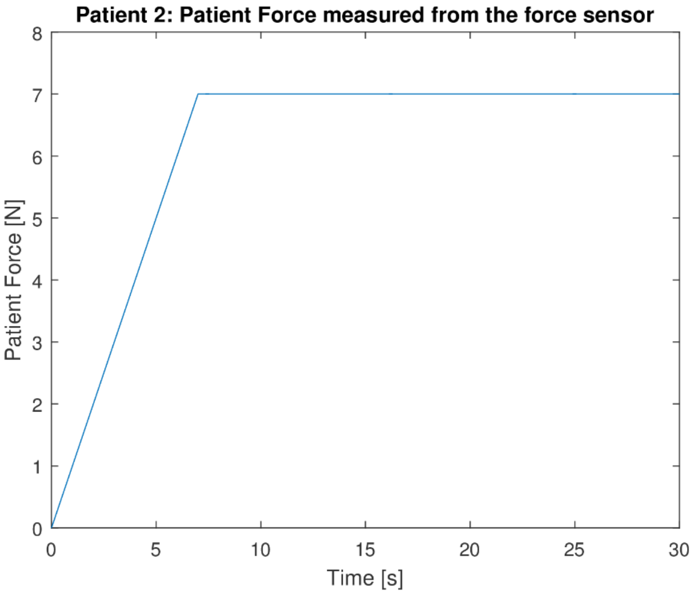
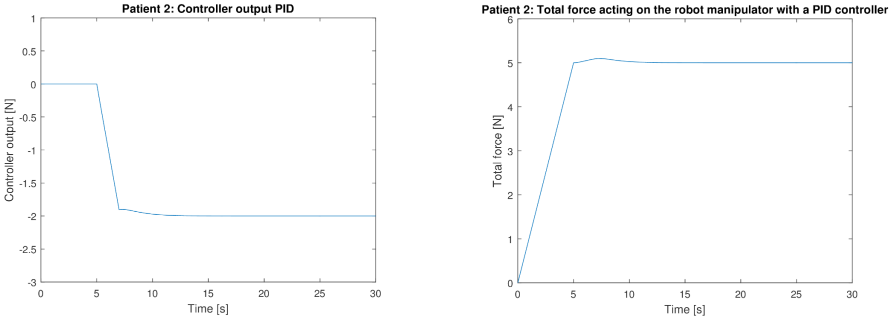
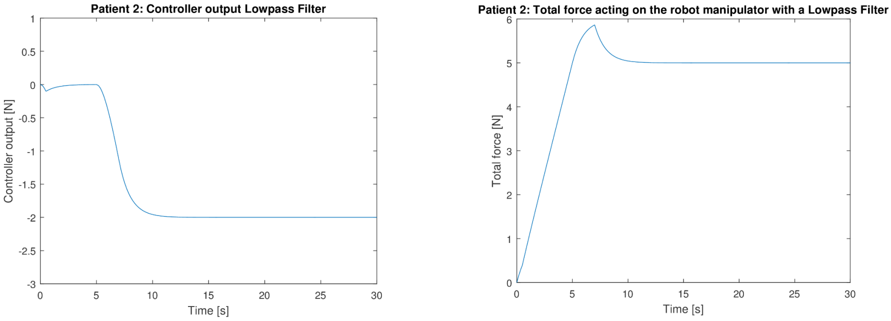


Figure 6.4: Patient 2: Force measured from force sensor



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.5: Patient 2: Control output and total force with PID



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.6: Patient 2: Control output and total force with low pass filter

6.1.3 Patient 3

Patient 3 might be the most interesting patient to investigate in a simulation. Patient 3 is struggling with spastic movements, but is able to apply the desired amount of force to the robot manipulator.

Figure 6.7 shows the output from the simulated force controller. It is clear that the spastic movements are severe for patient 3, and this is something the controller needs to take into account.

The PID controller for this scenario will work against the spastic movements of the patient and give a controller output that smooths the total force applied to the robot manipulator. This results in the robot moving smoothly even though the patient is not able to control his movements. See figure 6.8.

The low-pass filter will have to opposite effect for the case of patient 3. From the plots in figure 6.9 one can see that the control output is smoothed out and the total force acting on the patient is spastic. This will result in the robot not moving in a smooth trajectory as it will with the PID controller.

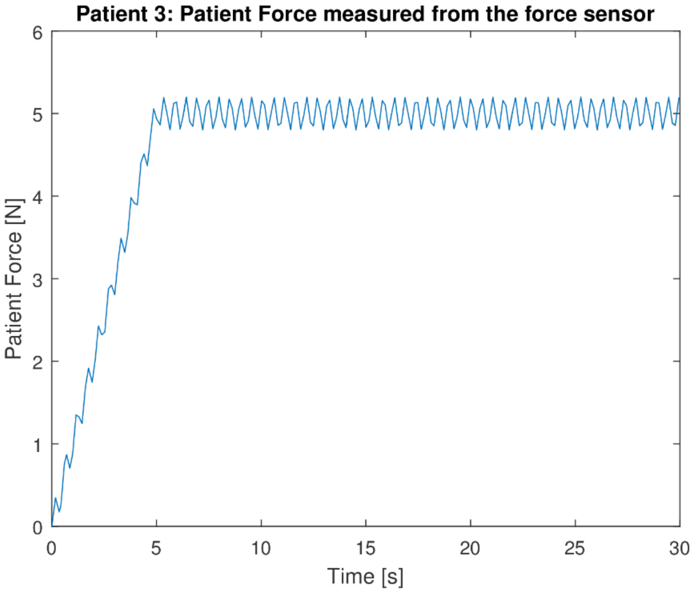
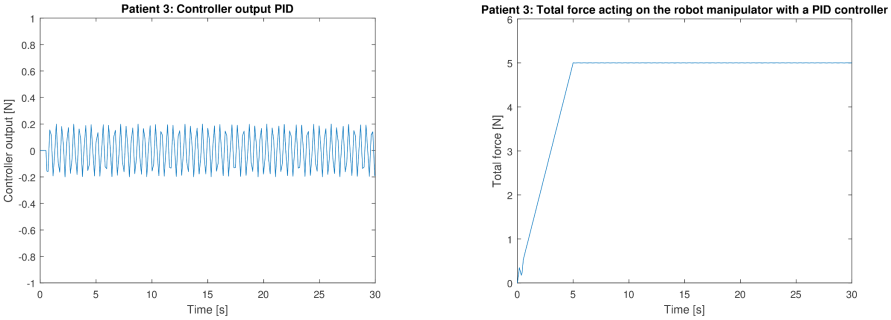
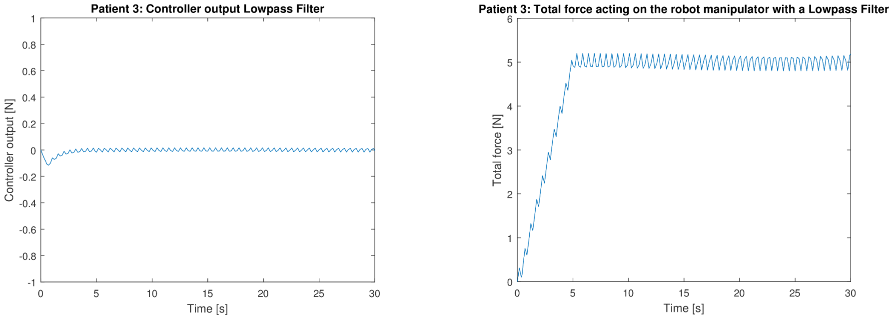


Figure 6.7: Patient 3: Force measured from force sensor



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.8: Patient 3: Control output and total force with PID



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.9: Patient 3: Control output and total force with low pass filter

6.1.4 Patient 4

From figure 6.10 it is possible to see that patient 4 is not strong enough to overcome the trigger value of 0.5 N. In this case the only force acting on the robot will be the little force the patient is applying.

According to the specification set in this thesis the controller will not act to guide a patient through the path if he or she is not able to apply a force to overcome the trigger. From the figures 6.11 and 6.12 it shows that the controller has the desired output.

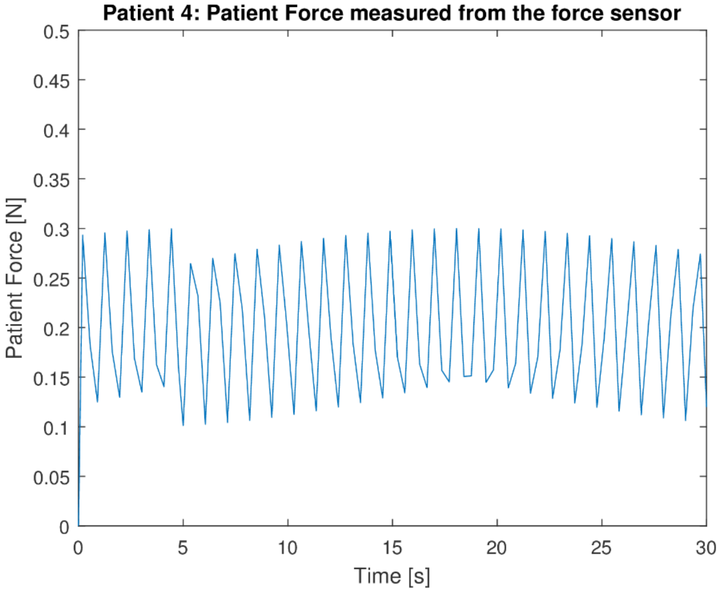
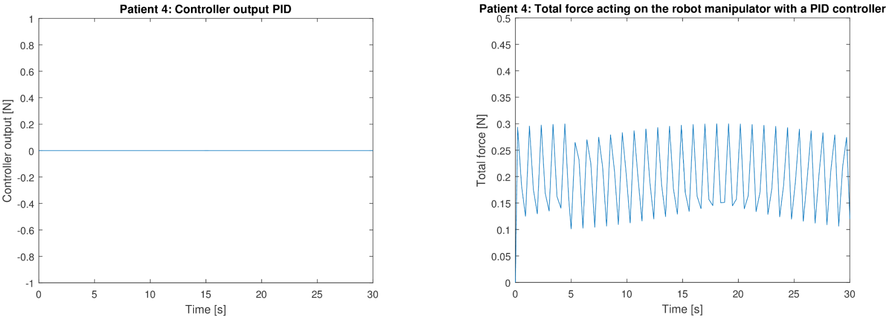
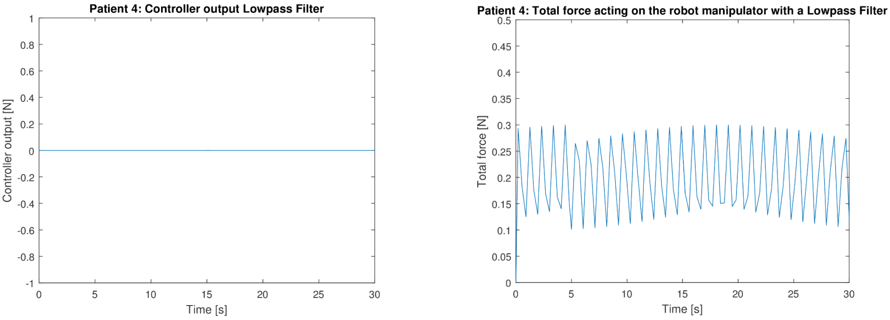


Figure 6.10: Patient 4: Force measured from force sensor



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.11: Patient 4: Control output and total force with PID



(a) Controller output **(b)** Total force acting on robot manipulator

Figure 6.12: Patient 4: Control output and total force with low pass filter

6.2 2 DOF Planar Robot Manipulator

For the 2 DOF planar manipulator the joint space controller was implemented and tested. Following that the joint space controller was attempted to extend in order to create a task space controller.

For the case of a rehabilitation robot the task space position control will be most useful in a rehabilitation environment. This is because it makes it easier to define a path for the patient to follow. It is however useful to look at the joint space control to understand the interaction between the position controller and the force controller.

As mentioned in section 5.2.2 the first important aspect to determine is the behaviour of the force controller. This was done in the previous section. The force controller seems to have the desired behaviour. For patient 1 the controller provides an assisting force making it an assistive controller. For patient 2 the controller provides a counteracting force making the controller provide a challenge based training. As hoped the controller also acts assistive for patient 3 by stabilizing the uncontrolled muscle movements.

In the following section the force PID controller have been combined with a PD position controller, which is this thesis' first simulations of a hybrid controller in a robot manipulator.

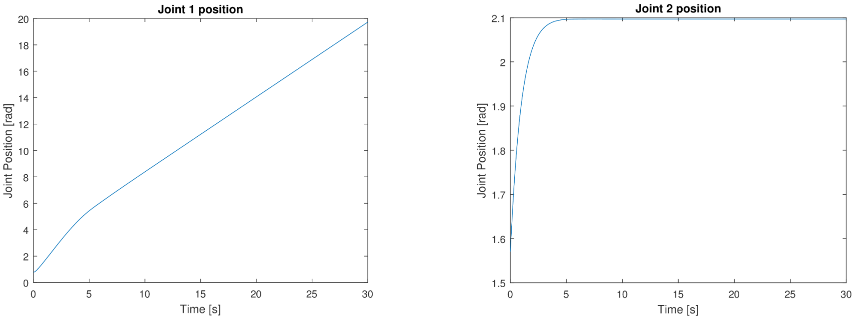
6.2.1 Joint Space Position Control

In the joint space controller it is not necessary to define a clear direction of the position controller output in Cartesian coordinates. Because of this it might be hard to make sure that the position controller works in the direction normal to the patient movement and the force controller. In a 2 link manipulator however it is possible to provide a position reference for joint 2 and let joint 1 be free. This way the reference path will create a circle whose radius depends on the position of joint 2.

Figure 6.13 shows the rehabilitation path for patient 1. Since the total force acting on the robot manipulator is similar for the PID controller with patient 1, 2 and 3 it is assumed that the simulations will be no different for those patients. It is clear that joint 2 converges to the reference value of $\frac{2\pi}{3}$. Joint 1 does not have a reference, so that when the force reference equals the joint friction the joint will have a constant angular speed and the position will have a linear increase. This becomes clear from figure 6.14.

The joint space controller shows the interpreted behaviour. The simulation plots shows the hybrid position/force controller with a behaviour as discussed in previous sections. The next step in the design of a hybrid controller will be to develop a task space position controller. This is discussed in the following section.

Figure 6.13: Robot manipulator path (x is end point)



(a) **(b)**

Figure 6.14: Joint movements with a hybrid controller on a 2 DOF robot manipulator

6.2.2 Task Space Position Control

Task space position control combined with a force controller is regarded to be the hybrid controller described as the design in this thesis. With task space position control it is possible to make sure that the position controller and the force controller controls separate directions when the controller is more complex than a 2 link planar manipulator in a circular path.

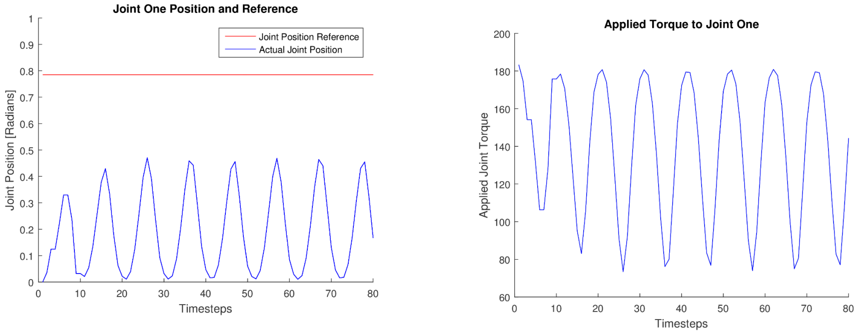
In section 4.3.2 task space control is described. This strategy was attempted in a simulation without luck. The author has not been successful in determining the problem of the simulations. It is assumed that there is an error in the transformation between forces and torques in the task space position controller. There might also be an issue with the attempts to eliminate singularities. This will need further investigation before concluding on an approach for task space control of the robot manipulator.

6.3 Simulations of the UR5

As an experiment of this thesis there was made an effort to create a joint space position controller for the 6 DOF UR5 robot in V-REP. Because the simulation tool V-REP was new to the writer there were some issues in this implementation.

The model for a 6 DOF robot was created by the use of the Denavit-Hartenberg convention and Euler Lagrange, and compared to the model created in [13]. Katharina Kufieta derives a model and a position controller in [13]. These results were verified by simulations. The model for the UR5 in this thesis coincides with the model from [13].

The simulations presented in figure 6.15 is based on a joint space controller. The reference is set to $q_{ref} = [\frac{pi}{4} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T$. Since the start position is $q_i = 0$ for all joints the only interesting joint to investigate is joint 1, the first rotational joint attached to the base link. All other joints in the simulation managed to stay at the start position.



(a) Position Reference and Actual Position of Joint 1

(b) Applied Joint Torque to Joint 1

Figure 6.15: Simulations of Joint Space Position Control in V-REP

By analyzing the force output and the movement of the joint in figure 6.15 it becomes clear that there is a need for a deeper understanding of V-REP before it is used in further simulations.

One problem that seems to appear is the link's ability to reach a reference. This is clearly shown in figure 6.15. It is clear that there are fluctuations in the joint position.

They seem to be periodic and not converging. The controller output on the other hand is applying a torque relative to the joint position and speed. As can be seen in figure 6.15b, the joint torque is positive during the whole simulation and should be driving the joint closer to the reference. It seems however that when the torque falls below a threshold the joint turns in the different direction. As joint 1 is the revolute joint attached to the base which rotates around the z-axis gravity will not be driving the joint in the opposite direction.

It is not clear to the author why the behaviour of the UR5 is as explained. There are several aspects of the simulations that could be wrong. It seems as the UR5 joints are set on a position reference initially.

When using the built in functions for reaching a joint reference the robot has no issues. It seems to be when the position controller is designed in Matlab and the built in function to apply force and torque is used the error occurs. The belief is that there is a built in controller working against the position controller designed in Matlab.

When attempts have been made to cancel out a possible drive from a initial controller the dynamics of the systems seems to fail and the robot simply falls apart with links no longer connected.

The author has not been successful in managing this challenge, and the UR5 controller is therefore not working with V-REP. This is further discussed in chapter 9.

Chapter 7

Discussion

This thesis contributes to the work of Erik Kyrkjebø at Sogn og Fjordane by designing and analyzing control strategies of an end-effector robotic rehabilitation device. For this thesis the verified controller design is based on a 2 DOF planar robot manipulator.

7.1 Functional Design

The functional design developed in this thesis is based on results from the literature review. It presents strategies the author believes will benefit a rehabilitation robot for the upper limb. The design consists of design attributes that is yet to be implemented and analyzed.

The suggestion is an approach that provides challenge-based training for the strong patients and assistive training for the weak patients. For the PD position controller and the PID force controller chapter 5 also provides suggestions to the challenge-based error-amplification approach for the patients further along in the rehabilitation.

It is likely that during future work with the rehabilitation robot suggested in this thesis there will be a need for implementations that have not been considered in this thesis. The design is merely a suggestion intended to evaluate in future work. There might also be drawn conclusions after implementation that some of the suggestions in this thesis is not optimal for the purpose of a rehabilitation robot. It will then be necessary to develop a different strategy.

To develop a rehabilitation robot is it important to consider the usability of the rehabilitation robot. In implementing a complex controller with several changeable parameters the usability of the robot might go down. This is something that is important to discuss with the target group. If the decision is made to make the system complex to reach a large group of patients it is important to create a good user interface.

7.2 Controller

The project assignment states that two of the main objectives has been to choose and implement a control strategy for robotic rehabilitation and suggest a functional design for experimental verification. The functional design for a rehabilitation robot is given in chapter 5 while the results from the implemented strategy is shown in chapter 6.

7.2.1 Verifying Position Control

In this thesis only one suggesting to a position controller was made. This was a PD controller. Through testing in joint space it seems that this approach is sufficient. If the simulations from task space control shows that the position ends up with a standard deviation outside what is defined to be the deadband there might be necessary to extend to a PID controller. It is however to the authors belief that this will not be needed.

7.2.2 Chosen Approach for Force Control

For patients without spastic movements it seems that there is not too much difference between a PID controller an a low pass filter. The though is that when the patients movement is generated from the actual patient and not a ramp the PID and the low pass filter will be even more similar.

The most interesting part of to investigate was the simulations of patient 3. The two strategies reacted very different to the spastic movements as earlier anticipated. The PID did counteract the spastic movements and makes the total force smooth. The low pass filter provides a smooth force resulting in a not smooth total force acting on the robot.

For this thesis the belief is that the PID controller will have the most desired behaviour. For a patient with unwanted spastic movements it is believed that it will feel like a relief to have the controller stabilize the limb. This is not a statement that has been discussed with rehabilitation specialists, but it is assumed to be a desired feature of the robot.

As explained in chapter 6 the force controller will make sure the approach is assistive when needed and challenge-based when the patient is strong. Another helpful part of this controller could be when the patient is strong enough to need a challenge-based control while still having uncontrolled muscle movements. By the design of the controller in this thesis the robot will then provide a counteracting force for the patient to overcome while stabilizing the arm. This is a way of having a combined assistive and challenge-based control and to the belief of the author this is a good outcome.

In section 2.3.3 the error-amplification approach was carefully described. One thought can be to introduce an error-amplification based approach in the force controller. By introducing a negative proportional gain the idea is that the spastic movements will be amplified. That approach however will need to be thoroughly discussed before introduction.

At least in early stages spastic movements are not something the patient is able to control, and an amplification of these movements might not have an ideal outcome. It is however a strategy to take into account for future development.

Triggered assistance

In the simulations of this thesis a triggered assistance approach was implemented. Through the results in chapter 6 the triggered assistance works as desired. There is however one scenario that is not considered. The controller in this thesis is built on a switch that changes the force reference based on the patients ability to apply 0.5 N (this value is easily changed). One problem could occur when the patient is having spastic movements that is varying around the limit of the trigger value. By the design of the controller in this thesis the force reference will then switch rapidly between 0 N and 5 N. This will most likely not have a desired outcome. One solution to this challenge could be that the switch is based on the mean value of the patient force. This way the spastic movements will not be an issue. Another solution could be to make sure the patient is able to apply a force over the trigger value for some time before the switch sets the reference value to 5 N. These are both strategies that could be tested to find the best suited approach.

Chapter 8

Conclusion

A literature review of end-effector based rehabilitation robots is presented in chapter 2. Based on this background theory a controller has been designed and verified through simulations through.

This thesis provides an opinion on the force control scenario of the robot manipulator along with simulations of a joint space hybrid position/force controller.

8.1 Controller Suggestions based on Simulation Results

The decision was made to use a PID controller as a force controller and a PD controller as the position controller. This was simulated on a 2 dof robot manipulator. The simulations had the desired outcome.

The position controller was early established to be a PD controller. The conclusion is that a PD controller seems to be sufficient for the purpose of this thesis. It is also the belief that the PD position controller will be sufficient for the controller on a 6 dof UR5 robot.

The conclusion for the hybrid controller has been to implement a PID controller for the purpose of controlling force. This was tested through simulations and the results are found in chapter 6. The PID force controller will stabilize the limb even if the movements of the limb is spastic. This is considered an advantage.

There is some way to go by including a task space robot controller as well as expanding the controller to be feasible for a 6 DOF robot manipulator as the UR5. There was made an attempt to introduce a joint space position controller on a UR5 robot in V-REP. The controller was designed by the model described in chapter 4. The model based controller was also compared to the work done by Katharina Kufieta ([13]) who developed a model based controller for the UR5 robot in her master thesis in 2014 with good results. The two controllers were based on the same model which leads to believe that the model is not the issue for the V-REP simulations.

8.2 Functional Design for Experimental Verification of the Results

For further testing and verification the author's recommendation is to implement the controller described in section 5.2 with a PID controller for force control and a PD controller with a deadband for position control. The reference model is also suggested in chapter 5.2.

Chapter 9

Future work

In the matter of robotic rehabilitation the will be a need for further investigation as long as the field of rehabilitation is progressing. This project will be continued by a student in his project and master thesis during the fall of 2016. Erik Kyrkjebø will also continue the work with his team in order to apply for a grant to develop a rehabilitation robot intended for use on upper limb of stroke patients.

9.1 Implementations of the functional design

A suggestion from the author is to continue the work of this thesis by implementing a model based task space controller on a 2 link robot manipulator in Matlab. Since the final robot application is though to be based on a task space controller the idea is that this approach is important to investigate in a simple controller in order to secure that the outcome is as desired.

Chapter 5 contains aspects that are intended to implement in the hybrid controller. The force reference model for the 6 DOF robot contains aspects of gravity that has not been considered in the implementation on the 2 link planar manipulator. These aspects are a part of future work.

Chapter 7 presents an approach to improve the trigger based assistance of the controller. The chapter presents two possible solutions. These are both solutions that could be tested in future work to verify which strategy provides the most suited attribute.

9.2 Implementation on an UR5

Next step in designing the rehabilitation robot will be to implement the controller designed in this thesis on the 6 DOF UR5 Robot. This can be done by using V-REP or any other simulation tool.

If the decision is to use V-REP the author will recommend having some background in using the tool or to set aside enough time to thoroughly understand V-REP and Remote API.

9.3 Experimental verification

In order to draw a final conclusion on the controller design it needs to be experimental verification on an actual robot manipulator. There is always the chance that unpredicted noise will affect the motions of the robot and that is important to distinguish before any robot can be taken into use.

9.4 Additional Features

In section 2.6 the importance of patient evaluation is discussed. Even though a patient evaluation system is not implemented in this thesis there are suggestions on how a possible evaluation approach could be designed in chapter 5. Even though the patient evaluation regime has not been the main contribution of this thesis the author still believes that the approach mentioned in chapter 5 is a possible solution.

The user interface is also important as discussed in chapter 7. This is also an aspect that is important to consider in the future work of the rehabilitation robot.

Bibliography

[1] D. Aoyagi, W. E. Ichinose, S. J. Harkema, D. J. Rinkensmeyer, and J. E. Bobrow. A robot and control algorithm that can synchronously assist in naturalistic motion during body-weight-supported gait training following neurologic injury. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 15(3):387–400, September 2007.

[2] K. Blokkum. Functional review on robotic rehabilitation of upper limb after stroke. *NTNU*, 2015.

[3] D. Chan, C. Chan, and D. Au. Motor relearning programme for stroke patients: a randomized controlled trial. *Clinical Rehabilitation*, (20):191 – 200, 2005.

[4] R. Colombo, S. Pisano, F.and Micera, A. Mazzone, C. Delconte, M. Carrozza, P. Dario, and G. Minuco. Robotic techniques for upper limb evaluation and rehabilitation of stroke patients. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 13(3):311–324, September 2005.

[5] P. Culmer, A. Jackson, M. C. Levesley, J. Savage, R. Richardson, J. A. Cozens, and B. B. Bhakta. An admittance control scheme for a robotic upper-limb stroke rehabilitation system. *IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005. Hentet fra the university of manchester.

[6] K. Fikkan. Hybrid force motion synchronization control of manipulators. *NTNU*, 2010.

[7] S. Fjeldaas. Tidlig rehabilitering av hjerneslag - farer og fordeler. *UIO*, 2013.

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

[9] B. Indredavik, R. Salvesen, H. Nss, and D. Thorsvik. Nasjonale retningslinjer for behandling og rehabilitering ved hjerneslag. Helsedirektoratet, 2010.

[10] A. E. Jackson, M. C. Levesley, J. A. Cozen, S. G. Makower, and R. J. O'Connor. Case study of patients participating in a randomized control trial of upper-limb robotic rehabilitation in acute stroke services. *IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2015.

[11] R. Kelly, D. Santibez, and J. A. L. Perez. Control of robot manipulators in joint space. 2005.

[12] S. Kousido, N. G. Tsagarakis, C. Smith, and C. D. G. Task-orientated biofeedback system for the rehabilitation of the upper limb. *IEEE 10th International Conference on Rehabilitation Robotics*, June 2007.

[13] K. Kufieta. Force estimation in robotic manipulators: Modeling, simulation and experiments. *NTNU*, 2014.

[14] B. Langhammer. Fysioterapi etter hjerneslag - en randomisert kontrollert studie. *Tidsskrift for Den Norske Laegeforening*, 2001.

[15] P. Langhorne, J. Bernhardt, and G. Kwakkel. Stroke rehabilitation. *The Lancet*, 377(9778):1693–1702, May 2011.

[16] Y. Liu and L. Ji. A hybrid force position control for a upper limb rehabilitation robot of series mechanism. In *MATEC Web of Conferences*, volume 40. EDP Sciences, 2016.

[17] R. C. V. Louriero, W. S. Harwin, K. Nagai, and M. Johnson. Advances in upper limb stroke rehabilitation: a technology push. *International Federation for Medisin and Biological Engineering*, 49:1103 – 1118, 2011.

[18] P. Maciejasz, J. Eschweiler, K. Gerlach-Hahn, and S. Jansen-Troy, A.and Leonhardt. A survey on robotic devices for upperlimb rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 11(3), 2014.

[19] L. Marchal-Crespo and D. J. Rinkensmeyer. Effect of robotic guidance on motor learning of a timing task. *Biomedical Robotics and Biomechatronics*, 2008.

[20] L. Marchal-Crespo and D. J. Rinkensmeyer. Haptic guidance can enhance motor learning of a steering task. *Journal of Motor Behavior*, 40(6):545 – 557, 2008.

[21] L. Marchal-Crespo and D. J. Rinkensmeyer. Review of control strategies for robotic movement training after neurologic injury. *Journal of NeuroEngineering and Rehabilitation*, 6(20), 2009.

[22] S. Moe, G. Antonelli, K. Y. Pettersen, and J. Schrimpf. Experimental results for set-based control within the singularity-robust multiple task-priority inverse kinematics framework. *IEEE Conference in Robotics and Biomimetics*, 2015.

[23] N. Nasr, B. Leon, G. Mountain, S. M. Nijenhuis, G. Prange, P. Sale, and F. Amirabdollahian. The experience of living with stroke and using technology: opportunities to engage and co.design with end users. *Disability and Rehabilitation*, 2015.

[24] M. K. O'Malley, A. Gupta, M. Gen, and Y. Li. Shared control in haptic systems for performance enhancement and training. *Journal of Dynamic Systems, Measurement, and Control*, 128(1):75 – 85, 2006.

[25] C. S. Poon. Sensorimotor learning and information processing by bayesian internal models. In *Engineering in Medicine and Biology Society, 2004. IEMBS'04. 26th Annual International Conference of the IEEE*, volume 2, pages 4481–4482. IEEE, 2004.

[26] M. R. P. Ragazzon. Robot manipulator collision handling in unknown environment without using external sensors. *NTNU*, 2012.

[27] M. H. Raibert and J. J. Craig. Hybrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, 103(2):126–133, 1981.

[28] C. Rsbak. Trombolyse ved hjerneslag - kliniske effekter og risiko. *UIO*, 2013.

[29] H. Seraji. Adaptive admittance control: An approach to explicit force control in compliant motion. *IEEE*, 1994.

[30] S. E. Shaw, D. M. Morris, G. Uswatte, S. McKay, J. M. Meythaler, and E. Taub. Constraint-induced movement therapy for recovery of upper-limb function following traumatic brain injury. *Journal of Rehabilitation Research and Development*, 2005.

[31] M. Spong, S. Hutchinsons, and M. Vidyasagar. *Robot Modeling and Control*. John Wiley & Sons, Inc., 2006.

[32] T. G. Sugar, J. He, E. J. Koeneman, J. B. Koeneman, R. Herman, H. Huang, R. S. Schultz, D. E. Herring, J. Wanberg, S. Balasubramanian, P. Swenson, and J. A. Ward. Design and control of rupert: A device for robotic upper extremity repetitive therapy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2007.

[33] D. L. Wedel and G. N. Saridis. An experiment in hybrid position/force control of a six dof revolute manipulator. *IEEE International Conference in Robotics and Automation (ICRA)*, pages 1638–1642, 1988.

[34] K. J. Wisneski and M. J. Johnson. Quantifying kinematics of purposeful movements to real, imagined, or absent functional objects: Implications for modellin trajectories for robot-assisted adl tasks. *Journal of NeuroEngineering and Rehabilitation*, 4(7), March 2007.

[35] G. Xu, A. Song, and H. Li. Control system design for an upper-limb rehabilitation robot. *Taylor and Frances Online*, 2010.

[36] Y. Yang, L. Wang, J. Tong, and L. Zhang. Arm rehabilitation robot impedance control and experimentation. *International Conference on Robotics and Biomimetics*, pages 914 – 918, 2006.

[37] T. Yoshikawa. Dynamic hybrid position/force control of robot manipulators - description of hand constraints and calculation of joint driving force. *IEEE Journal of Robotics and Automation*, 1987.

[38] T. Yoshikawa, T. Sugie, and M. Tanaka. Dynamic hybrid position/force control of robot manipulators - controller design and experiment. *IEEE Journal of Robotics and Automation*, 1988.

Appendix

A. Inertia Tensors

In order to derive the dynamic model for the robot manipulator it is necessary to have knowledge of the inertia tensors of each link. The inertia tensors are constant values and for some industrial manipulators they are given. The inertia tensor can also be calculated:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & Ixz \\ I_{yx} & I_{yy} & Iyz \\ I_{zx} & I_{zy} & Izz \end{bmatrix}$$

where

$$I_{xx} = \iiint (y^2 + z^2)\rho(x, y, z)dxdydz$$

$$I_{yy} = \iiint (x^2 + z^2)\rho(x, y, z)dxdydz$$

$$I_{zz} = \iiint (x^2 + y^2)\rho(x, y, z)dxdydz$$

$$I_{xy} = I_{yx} = -\iiint xy\rho(x, y, z)dxdydz$$

$$I_{xz} = I_{zx} = -\iiint xz\rho(x, y, z)dxdydz$$

$$I_{yz} = I_{zy} = -\iiint yz\rho(x, y, z)dxdydz$$

B. 2 Link Robot Manipulator Model Values

Link	θ_i	d_i [m]	a_i [m]	α_i [rad]
1	θ_1^*	0	l1	0
2	θ_2^*	0	l2	0

Table 9.1: Denavit-Hartenberg parameters for the UR5

Description	Notation	Value
Length of Link 1	l1	0.5 [m]
Length of Link 2	l2	0.5 [m]
Mass of Link 1	m1	8 [kg]
Mass of Link 2	m2	8 [kg]
Inertia relative to center of mass (Link 1)	I_1	10 [kg m^2]
Inertia relative to center of mass (Link 2)	I_2	10 [kg m^2]

Table 9.2: Physical parameters of the 2 link planar manipulator

B.1. Force Controller

The force controller gains for the PID controller matlab are:

Gain	Value
K_p	12
K_d	6
K_i	6

Table 9.3: Controller gain for the force controller in the 2 dof manipulator

The implemented low pass filter had the form:

$$K\frac{1}{\tau s+1}$$

where $K = 1$ and $\tau = 1$ was implemented.

B.2. Position Controller

The controller gains for the position controller implemented in matlab are:

$$K_0 = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$$

$$K_1 = \begin{bmatrix} 20 & 0 \\ 0 & 20 \end{bmatrix}$$

C. 6 DOF UR5 Robot Manipulator

UR5 parameters used for deriving the model and controller. Values taken from [13].

Link	Mass [kg]	Link Length [m]	Radius [m]
1	3.700	0.130	0.059
2	8.393	0.543	0.043
3	2.275	0.489	0.038
4	1.219	0.105	0.038
5	1.219	0.105	0.038
6	0.188	0.035	0.038

Table 9.4: UR5 parameters

Link	Center of Mass Vector in Base Frame
1	$[0 \quad -0.0256 \quad 0.0019]^T$
2	$[0.2125 \quad 0 \quad 0.1134]^T$
3	$[0.1199 \quad 0 \quad 0.0265]^T$
4	$[0 \quad -0.0018 \quad 0.0163]^T$
5	$[0 \quad 0.0018 \quad 0.0163]^T$
6	$[0 \quad 0 \quad -0.0012]^T$

Table 9.5: UR5 vector to center off mas in base frame

The inertia matrices of the UR5 are as follows:

$$I_1 = \begin{bmatrix} 0.0067 & 0 & 0 \\ 0 & 0.0064 & 0 \\ 0 & 0 & 0.0067 \end{bmatrix} \qquad I_2 = \begin{bmatrix} 0.0149 & 0 & 0 \\ 0 & 0.3564 & 0 \\ 0 & 0 & 0.3553 \end{bmatrix}$$

$$I_3 = \begin{bmatrix} 0.0025 & 0 & 0.0034 \\ 0 & 0.0551 & 0 \\ 0.0034 & 0 & 0.0546 \end{bmatrix} \qquad I_4 = \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix}$$

$$I_5 = \begin{bmatrix} 0.0012 & 0 & 0 \\ 0 & 0.0012 & 0 \\ 0 & 0 & 0.0009 \end{bmatrix} \qquad I_6 = \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.0001 & 0 \\ 0 & 0 & 0.0001 \end{bmatrix}$$