Detecting Changes During the Manipulation of an Object Jointly Held by Humans and Robots

VALERIA REYNAGA BARBA
Detecting Changes During the Manipulation of an Object Jointly Held by Humans and Robots

Degree Project in Computer Science and Communication, Second Cycle (DA226X)

VALERIA REYNAGA BARBA

Master's Thesis at Computer Vision and Active Perception Department
Supervisors: Yiannis Karayiannidis <yiankar@kth.se>
            Christian Smith <ccs@kth.se>
Examiner: Danica Kragic <dani@kth.se>
Acknowledgements

First and foremost, I would like to thank my mother, Isela Barba Ramos, for everything she has taught me and for supporting me in becoming better professionally and personally. She has taught me that hard work leads to success. She is my role model and I dedicate this work to her.

I would like thank my supervisors, Yiannis Karayiannidis and Christian Smith, for guiding me and providing the support I needed to complete this research work.

I would like to thank the students that helped me collect data for this research work by performing the transition sequence. They are: Einar Snorrason, Carlos Javier Pedreira, Jesper Thörnberg, Erik Sillén, Unnar Axelsson, Jacob Greenberg, Clara Scherer and Charles Barnouin.

Finally, I would like to specially thank Einar Snorrason for the support he has given me throughout the time I did this research work.
Abstract

In the last decades research and development in the field of robotics has grown rapidly. This growth has resulted in the emergence of service robots that need to be able to physically interact with humans for different applications. One of these applications involves robots and humans cooperating in handling an object together. In such cases, there is usually an initial arrangement of how the robot and the humans hold the object and the arrangement stays the same throughout the manipulation task. Real-world scenarios often require that the initial arrangement changes throughout the task, therefore, it is important that the robot is able to recognize these changes and act accordingly. We consider a setting where a robot holds a large flat object with one or two humans. The aim of this research project is to detect the change in the number of agents grasping the object using only force and torque information measured at the robot’s wrist. The proposed solution involves defining a transition sequence of four steps that the humans should perform to go from the initial scenario to the final one. The force and torque information is used to estimate the grasping point of the agents with a Kalman filter. While the humans are going from one scenario to the other, the estimated point changes according to the step of the transition the humans are in. These changes are used to track the steps in the sequence using a hidden Markov model (HMM). Tracking the steps in the sequence means knowing how many agents are grasping the object. To evaluate the method, humans that were not involved in the training of the HMM were asked to perform two tasks: a) perform the previously defined sequence as is, and b) perform a deviation of the sequence. The results of the method show that it is possible to detect the change between one human and two humans holding the object using only force and torque information.
Referat

Detektera skillnader under manipulationen av ett objekt som gemensamt hålls av människor och robotar

### Contents

1 **Introduction**  
1.1 **Goal** ........................................... 2  
1.2 Sustainable development and ethical aspects ........... 2  
1.3 **Outline** ........................................ 2  

2 **Related Work**  
2.1 Cooperative manipulation tasks .............................. 5  
2.1.1 Challenges in cooperative manipulation .................. 5  
2.2 **Contact point estimation** .......................... 7  

3 **Problem Description**  
3.1 **Problem Definition** .................................. 9  
3.2 Physical system ...................................... 11  
3.2.1 The robot ........................................... 11  
3.2.2 The sensor ........................................ 12  
3.2.3 The object of manipulation ............................. 14  
3.3 Proposed solution ..................................... 15  
3.3.1 The transition sequence ................................ 15  

4 **Background**  
4.1 Kalman filter .......................................... 19  
4.1.1 The Kalman filter algorithm .......................... 20  
4.2 Hidden Markov Models .................................. 21  
4.2.1 Discrete Hidden Markov Models ...................... 21  
4.2.2 Continuous Hidden Markov Models .................... 25  

5 **Grasping point estimation**  
5.1 Kalman filter implementation ............................ 27  
5.1.1 Choosing the model .................................. 27  
5.1.2 Tuning the Kalman filter ............................ 32  

6 **Tracking the number of agents**  
6.1 Hidden Markov Model Implementation ..................... 37  
6.1.1 Choosing the model .................................. 37
6.1.2 Training the HMM ............................................ 40

7 Experiments ......................................................... 55
  7.1 Description of the experiments ............................................ 55
    7.1.1 Experimental setup ............................................ 55
    7.1.2 Experimental tasks ............................................ 56
  7.2 Results ................................................................ 58
    7.2.1 Results for model HMM$_F$ ............................................ 58
    7.2.2 Results for model HMM$_{NF}$ ...................................... 62

8 Conclusions .......................................................... 65
  8.1 Summary .............................................................. 65
  8.2 Future Work ............................................................ 66

Bibliography ............................................................. 67

Appendices ................................................................. 70

A Grasping point estimator results ....................................... 71
  A.1 Group 1 .............................................................. 71
    A.1.1 Task 1 .............................................................. 71
    A.1.2 Task 2 .............................................................. 73
  A.2 Group 2 .............................................................. 75
    A.2.1 Task 1 .............................................................. 75
    A.2.2 Task 2 .............................................................. 79

B HMM results .......................................................... 83
  B.1 HMM$_F$ .............................................................. 83
    B.1.1 Task 1 .............................................................. 83
    B.1.2 Task 2 .............................................................. 85
  B.2 HMM$_{NF}$ ............................................................ 88
Chapter 1

Introduction

The demand for robots to physically interact and assist humans is growing for different applications, some in manufacturing industries, such as assembly lines for automobile manufacturing plants, aerospace industry and construction fields [1]. Other environments where robots are expected to be found are in office, house, medical and welfare applications [2], including robots that act as companions [3]. The purpose of having robots assist humans in any task should be to increase their capabilities by increasing precision [1], while reducing fatigue and stress [3].

In tasks where humans cooperate to accomplish a common task, (either by direct contact or through an object), they communicate through speech, gestures and the haptic channel. Haptics refers to manual sensing of surrounding objects and environments through the sense of touch [4]. It is through the haptic channel that humans exchange forces and motions to negotiate and accomplish a given task, this is known as physical interaction [5]. Moreover, physical human-robot interaction (pHRI) refers to the interaction between humans and robots by direct or indirect contact. Examples of pHRI tasks are dancing [6], teaching a robot by demonstration [7] or jointly manipulating an object [8].

When jointly manipulating a long object, there is an initial arrangement established of how many agent[1] are holding the object and where they will be holding, and this arrangement is expected to remain the same throughout the duration of the task. However, real-world scenarios often require that the initial arrangement changes. For example, one more agent is required to hold the object for only part of the manipulation task, or one or more agents are required to switch places with new agents. In such cases, it is important that the robot is able to recognize these changes and act accordingly because the robot has limitations in the way it moves with respect to where the humans are grasping. These limitations are set so that the humans are not disturbed by forces or torques in directions that are uncomfortable

---

[1] In this report, an agent refers to a human or a robot involved in the joint manipulation of an object.
CHAPTER 1. INTRODUCTION

for humans in general.

1.1 Goal

The aim of this research project is to detect and identify a change in the arrangement of agents using only force and torque information measured at the wrist of the robotic arm grasping the object. The case considered in this work is to distinguish between two scenarios: 1) one human grasping the object and 2) two humans grasping the object. To distinguish between these two scenarios, there are some assumptions made:

- The humans grasping the object and the robot do not perform a manipulation task.
- The initial scenario is always one human grasping the object with the robot.
- A predefined sequence of steps must be performed by the humans to transition from the initial to the final scenario.

The proposed method to detect a change in the number of agents consists of estimating one grasping point using a Kalman filter and the force-torque information, and then using the estimated grasping point to track the steps in the predefined sequence using a hidden Markov model. With this, we know how many agents are grasping the object by knowing which step of the sequence the agents are in.

1.2 Sustainable development and ethical aspects

Robots are expected to integrate in society in the near future, for this, robots should be able to understand what is happening in their surroundings so that the interaction with humans is safe and comfortable. In a broad perspective, the research done aims to provide a better understanding of the environment of the robot, which is determining the number of people grasping the object with the robot. The more the robot understands its surroundings, the better the robot can act with respect to this understanding, thus, creating a safer human-robot interaction.

1.3 Outline

The aim of this chapter is to introduce the reader in the research field of pHRI and define the goal of this research project. The remainder of the thesis is organised as follows. Chapter 2 provides relevant research related to pHRI and to the problem at hand. Chapter 3 formalizes the problem and introduces why the solution is composed the way it is. Chapter 4 provides the background of the two key methods that compose the proposed solution of the problem at hand. Chapter 5 explains
1.3. OUTLINE

the implementation of the Kalman filter as the estimator of one grasping point. Chapter 5 describes the implementation of the hidden Markov model to determine when a change happened and what change happened given the estimation. Chapter 6 describes the experiments performed for the evaluation of the proposed solution and the results obtained from them. Finally, the conclusions of this thesis will be presented in Chapter 8.
Chapter 2

Related Work

2.1 Cooperative manipulation tasks

As mentioned previously, there is a big interest in humans to physically interact with robots for different applications. One relevant problem in physical human-robot interaction (pHRI) is a scenario where humans and robots cooperatively manipulate an object. In such cooperative manipulation tasks, the concepts of passive and active behaviour in robots have been introduced. A robot is passive when it only reacts according to a human imposing a trajectory [9, 10]. On the other hand, active participation refers to a robot that can take the lead or follow the human depending on the situation, in the same way as humans exchange roles when manipulating objects [10].

Cooperative manipulation tasks include kinesthetic teaching, where humans demonstrate explicitly how the object should be moved by the robot; it also includes joint manipulation tasks for moving an object from one place to another [11]. One approach to kinesthetic learning is presented in [12], where the robot control is learnt during physical interaction with humans while only using haptic and proprioceptive feedback. Another approach to this is presented in [11], where a probabilistic framework is used to encode the dynamics of the motion and haptic communication during physical collaborative tasks. In [13], the robot learns by segmenting the task that the human demonstrates, and learns important variables that are linked to each segment of the task.

2.1.1 Challenges in cooperative manipulation

Safety

Different challenges have been studied regarding joint manipulation tasks between humans and robots. One of them is regarding safety in such tasks, since when a robot is passive, it takes care of the load and stabilization while the human is lead-
CHAPTER 2. RELATED WORK

ing the trajectory [11]. To improve safety, a common approach is to use different
types of impedance controllers. In [12] mentioned previously, besides learning a task
by demonstration, they use an adaptive control algorithm to tune the impedance
parameter to ensure the accurate reproduction of the task. An impedance controller
based on human impedance characteristics is implemented in a robot to perform
a cooperative task with a human in [14]. In [15] an impedance control is used for
a manipulation task where the robot interacts with the human estimating his/her
intent from observing the control effort. Impedance control has been proposed as
well for a mobile robot helper in [2] and [16].

Intention of motion

Another challenge in cooperative manipulation tasks is interpreting the intention
of motion of the human while manipulating an object so that the robot can adapt
accordingly. In [17], a system was proposed where multiple robots and humans
execute a task and where the robots are commanded only by the forces and torques
the humans apply. The challenge of interpreting intention of motion arises when
the object to be manipulated is long given that humans cannot effectively apply
torques at the point where they are grasping, it is easier for the human to apply
forces through the object to obtain rotation [9]. This, in turn, makes it hard for the
robot to distinguish the intention of the human to rotate or to translate from force
and torque measurements [11,11].

There have been different ways to address the above described problem. One,
is to set virtual constraints to allow the robot to only move in certain directions,
this approach disregards the disambiguation of the humans’ intention in terms of
rotation and translation, and the human must adapt to move only in certain direc-
tions to guide the robot. In [18], the robot is constrained to move as if the object
in manipulation was attached to a wheel. Another example of such constraints is
presented in [19], where the human manipulates the object as if it was a water hand
pump. Another way to address the ambiguity problem is to consider the possibility
of having both types of motion (rotation and translation) and determining them
from the commands given by the human through different channels. In [11], only
force measurements are used to determine the intention of the human using the
magnitude of the force applied by the human, and with this, the robot switches
between two modes of operation: translation and rotation. In addition to haptic
feedback, other channels of communication have been considered to distinguish hu-
man intentions. In [20], besides sensing the force applied by the human, a speech
recognition system is used to allow the human to guide the robot by saying specific
commands.

Recent studies have focused on the active behaviour of robots. In [21], the robot
is proactive in the way that it can guess the human’s trajectory and reduce the in-
2.2 CONTACT POINT ESTIMATION

interaction force. The robot can also switch between the role of follower and leader
during the manipulation task, although the robot is controlled with a joystick while
having a leader role. Another approach is to decide how to designate these roles as
in [9], where strategies for static and dynamic role allocation are proposed for the
manipulation of bulky objects.

2.2 Contact point estimation

The research mentioned above assumes that the initial distribution and number of
the agents carrying the object will remain constant throughout the complete trans-
portation task. Detection of changes in the number of agents or their grasping points
during cooperative manipulation has not been studied as such. As we get closer to
getting robots to act as humans do in cooperation tasks, we have to address the
cases where the distribution and number of agents may change during manipulation.

Human grasping point

When an object is held by a number of agents (humans or robots) there are certain
constraints imposed on the object at the points where the object is grasped. There
have been approaches towards estimating the grasping point. In [10], a method is
proposed for estimating the constraints imposed by a human grasping an object
with a robot, and with this, know where the human is grasping so the robot can
plan trajectories to manipulate the object while subject to the constraints. In [22],
wearable motion sensors were used to estimate the pose of the human grasp, where
they are able to identify the unknown relative displacement orientation of the hu-
man.

Object contact point

Other studies have been done for scenarios that do not involve estimating the human
grasping point; they focus on finding points where the grasped object is in contact
with the environment. In [23], for example, the different constraints that can be
imposed on a polyhedral object by its contact with the environment are linked to
different measurement models. Transitions from one measurement model to another
are determined by a statistical inconsistency of the measurements with respect to
the current measurement model. Analogous work has been done in [24] to estimate
the contact state between a robot and its environment for a peg-in-hole insertion
task. In this work, a non-linear least-squares algorithm is used for the multiple
model estimation to determine the most likely contact state using a hidden Markov
model. This is done by running multiple model-based estimators at every specified
time step to determine the most likely contact state. In [25], a multilayer percep-
tron (MLP) is used to identify the contact state during an assembly process. They use the force and position measurements as input to the MLP and the individual outputs correspond to all the possible contact states, where the one with the highest value is chosen and is the input to a controller that guides the work piece through a series of events to accomplish the assembly task. Furthermore, in [26] a Bayesian static multiple model (SMM) is used to determine the various possible object-finger contact modes, and they fuse stereo vision, force-torque information and joint angle encoder measurements to estimate and track the location of a grasped object by a robotic hand.

Estimating the contact point for tool manipulation tasks has also been studied, like in [27], where the task of the robot is to machine contour tracks. In this work, they use force and velocity information to detect the contact point. For [27] and the studies described in this section the object model is known. There have been studies where using the imposed constraints on an object can be used without considering the object model. This is the case of [28], where they are able to estimate and track online where a tool, that is grasped by a robot arm, is in contact with the environment based on an adaptive estimation scheme and force and torque measurements from a sensor mounted at the wrist of the robot’s arm.
Chapter 3

Problem Description

In this chapter, the problem to be solved is described thoroughly as well as the physical system and an introduction to the proposed solution is presented. In the description of the problem the notation that is used throughout the thesis is introduced. The physical system described comprises the robot, the sensor and the object used for manipulation. The proposed solution involves performing a change in the grasping of the object using a predefined sequence which is defined here. The way that the information from performing this transition sequence is used is only explained briefly here for it is explained in detail in the following chapters.

3.1 Problem Definition

Consider the scenario in Figure 3.1(a), where one robot and one human are grasping the object. The frame $\{h_1\}$ has its origin at the point where the human grasps the object. As for the robot, the frame $\{r\}$ has its origin at the point where the gripper grasps the object, and the sensing frame of the force-torque sensor is defined as $\{s\}$.

Figure 3.1. Example of scenarios of humans and robots holding an object. (a) One human holding an object with the robot. (b) Two humans holding an object with the robot.
In the case where two humans are holding the object along with the robot, like the one shown in Figure 3.1(b), the points where the humans are grasping are referred to as the origins of frames \( \{h_1\} \) and \( \{h_2\} \).

While holding the object at a single point, humans have difficulties applying torques around the point they are grasping; it is easier for them to apply forces at the grasping point. In the case of Figure 3.1, whatever forces the humans apply, they will be perceived at the sensor frame along with the torque generated by applying these forces at a distance. As an example, assume that the agent in Figure 3.2 applies a force \( F_{h_1} \) somewhat towards \( \{s\} \). This force generates a torque at the sensor frame that depends on the vector \( r_s \) connecting the origins of frames \( \{h_1\} \) and \( \{s\} \). The forces and torques produced by the human are mapped to the force vector \( F_s \) and the torque vector \( \tau_s \). The relationship between these two vectors is given by:

\[
\tau_s = r_s \times F_s \tag{3.1}
\]

It is important to note that the point where the robot grasps the object, at the origin of \( \{r\} \), has no relevance to this problem, since the wrenches are expressed in the sensor frame \( \{s\} \).

The distance from the sensing frame to the grasping point of the human can be determined by estimating vector \( r_s \) shown in Figure 3.2. The estimation of vector \( r_s \) will be denoted as \( \hat{r}_s \) and it can be obtained by using the relationship in (3.1), with \( F_s \) and \( \tau_s \) being known from measurements.

---

1 A wrench refers to the pair of vectors, force and torque, acting on a rigid body [29].
3.2. PHYSICAL SYSTEM

Estimating vector $\hat{r}_s$ when there is one agent grasping the object along with the robot is straightforward. However, what happens when there are two agents holding the object? In that case, each agent can produce wrenches at their respective grasping points, and the measured wrench $W_s$ becomes a combination of the applied wrenches. The result of the estimation will still be the distance to one point. By intuition, we could say that the estimation would give the distance to a point in between the agents. However, where the point will be depends on where the agents are grasping, and the magnitude and direction of the forces they exert.

Consider again the two scenarios in Figure 3.1. Assume that, initially, the configuration of agents is the one shown in Figure 3.1(a). Then assume that another agent is needed to perform a task and that both agents would like to end up holding the object as shown in Figure 3.1(b). As stated above, it is unknown where the estimation of $r_s$ will converge to in the latter case. However, to go from the initial scenario to the final scenario, the agents will perform a sequence of steps, grasping at different points during the transition. This means that $\hat{r}_s$ will undergo certain changes that depend on where the agents choose to grasp to transition between the initial and final scenarios.

If a defined sequence is followed to go from one scenario to the next, and if the final and initial step of the sequence are associated with a number of agents grasping the object, then it is a matter of determining what step of the sequence the agents are in to determine how many agents are grasping the object. A transition sequence between two scenarios will be defined as $S$ and the number of transitions or states in the sequence as $N$, such that $S = \{S_1, S_2, \ldots, S_N\}$. The number of agents holding the object will be referred to as $\Lambda$. It is assumed that each state $S_i$ produces a different estimation of $\hat{r}_s$. Since this is a problem of tracking, $\hat{r}_s$ will be continuously updated and associated to a state belonging to $S$. Therefore, the number of agents holding the object at each time step $\Lambda_t$ is given by the point $\hat{r}_{s_t}$ that is produced by being in state $S_i$:

$$\Lambda_t = f(\hat{r}_{s_t} \mid S_i); \quad 1 \leq i \leq N \quad (3.2)$$

3.2 Physical system

3.2.1 The robot

The robot used for this research project is a PR2 robot from Willow Garage located in the Robotics Lab at the Computer Vision and Active Perception (CVAP) department at KTH Royal Institute of Technology. An image of a PR2 robot is shown in Figure 3.3. Each complete arm of the PR2 has 8 DOF: 1 DOF for the gripper, 3 DOF for the wrist, and 4 DOF for the rest of the joints of the arm. The arms are current controlled and backdriveable, meaning they can be moved by external
forces regardless if there is current in the motors to control the joints \cite{30}. To hold the object with the other agents, only the right arm is used.

![PR2 Robot from Willow Garage \cite{31}.](image)

### Software

The software on the PR2 robot is based on ROS (Robot Operating System). Therefore all software created in this project is suitable to run on ROS. ROS is an open-source, meta-operating system for any robot. It provides the services expected from an operating system, including hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. It also provides tools and libraries for obtaining, building, writing, and running code across multiple computers \cite{32}.

Two relevant elements of ROS are \textit{nodes} and \textit{topics}. Nodes are executable programs that can run parallel to each other and that can communicate with each other through a topic by transmitting or "publishing" a message. One or more nodes that wish to receive that message should "subscribe" to the topic where the message is being published. If there is a mention of any of these two concepts in the following sections of the report, please refer to this paragraph for comprehension. For more information about nodes and topics refer to \cite{33}.

#### 3.2.2 The sensor

The PR2 at KTH has one force/torque sensor located at the wrist of each arm. The sensor is an ATI Mini 45 force/torque transducer that measures outputting forces and torques from all three Cartesian coordinates \((x, y, \text{and } z)\) \cite{34}, see Figure \ref{fig:3.4}.

The sensor measurements are available on a topic in ROS. The topic provides the measured wrench in the three axes in SI units at a frequency of 1 kHz. Because
3.2. PHYSICAL SYSTEM

Figure 3.4. Right arm of the PR2 with the ATI Mini45 force/torque sensor at the wrist.

the data from the topic needs processing before using it to solve the problem at hand, the measurements obtained directly from this topic will be denoted as 'raw' measurements from here on despite the fact that they are not the raw output of the sensor.

Sensor Calibration

There are two problems with the sensor measurements that need to be addressed before using the data:

1. The sensor presents readings when there are no external forces or torques being applied to it. This is partially because the force-torque itself has an offset [35], and because of the weight of the gripper attached to the sensor.

2. The sensor frame is rotated with respect to the desired frame, previously defined as frame \( \{s\} \). This is because of how the sensor is physically mounted on the arm of the PR2, see Figure 3.5. As a result, applied forces and torques are mapped to an undesired frame, e.g., a vertical force applied on the gripper will be measured in two axes instead of the vertical axis.

The first problem to address was to remove the offset. The aim was to have a reading of zero for the forces and torques in all directions when there were no external wrenches applied to the gripper. To do this, the first 1000 samples were averaged and subtracted from the readings of each axis of the force and torque. If the measurements become zero during a time when the robot is holding the object and it remains flat in the X-Z plane, the wrench that is generated by the weight of the object is removed. Thus, the only readings present are the ones produced by the agent/agents.

To solve the second problem, the rotation between the sensor frame and the desired frame needed to be found. The sensor frame is shown in Figure 3.5(a), and the desired frame is shown in Figure 3.5(b). In ROS, it is easy to obtain
transformations between frames if they belong to the same simulated system. The PR2 used already has all of its components in simulation and therefore it was trivial to find the rotation between the two frames.

The transformed wrenches will be referred to as $W_s$, and they will be used as an input of the proposed solution. The transformed wrench was transmitted with a frequency of 1 kHz.

### 3.2.3 The object of manipulation

The object used for the research project is a rectangle of cardboard shown in Figure 3.6. There are four highlighted points that will be used. The point named *gripper*, as the name says, is where the right arm of the PR2 will be grasping the cardboard throughout the whole research project. The other three points, highlighted in blue, are points that the agents can grasp.

![Figure 3.5. Illustration of the physical sensor frame and the desired sensor frame. (a) The actual sensor frame; (b) The desired sensor frame.](image)

![Figure 3.6. The cardboard used for manipulation.](image)
3.3 PROPOSED SOLUTION

3.3 Proposed solution

The problem at hand can be divided into two subproblems: a) estimation of the grasping point, b) tracking the current state of the performed sequence. To do this, a transition sequence must be defined previously.

The proposed solution to the problem involves three major elements presented in Figure 3.7. First, the predefined sequence must be performed by the agents, which will produce specific wrenches $W_s$ related to a single configuration of grasping points. The relationship between torques and forces in (3.1) is then used to estimate the grasping point $r_\ast$ using a Kalman filter. To be able to make sense of the transition sequence, an HMM is used to decode the sequence using $r_\ast$ as the input. Where a change from the initial state to the next means knowing that there was a change in the number of agents, and identifying the final step of the sequence means knowing how many agents are holding the object after the change has happened, i.e. knowing $\Lambda$.

![Figure 3.7. The three components of the proposed solution to the problem.](image)

3.3.1 The transition sequence

To create a transition sequence, first, the initial and final scenario was chosen. The initial was chosen as one human holding the cardboard along with the robot and the final scenario as two humans holding the cardboard along with the robot. These two scenarios are shown in Figure 3.8(a) and 3.8(d) respectively.

Assumptions

The next step to follow was to establish how the transition would happen. The following constraints were established with the purpose of allowing disambiguation of the grasping point and providing enough time for the estimator to converge:

- The agents grasp one point of the object using one hand.
- The grasping points lie in the X-Z plane. In other words, the task involves only planar manipulation.
• A change is defined as the addition or subtraction of an agent to the current configuration.

• Only one change can occur at each step, and the agents remain in the same step for the time period it takes the estimator to converge.

• The least amount of agents holding the object is one, meaning the robot will not be left alone holding the object.

The steps in the sequence

A total of four steps were defined as the transition sequence. The human holding the cardboard in the beginning of the sequence will be referred to as Agent 1, and the human that comes in after step one will be referred to as Agent 2. The names of the points where the agents can hold the cardboard are: middle, right and left, the robot will be holding at the same point throughout the sequence on the point marked as gripper, the location of these four points can be seen in Figure 3.6. The steps of the sequence are shown in Figure 3.8 and described below:

• **Step 1**: Initially, Agent 1 grasps the middle point of the cardboard. Hence, frame \{h_1\} has its origin at the middle point.

• **Step 2**: Agent 2 grasps the right point on the cardboard, i.e., the origin of frame \{h_2\} is at the right point. In this step, both agents hold the object.

• **Step 3**: Consequently, Agent 1 must stop holding the middle point to be able to relocate. In this step, only Agent 2 is holding the object along with the robot.

• **Step 4**: Finally, Agent 1 holds the object again at the left point on the cardboard. At this point both agents are grasping the object, the frame \{h_1\} has its origin at the left point and frame \{h_2\} at the right point.
3.3. PROPOSED SOLUTION

Figure 3.8. Defined transition sequence: (a) Agent 1 begins grasping across from the robot. (b) Agent 2 grasps one side of the board (c) Agent 1 stops grasping. (d) Agent 1 grasps at a point on the opposite side of Agent 2.
This chapter provides background to the Kalman filter and hidden Markov models, which are used to solve the problem at hand. The Kalman filter is described including the way the system is modelled, with the process and the measurement model. The Kalman filter algorithm and its steps are described in detail. Furthermore, to introduce the concept of a hidden Markov model (HMM), a brief definition of a Markov model will be presented. The elements of an HMM will be listed for the discrete case to then define the continuous case of these models. The type of HMM used to solve the problem at hand is a continuous HMM using a mixture of Gaussians.

4.1 Kalman filter

What is the Kalman filter?

The Kalman filter is an algorithm that consists of a recursive set of mathematical equations that estimate the state of a process or system minimizing the mean of the squared error [36]. It was invented by Swerling and Kalman [37] and published in 1960 to filter and predict linear Gaussian systems. It relies on the assumptions that the next state is a linear function of the previous state and that the process and measurement noises are white and Gaussian [37]. It is recursive in that it does not require data to be stored every time a new measurement is taken [38]. It has the advantage of supporting estimations of past, present and future states even when there is no precise model of the system [36].

Usually, the state of a system cannot be measured directly, and it must be inferred from the variables of the system that can be measured [38]. Additionally, it is typically the case that the system is driven by other inputs other than the known controls, and that the relationship between the state and the measurements are known with a certain degree of uncertainty [38]. Furthermore, sensors will provide noisy, biased or inaccurate measurements.
CHAPTER 4. BACKGROUND

The Kalman filter is optimal in the way that it estimates the state of a system with a combination of the process and measurement device dynamics, the statistical description of the system noises, measurement errors, and uncertainty in the dynamic models, and it also includes any information about initial conditions of the state to be estimated [38]. Consequently, the estimate of the state $x$ is represented as a Gaussian probability density function (pdf) rather than a discrete value; it is characterized by a mean $\mu$ and a covariance $\Sigma$ [37]:

$$p(x) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right\}$$  \hspace{1cm} (4.1)

The process model

The Kalman filter assumes that the state transition of the system is linear in its arguments, and that the state at time $t$ evolved from the previous state at time $t-1$ following the model:

$$x_t = A_t x_{t-1} + B_t u_t + w_t$$  \hspace{1cm} (4.2)

Where $x_t$ and $x_{t-1}$ are state vectors and $u_t$ is the control input vector. The state and control vectors are vertical and have size $n$ and $m$ respectively. $A_t$ is the state transition matrix of dimension $n \times n$, and $B_t$ is the control input matrix of size $n \times m$. The random variable $w_t$ is a vector that models the uncertainty introduced by the state transition. It has the same dimensionality as the state vector and is assumed independent, white and with normal probability distribution $p(w) \sim N(0, Q)$ [36]. Matrix $Q$ is a symmetric, positive-semidefinite, quadratic matrix of size $n \times n$.

The measurement model

The Kalman filter also assumes that observations are linear functions of the state with added Gaussian noise according to:

$$z_t = C_t x_t + v_t$$  \hspace{1cm} (4.3)

Where $z_t$ is the measurement vector of size $k$, $C_t$ is a matrix of size $k \times n$ and the vector $v_t$ with size $k$ describes the measurement noise, it is assumed independent, white and with normal probability distribution $p(v) \sim N(0, R)$ [36]. Matrix $R$ is a symmetric, positive-semidefinite, quadratic matrix of size $k \times k$.

4.1.1 The Kalman filter algorithm

The Kalman filter provides an estimation of the mean $\mu_t$ and the covariance matrix $\Sigma_t$ that describe the state $x_t$ by a Gaussian pdf stated in (4.1). The mean $\mu_t$ is a vector with the same dimensionality as $x_t$, and $\Sigma_t$ is a quadratic matrix that is
symmetric and positive-semidefinite with dimensionality of the state $x_t$ squared.

The Kalman filter algorithm is shown in Table 4.1. Its inputs are the estimated mean and covariance matrix at the previous time step: $\mu_{t-1}$ and $\Sigma_{t-1}$, as well as the control input and measurement information at the current time step: $u_t$ and $z_t$. The outputs are the estimated mean vector and covariance matrix at the current time step: $\mu_t$ and $\Sigma_t$.

Algorithm 1: Kalman filter

\begin{align*}
\text{Input: } & \mu_{t-1}, \Sigma_{t-1}, u_t, z_t \\
1 & \tilde{\mu}_t = A_t \mu_{t-1} + B_t u_t \\
2 & \tilde{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t \\
3 & K_t = \tilde{\Sigma}_t C_t^T (C_t \tilde{\Sigma}_t C_t^T + R_t)^{-1} \\
4 & \mu_t = \tilde{\mu}_t + K_t (z_t - C_t \tilde{\mu}_t) \\
5 & \Sigma_t = (I - K_t C_t) \tilde{\Sigma}_t
\end{align*}

Output: $\mu_t, \Sigma_t$

Table 4.1. Kalman filter algorithm

The Kalman filter algorithm can be divided in two steps: \textit{prediction} and \textit{measurement update}. The prediction step comprises lines 1 and 2 of the algorithm in Table 4.1. In this step, a prediction of the mean, $\tilde{\mu}_t$, and covariance matrix, $\tilde{\Sigma}_t$, are calculated using the process model parameters from (4.2). The measurement update step comprises lines 3 to 5, where the current measurements are combined with the predicted values $\tilde{\mu}_t$ and $\tilde{\Sigma}_t$, i.e., the current measurements update the values from the prediction step. Line 3 computes the variable $K_t$, called the \textit{Kalman gain}. The Kalman gain specifies the degree to which the measurements are incorporated in the new and final estimate of the mean, $\mu_t$, and the covariance matrix, $\Sigma_t$. These calculations are shown in lines 5 and 6, where the update is a portion of the discrepancy between the actual measurements and the predicted measurements given by the measurement model in (4.3).

4.2 Hidden Markov Models

4.2.1 Discrete Hidden Markov Models

What is a Markov model?

Consider a system that can be modelled as being in one of a finite number of states $N$, at any time $t$. Consider also that this system undergoes a state transition at
each time step. These transitions occur according to a set of probabilities associated with each state, in some cases with the possibility to transition back to the same state [39].

![Figure 4.1. Example of a Markov chain with three states and all possible state transitions with probability $a_{ij}$.](image)

An example of a Markov chain is presented in Figure 4.1. In this example, the system can be described by being in any of three states: $S_1$, $S_2$ or $S_3$. The system can transition between these three states, including staying in the same state according to a set of state transition probabilities.

To describe the Markov chain, we will consider the case of a discrete Markov chain of first order which gives a probabilistic description that considers only the current state $q_t$ and the previous state $q_{t-1}$. In this way, the probability of being at any state $S_i$ and transitioning to state $S_j$ can be described as:

$$a_{ij} = P[q_t = S_j|q_{t-1} = S_i], \quad 1 \leq i, j \leq N \quad (4.4)$$

Where:

$$a_{ij} \geq 0, \quad \text{and} \quad \sum_{i=1}^{N} a_{ij} = 1$$

The output of the Markov process is a set of states that are produced at each time step and each state corresponds to a physical event that can be observed. Therefore, we can call this process an observable Markov model [39].

**What is a hidden Markov model?**

There are types of problems where the states or physical events that we wish to determine cannot be directly observed (they are hidden), but there are other events that can be observed and which provide probabilistic information about the 'hidden' states [40]. A *hidden Markov model* (HMM) is defined by Rabiner in [39] as 'a
4.2. HIDDEN MARKOV MODELS

doubly stochastic process with an underlying stochastic process that is not observable, but can only be observed by another set of stochastic processes that produce the sequence of observations.

A general structure of a hidden Markov model is depicted in Figure 4.2. The state sequence of the unobservable Markov process is represented by \( X = \{X_1, X_2, ..., X_T\} \), and the observation sequence produced by the state sequence is denoted by \( O = \{O_1, O_2, ..., O_T\} \).

**Figure 4.2.** A hidden Markov model. The sequence below the dashed line represents the physical events that can be observed. These observations are assumed to be produced by an unobservable Markov process (above the dashed line). \( X_t \) can be any possible state from the set \( S \) and \( O_t \) can be any observation symbol from the set \( V \) for \( 1 \leq t \leq T \).

**Elements of an HMM**

According to Rabiner, a hidden Markov model is characterized by five elements:

1. The number of states in the model \( N \). There are practical applications where the number of states is related to a physical event. The set of individual states is denoted by \( S = \{S_1, S_2, ..., S_N\} \), and the state at time \( t \) as \( q_t \).

2. The number of the different observation symbols per state \( M \). The observation symbols are the physical output of the modelled system. The set of individual symbols is denoted as \( V = \{V_1, V_2, ..., V_M\} \).

3. The state transition probability distribution \( A = \{a_{ij}\} \), where:

\[
a_{ij} = P[q_{t+1} = S_j | q_t = S_i], \quad 1 \leq i, j \leq N \tag{4.5}
\]
4. The observation symbol probability distribution $B = \{b_j(k)\}$ where:

$$b_j(k) = P[V_k \text{ at } t|q_t = S_j], \quad 1 \leq j \leq N, \quad 1 \leq k \leq M$$  \hspace{1cm} (4.6)

5. The initial state distribution $\pi = \{\pi_i\}$ where:

$$\pi_i = P[q_1 = S_i], \quad 1 \leq i \leq N$$  \hspace{1cm} (4.7)

To specify an HMM completely, the elements $N, M, A, B$ and $\pi$ should be specified. For simplicity, we will denote a complete parameter set of an HMM by:

$$\lambda = (A, B, \pi)$$  \hspace{1cm} (4.8)

The transitions between states in HMMs are given by the connections between them. Some HMMs have connections that make it possible to reach any possible state from the current state, such is the case presented in Figure 4.1. In this case, the state transition probability $a_{ij}$ is greater than 0 for all $i, j$. Moreover, HMMs can have interconnections such that not every state can be reached from a specific state. In this case, we have $a_{ij} = 0$ for one or more $(i, j)$ pairs.

**The three problems**

To be able to use HMMs for real-world applications, it is important to know the problems that can be solved by these models. There are three fundamental problems addressed by HMMs described below as stated in [39].

**Problem 1:**

Given the model $\lambda = (A, B, \pi)$ and an observation sequence $O = O_1 \ O_2 \ldots \ O_T$, we would like to efficiently compute the probability of the observation sequence given the model, i.e. $P(O|\lambda)$. Calculating this probability can be useful when choosing between different models, the higher the $P(O|\lambda)$, the better the model matches the observations.

**Problem 2:**

Given the model $\lambda = (A, B, \pi)$ and an observation sequence $O = O_1 \ O_2 \ldots \ O_T$, we would like to choose the corresponding sequence $X = X_1 \ X_2 \ldots \ X_T$ that is optimal in a meaningful sense. In other words, we would like to provide the sequence of states that best explains the observations. By doing this, we could say we are uncovering hidden part of the HMM [40].
4.2. HIDDEN MARKOV MODELS

Problem 3:

Given an observation sequence $O = O_1 O_2 ... O_T$ and the model parameters $N$ and $M$, we would like to find the model parameters $\lambda = (A, B, \pi)$ that maximize the probability $P(O|\lambda)$. This is done by using an observation sequence, $O$, to train or fit the model to the observations.

4.2.2 Continuous Hidden Markov Models

There are applications in the real world, where the observations provided by the system we wish to model are continuous instead of a finite set of discrete symbols. In such cases, we can no longer use a discrete probability density within each state of the model, but we can use continuous observation densities [39]. The most common way to represent the form of the probability density function (pdf) over the observations is to use a finite mixture of density functions of the form:

$$b_j(O) = \sum_{m=1}^{M} c_{jm} \mathcal{N}[O, \mu_{jm}, U_{jm}]$$

(4.9)

where $O$ is the vector being modelled, $c_{jm}$ is the mixture coefficient for the $m$–th mixture in state $j$. The function $\mathcal{N}$ is any log-concave or elliptically symmetric density with mean vector $\mu_{jm}$ and covariance matrix $U_{jm}$. The mixture gains or weights, $c_{jm}$, satisfy the stochastic constraints:

$$c_{jm} \geq 0, \quad 1 \leq j \leq N, \quad 1 \leq m \leq M$$

$$\sum_{m=1}^{M} c_{jm} = 1, \quad 1 \leq j \leq N$$

(4.10)

It is worth mentioning two advantages with using the pdf presented in (4.9). One is that it makes it possible to reestimate the parameters that describe it consistently during training. The other advantage is that it can be used in a wide range of applications since it can approximate arbitrarily, closely, any finite continuous density function [39]. A common function that is used as the pdf $\mathcal{N}[O, \mu_{jm}, U_{jm}]$ is a Gaussian density. A $d$-dimensional Gaussian pdf is defined as [24]:

$$\mathcal{G}(\mu, \Sigma)(x) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\{-\frac{1}{2}(x - \mu)^{T}\Sigma^{-1}(x - \mu)\}$$

(4.11)
The pdf over the observations as a mixture of $M$ Gaussians for each state $j$ can be written as:

$$b_j(O) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm})(O), \quad 1 \leq j \leq N \quad (4.12)$$
Chapter 5

Grasping point estimation

In this chapter, the implementation of the Kalman filter to estimate the grasping point of the agents is described. The input of this estimator is the wrench measured at the sensor. The implementation can be divided into two parts: choosing the model and tuning the parameters of the Kalman filter (process and measurement noise). The way the model is chosen is explained as well as how the estimator parameters were chosen and the criteria for choosing them.

5.1 Kalman filter implementation

5.1.1 Choosing the model

Let us recall the two equations from Chapter 4 that the Kalman Filter assumes the system or process is governed by:

1. The process model, (4.2):
   \[
   x_t = A_t x_{t-1} + B_t u_t + w_t
   \]

2. The measurement model, (4.3):
   \[
   z_t = C_t x_t + v_t
   \]

Defining the process model

The first step was to define what will be the state vector \( x_t \) to be estimated. As discussed previously, it is desired to determine the distance between the origins of the frames \( \{h_1\} \) and \( \{s\} \), i.e. vector \( r_s \):

\[
\begin{bmatrix}
  r_{sx} \\
  r_{sy} \\
  r_{sz}
\end{bmatrix}
\]

\( x_t = \begin{bmatrix}
  r_{sx} \\
  r_{sy} \\
  r_{sz}
\end{bmatrix} \)
CHAPTER 5. GRASPING POINT ESTIMATION

It is assumed that the state will only be updated by measurements provided by the sensor. Therefore the process model is chosen so that the current state remains the same as the previous state with some added Gaussian noise and no additional input is considered. For some applications, matrix $A_t$ may change overtime, in this specific case, it is defined as constant throughout the estimation. Thus, (4.2) becomes:

$$x_t = A x_{t-1} + w_t$$ (5.1)

and matrix $A$ is defined as:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ (5.2)

As defined in Chapter 4, $w_t$ is a random variable with normal probability distribution:

$$p(w) \sim N(0, Q)$$

The covariance matrix $Q$ is:

$$Q = \begin{bmatrix} \sigma_{rx}^2 & 0 & 0 \\ 0 & \sigma_{ry}^2 & 0 \\ 0 & 0 & \sigma_{rz}^2 \end{bmatrix}$$

Where $\sigma_{rx}$, $\sigma_{ry}$, and $\sigma_{rz}$ are the process noise parameters for each element of the state vector. For simplicity, it is assumed that:

$$\sigma_{rx} = \sigma_{ry} = \sigma_{rz} = \sigma_r$$ (5.3)

Which gives:

$$Q = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_r^2 & 0 \\ 0 & 0 & \sigma_r^2 \end{bmatrix}$$ (5.4)

Defining the measurement model

To update the state vector with measurements, the relationship between what can actually be measured and the state should be established. This relationship has been defined in (3.1):

$$\tau_s = r_s \times F_s$$
5.1. KALMAN FILTER IMPLEMENTATION

However, this relationship should be expressed in matrix form to fit the measurement model \([4.3]\). A cross product can be represented in matrix form in the following way \([41]\):

\[ \mathbf{a} \times \mathbf{b} = S(\mathbf{a})\mathbf{b} = -S(\mathbf{b})\mathbf{a} \tag{5.5} \]

where \(S(\bullet)\) refers to the skew-symmetric matrix of the vector in parenthesis. The skew-symmetric matrix of vector \(\mathbf{a}\) is:

\[
S(\mathbf{a}) = \begin{bmatrix}
0 & -a_3 & a_2 \\
a_3 & 0 & -a_1 \\
-a_2 & a_1 & 0
\end{bmatrix} \tag{5.6}
\]

Using \((5.5)\) the relationship in \((3.1)\) can be rewritten as:

\[
\tau_s = \mathbf{r}_s \times \mathbf{F}_s = -S(\mathbf{F}_s)\mathbf{r}_s \tag{5.7}
\]

Hence, the measurement vector is defined as \(z_t = \tau_s\) at time \(t\):

\[
z_t = \begin{bmatrix}
\tau_{szt} \\
\tau_{syt} \\
\tau_{szt}
\end{bmatrix} \tag{5.8}
\]

and \(C_t = -S(\mathbf{F}_s)\) at time \(t\). Using \((5.6)\), matrix \(C_t\) is defined as:

\[
C_t = \begin{bmatrix}
0 & F_{szt} & -F_{syt} \\
-F_{szt} & 0 & F_{szt} \\
F_{syt} & -F_{szt} & 0
\end{bmatrix} \tag{5.9}
\]

Because \(C_t\) gets its values from the measurements, it is clear that this matrix will change over time.

Finally, the parameters of the added noise of the measurement model were defined. The random variable \(v_t\) has normal probability distribution:

\[
p(v) \sim N(0, R)
\]

where the covariance matrix \(R\) is:

\[
R = \begin{bmatrix}
\sigma^2_{mx} & 0 & 0 \\
0 & \sigma^2_{my} & 0 \\
0 & 0 & \sigma^2_{mz}
\end{bmatrix} \tag{5.10}
\]
with $\sigma_{m_x}$, $\sigma_{m_y}$, and $\sigma_{m_z}$ being the measurement noise parameters for each element of the measurement vector. For simplicity it is assumed that:

$$
\sigma_{m_x} = \sigma_{m_y} = \sigma_{m_z} = \sigma_m \tag{5.11}
$$

Hence, matrix $R$ becomes:

$$
R = \begin{bmatrix}
\sigma_m^2 & 0 & 0 \\
0 & \sigma_m^2 & 0 \\
0 & 0 & \sigma_m^2
\end{bmatrix} \tag{5.12}
$$

Testing the estimator

To evaluate the performance of the estimator, a scenario with a known grasping point was chosen. This scenario is shown in Figure 5.1 where $r_s = [0.29, 0, 0.18]^T$ in meters. The Kalman filter algorithm was implemented in ROS as a node that publishes the output of the estimator with a frequency of 1 kHz. An initial guess of the state and the covariance matrix should be provided to start running the algorithm. They were defined as:

$$
\hat{r}_{s_{t-1}} = \begin{bmatrix}0 & 0 & 0\end{bmatrix}^T \tag{5.13}
$$

$$
\Sigma_{t-1} = \begin{bmatrix}1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
$$

![Figure 5.1. Testing scenario of the Kalman filter implementation, where vector $r_s$ is known.](image)
5.1. KALMAN FILTER IMPLEMENTATION

For the first evaluation of the Kalman filter, no real tuning of the parameters was done, a couple of values were given to the system to test the estimator. During the tests, the person grasping the known point had to exert force towards the robot for the estimator to converge to the correct values. The results of the estimator for two different values of the measurement noise parameters are shown in Figure 5.2. The dotted line represents the ground truth of the grasping point. At first glance, one can see that the bigger the measurement noise, the smoother the signal. The reason for this will be explained in the next section. For now, the evaluation of the model is to be described.

![Graphs showing estimation results of Kalman filter with different noise parameters](image)

Figure 5.2. Estimation results of Kalman filter with two different values of $\sigma_m$, where the dotted line represents the ground truth.

To evaluate the estimator for each set of parameters, the metrics considered were the response time, the mean and standard deviation of the settled value and the error mean of the steady state error: $r_s - \mu_{r_s}$. The results of these metrics for both cases are presented in Table 5.1. Because the estimator provides a smoother response for the second set of noise parameters, the response time is also higher. The values of the mean and standard deviation of the estimation, and the error are presented in the three components: $x$, $y$ and $z$. It is clear from the plots in Figure 5.2(a) that the spread of the values for the first set of parameters is higher. As for the error, the difference is not significant compared to the difference in response times.
CHAPTER 5. GRASPING POINT ESTIMATION

The choice of parameters will depend on the application and in this research, accuracy is not the priority. The estimator is expected to provide a fast response to changes, without getting too much variation, i.e., get rid of some of the noise. In the results shown in Table 5.1, we can see that the estimator converges in the right direction: towards the grasping point.

The choice of parameters will depend on the application and in this research, accuracy is not the priority. The estimator is expected to provide a fast response to changes, without getting too much variation, i.e., get rid of some of the noise. In the results shown in Table 5.1, we can see that the estimator converges in the right direction: towards the grasping point.

### 5.1.2 Tuning the Kalman filter

Tuning the Kalman filter refers to finding the parameters of the process noise, $\sigma_r$, and the measurement noise, $\sigma_m$, that gives the best estimate of the state. Determining what is the best estimate depends on what is expected of the estimator, e.g., a smooth estimate in contrast to a fast response. It is important to note that, although the names of the parameters indicate they are the noise of the physical system or the noise of the sensor, in this application this is not the case, since the noise of the measurements does not only come from the sensor, but also from the manipulation of the object by the humans and the control action of the robot if the object is pressed to hard.

To tune the Kalman Filter, one must first understand what the two parameters to be tuned mean. The process noise $\sigma_r$ can be seen as how much the process model is trusted. The same concept applies to the measurement noise $\sigma_m$. Therefore, bigger noises mean less trust in the models. Recalling the Kalman gain concept in Chapter 4, we know that the smaller the Kalman gain, the less the measurements will be incorporated in the estimation. Recalling its calculation from line 3 of the algorithm in Table 4.1:

$$K_t = \frac{\overline{\Sigma_t}C^T_t}{(C_t\overline{\Sigma_t}C^T_t + R_t)}$$

we see that the bigger the measurement noise is with respect to the process noise, the smaller the Kalman gain will be.

<table>
<thead>
<tr>
<th>Noise parameters</th>
<th>Response time (s)</th>
<th>$\mu_{r_s}$ (cm)</th>
<th>$\sigma_{\mu_{r_s}}$ (cm)</th>
<th>$r_s - \mu_{r_s}$ (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_r = 0.01$</td>
<td>5</td>
<td>x</td>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>$\sigma_m = 10$</td>
<td></td>
<td>25.4</td>
<td>1.5</td>
<td>16.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\sigma_r = 0.01$</td>
<td>10.7</td>
<td>x</td>
<td>y</td>
<td>z</td>
</tr>
<tr>
<td>$\sigma_m = 50$</td>
<td></td>
<td>25.7</td>
<td>0.7</td>
<td>14.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1. Evaluation of estimator for a single grasping point for two values of $\sigma_m$
5.1. KALMAN FILTER IMPLEMENTATION

Performing sequence for tuning

To tune the estimator, data was collected from the sensor while two agents performed the transition sequence. The way this was carried out is described below:

- The robot was set to hold the cardboard with its right arm at the grip point, see Figure 3.6.
- The cardboard was made completely flat without applying any external forces to perform calibration.
- The wrench $W_s$ was recorded while the two agents performed the defined transition sequence, see Figure 3.8. They performed the changes every 5 seconds applying force towards the robot while grasping.

Note: The amount of time between transitions was set arbitrarily, and after running the data on the estimator, it would be determined if 5 seconds was enough time for the estimator to converge.

Choosing the parameters

The transition sequence was performed several times to obtain different data sets to tune the Kalman filter with. In the data collected, the sensor is not the only reason there can be fluctuations in the measurements, the measurements also reflect how the agents apply forces. For this reason, there is no real way to determine what values to start with, so different pairs of values were used on the estimator for the data collected. The values chosen for the process noise $\sigma_r$ and measurement noise $\sigma_m$ are shown in Table 5.2; all combinations between $\sigma_r$ and $\sigma_m$ were evaluated on the estimator.

<table>
<thead>
<tr>
<th>$\sigma_r$</th>
<th>$\sigma_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\sigma_r$</th>
<th>$\sigma_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1000</td>
</tr>
<tr>
<td>50</td>
<td>2000</td>
</tr>
<tr>
<td>100</td>
<td>3000</td>
</tr>
<tr>
<td>200</td>
<td>4000</td>
</tr>
</tbody>
</table>

Table 5.2. Noise parameters used for tuning

From all the combinations, only certain values produced meaningful results. These values were tried again on different data sets, to not overfit the noise parameters to a specific data set.

Results of the Kalman filter for different parameters

The results for values that provided meaningful results from the estimator are shown in Figure 5.3. The initial value of the grasping point for the estimator was changed to: $\hat{r}_{s_{-1}} = \begin{bmatrix} 0 & 0 & 0.5 \end{bmatrix}^T$.
CHAPTER 5. GRASPING POINT ESTIMATION

(a) $\sigma_r = 0.01$
$\sigma_m = 10$

(b) $\sigma_r = 0.01$
$\sigma_m = 30$

(c) $\sigma_r = 0.01$
$\sigma_m = 50$

(d) $\sigma_r = 0.01$
$\sigma_m = 100$

Figure 5.3. Results of the Kalman filter for the transition sequence for different noise parameters.
5.1. KALMAN FILTER IMPLEMENTATION

Analysing the plots in Figure 5.3, the results in (a) are too noisy which might give the impression of a change when there isn’t one. The results in (d) are too smooth, which makes it slower to reflect the change in the signal. The results in (b) are smoother than in (a) but there are still some visible fluctuations due to noise. The results in (c) are smoother than (a) and (b), yet they still show the high peak in the $z$ component, unlike the result in (d). Therefore, the parameters chosen for the estimator are:

\[
\sigma_r = 0.01 \\
\sigma_m = 50
\]

These parameters are used for estimating the grasping point that will be tracked using a hidden Markov model, which will be explained in the next chapter.
Chapter 6

Tracking the number of agents

In this chapter, the way to track the number of agents is described. For this, the output of the previously described grasping point estimator will be the input to a continuous hidden Markov model. It is desired to link each step of the sequence with an observation to determine how many agents are grasping the object, with this, if there is a jump from the initial state to the subsequent state, there is an indication that a change has occurred, and once the last state is reached, it will be known that there are 2 agents holding and that the transition between 1 and 2 agents has terminated.

6.1 Hidden Markov Model Implementation

One of the three fundamental problems that HMMs solve, problem 2 described in Chapter 4, is determining which state of the possible hidden states is more likely to have produced a given observation having a set of model parameters. Since the sequence has already been determined, there is a good initial guess of what the model should look like, but there is still the need to link the observations to the steps in the sequence. For that, the third problem described in Chapter 4 should be tackled first. In that case, it is needed to obtain training data to adjust the model to the observations. The implementation of the continuous HMM was obtained from [42], which is based on Rabiner’s tutorial [39].

6.1.1 Choosing the model

Seeing that the vector $\hat{r}$ is produced in a continuous manner, a continuous hidden Markov model is chosen to represent the transition sequence. Table 6.1 presents a compilation of the notation of the elements needed to describe a continuous HMM. The choice of parameters in the table will be described next.
CHAPTER 6. TRACKING THE NUMBER OF AGENTS

Table 6.1. Notation of elements in a continuous HMM

| $N$ | Number of hidden states in the model |
| $S$ | $\{S_1, S_2, ..., S_N\}$ = Set of possible states |
| $\pi$ | Initial state distribution |
| $A$ | State transition matrix with size $N \times N$ |
| $b_j(O)$ | Probability density function over the observations per state |
| $O$ | $\{O_1, O_2, ..., O_T\}$ = Observation sequence |
| $T$ | Length observation sequence |
| $M$ | Number of mixtures of density functions |
| $X$ | $\{X_1, X_2, ..., X_T\}$ = State sequence |

The states

The number of states were chosen as the number of steps it takes to transition from having one agent grasping to having two, given that it is assumed that each step produces different observations. Hence, $N = 4$ and $S = \{S_0, S_1, S_2, S_3\}$, the association of each hidden state to each step in the sequence is shown in Table 6.2, refer to Section 3.3.1 for the full description of the transition sequence.

<table>
<thead>
<tr>
<th>State</th>
<th>Association</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>Step 1</td>
</tr>
<tr>
<td>$S_1$</td>
<td>Step 2</td>
</tr>
<tr>
<td>$S_2$</td>
<td>Step 3</td>
</tr>
<tr>
<td>$S_3$</td>
<td>Step 4</td>
</tr>
</tbody>
</table>

Table 6.2. Association of hidden states to the steps in the transition sequence.

Considering that the transition sequence always starts in state $S_0$, the initial state distribution is set as:

$$\pi = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \tag{6.1}$$

To make an initial guess of the transition matrix $A$, two things must be considered. First, it is known that the state transitions occur consecutively. Second, each transition happens every 5 seconds, otherwise stated, each state remains the same for 5 seconds. Thus, there is a higher probability to remain in the same state than to transition to the next state. Keeping in mind that each row of the $A$-matrix should add to 1, the state probability distribution is chosen as:
6.1. HIDDEN MARKOV MODEL IMPLEMENTATION

\[
A = \begin{bmatrix}
0.9 & 0.1 & 0 & 0 \\
0 & 0.9 & 0.1 & 0 \\
0 & 0 & 0.9 & 0.1 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]  \hspace{1cm} (6.2)

The observations

The observations are chosen to be two of the three components of vector \( \hat{r}_s \): \( \hat{r}_{sx} \) and \( \hat{r}_{sz} \). The probability density function that will model the observations per state, \( b_j(O) \), is chosen as a single Gaussian given that it is assumed that each step in the sequence gives a somewhat steady point. Recall equation (4.12) that describes the pdf over the observations as a mixture of \( M \) Gaussians from Chapter 4:

\[
b_j(O) = \sum_{m=1}^{M} c_{jm} G(\mu_{jm}, \Sigma_{jm})(O), \hspace{1cm} 1 \leq j \leq N
\]

Considering that \( M = 1 \) and the constraints in (4.10), the mixture coefficient \( c_{jm} = 1 \) for all states. Rewriting (4.12) with (4.11) as a two-dimensional Gaussian (two dimensions because of the two components of \( \hat{r} \)), we have:

\[
b_j(O_t) = \frac{1}{(2\pi|\Sigma_j|)^{1/2}} \exp\left\{-\frac{1}{2}(O_t - \mu_j)^T \Sigma_j^{-1}(O_t - \mu_j)\right\}, \hspace{1cm} 1 \leq j \leq N \hspace{1cm} (6.3)
\]

The length of the observation sequence \( T \) depends on the time the transition takes. There is a limitation with the size of \( T \) in the implementation of the HMM used, especially when used for training. This limit was found empirically as \( T < 500/d \), where \( d \) is the number of features used. Because of the limitation of the length of the observation vectors \( T \), the output of the estimator was sampled to reduce the amount of data that is used as input to the HMM. The signal was sampled with the frequency of 5 Hz. Given that the transition sequence lasts 20 seconds, the observation sequence will have a length of 100 samples, which is less than the limit found. The output of the estimator was sampled for training and for testing the identification of the sequence.

As stated previously, the link between the observations and the hidden states should be found by training. However, initial values should be given before starting to train, therefore, the initial values of the means and covariances are chosen arbitrarily as:
CHAPTER 6. TRACKING THE NUMBER OF AGENTS

\[ \mu_j = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}^T \]  
(6.4)

\[ \Sigma_j = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad 1 \leq j \leq N \]

6.1.2 Training the HMM

In order to train the HMM, three people were asked to produce the training data. It was required that they had no prior knowledge of the tasks they would perform in order to provide a more realistic model.

Training data

Three different pairs were made from these three people. Each pair was asked to perform the transition sequence twice. The first time, the pairs would perform the sequence as if they would grasp normally following a set of instructions. The second time, they would perform the sequence with the same instructions as before, but applying a certain force towards the robot.

The humans will be referenced as Person 1, Person 2 and Person 3, and the pairs formed by them are shown in Table 6.3, where the letter P represents the word Person, e.g., P1 refers to Person 1. In the table, a person is underlined if that person played the role of Agent 1 in the corresponding pair; the role was changed so each person played both roles for both sets of instructions.

<table>
<thead>
<tr>
<th>Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1 - P2</td>
</tr>
<tr>
<td>P1 - P3</td>
</tr>
<tr>
<td>P2 - P3</td>
</tr>
</tbody>
</table>

Table 6.3. Pairs made to perform the transition sequence.

The first time the humans performed the sequence, they were explained the purpose of this research and received the following instructions:

- They were asked to perform the transition sequence which was explained to them and to make it completely clear, the images in Figure 3.8 were shown to them.

- They were told that they would receive instructions on when to change to the next step, and that the changes would be every 5 seconds.
6.1. HIDDEN MARKOV MODEL IMPLEMENTATION

- They were asked to grasp only at the marked points on the cardboard using one hand.
- Finally, they were asked to try to keep the cardboard flat.

The second time they performed the transition sequence, they were asked to keep in mind the same instructions as before and they were asked to also apply "a little bit" of force towards the robot.

**Analysing estimator results**

Since it is assumed that the observation per state can be modelled with a single Gaussian based on the information obtained in the previous chapter, the output of the estimator was analysed to see if the representation would hold.

The results of the estimator for both sets of instructions for each pair are presented below. In Table 6.4, the results for Person 1 and Person 2 after performing the transition sequence without applying force and applying force towards the robot. The results for both sets of instructions for Person 1 and Person 3 are presented in Table 6.5 and for Person 2 and Person 3 are presented in Table 6.6. See Figure 6.1 where the sensor frame is shown and the points where the agents should grasp are indicated. Use it to compare the results of the estimator to where the grasping point estimator should converge to. For clarification, recall the steps of the sequence in Figure 3.8 in Chapter 3.
### Table 6.4. Grasping point estimator results for P1-P2 for two different instructions.

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Estimator result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents did not apply force</td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>Agents applied force</td>
<td><img src="image" alt="Graph" /></td>
</tr>
</tbody>
</table>

**Person 1 and Person 2**

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Estimator result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents did not apply force</td>
<td><img src="image" alt="Graph" /></td>
</tr>
<tr>
<td>Agents applied force</td>
<td><img src="image" alt="Graph" /></td>
</tr>
</tbody>
</table>
6.1. HIDDEN MARKOV MODEL IMPLEMENTATION

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Estimator result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents did not apply force</td>
<td><img src="image" alt="Graphs showing estimator results forAgents did not apply force" /></td>
</tr>
<tr>
<td>Agents applied force</td>
<td><img src="image" alt="Graphs showing estimator results forAgents applied force" /></td>
</tr>
</tbody>
</table>

Table 6.5. Grasping point estimator results for P1-P3 for two different instructions.
CHAPTER 6. TRACKING THE NUMBER OF AGENTS

<table>
<thead>
<tr>
<th>Instructions</th>
<th>Estimator result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agents did not apply force</td>
<td><img src="image" alt="Graphs" /></td>
</tr>
<tr>
<td>Agents applied force</td>
<td><img src="image" alt="Graphs" /></td>
</tr>
</tbody>
</table>

Table 6.6. Grasping point estimator results for P2-P3 for two different instructions.
6.1. HIDDEN MARKOV MODEL IMPLEMENTATION

Looking at the estimator results, it is evident that applying a slight force towards the robot makes the changes more noticeable and the values in between changes more stable. It is also noticeable that, when the agents don’t apply force, the values for $\hat{r}_s_x$ are small and positive, and there are big changes in $\hat{r}_s_y$, while the values for $\hat{r}_s_z$ remain the same for both cases. Because of how the system is set up, the values of $\hat{r}_s_y$ should in fact be negative at least in step 3 of the sequence, where only Agent 2 is grasping the object at the right point of the cardboard. The signal for this step corresponds to the signal between 10 and 15 seconds. Since the agents tried to keep the object flat, having values of up to negative 20cm in $\hat{r}_y$ is not possible, as is the case when agents do not apply force. So, one could say that when applying force, the results of the estimator make more sense. In conclusion, the signals are very different for each pair when force is applied and when it is not applied but the three pairs show somewhat similar signals for each of the cases of applying and not applying force. For this reason, instead of training to obtain one HMM for both cases, the HMM was trained for two different cases, one HMM for cases where a slight force is applied towards the sensor, and one HMM for the case where the agents do not apply any force towards the sensor.

**HMM for agents applying force: HMM$_F$**

The first case to train for was for agents that apply force towards the robot, the parameters obtained from this training will define the HMM for this case: HMM$_F$. The training was done using data from different pairs where two different parameters were obtained. Both parameter sets were tested for all three pairs. The set that better identified the transition sequence was chosen. As input for training and decoding the sequence, the HMM took the sampled output of the estimator. The sampled signal for each pair is presented in Table 6.8. One set of parameters was obtained by training with the three sets of training data, and the other was trained with only two datasets. The two sets of parameters are referred to as:

- $\lambda_{F1}$: Trained with $P1$-$P2$, $P2$-$P3$, $P1$-$P3$
- $\lambda_{F2}$: Trained with $P1$-$P3$, $P2$-$P3$

The reason to exclude the training sequence from Person 1 and Person 2 in $\lambda_{F2}$ was because the output of the estimator for the $x$ component did not present the clear change between step 1 and step 2, therefore it was excluded of the training.

The results of using both sets of parameters to decode the sequence are summarized in Table 6.7, where each of them was evaluated on how many states were identified correctly. A state is considered to be correctly identified if the state matches the step the agents are in at the given sample, excluding the parts of the observation where the estimator has not converged.

45
## CHAPTER 6. TRACKING THE NUMBER OF AGENTS

The plots resulting of the parameter sets \( \lambda_{F_1} \) and \( \lambda_{F_2} \) for each pair are presented in Table 6.9. Each plot has an indication of which state each sample should be, e.g., the first 25 samples should be identified as state 0. The changes will not always happen every 25 samples since there may be a delay between the given instruction and the performed action.

**Table 6.7.** Results of the parameter sets for all three pairs when force is applied towards the sensor by the agents.

<table>
<thead>
<tr>
<th>Parameter set</th>
<th>Pairs</th>
<th>Results</th>
<th>Correct states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P1-P2</td>
<td>✓  x</td>
<td>✓  ✓</td>
</tr>
<tr>
<td>( \lambda_{F_1} )</td>
<td>P1-P3</td>
<td>✓  ✓  x</td>
<td>✓  x</td>
</tr>
<tr>
<td></td>
<td>P2-P3</td>
<td>✓  ✓  x</td>
<td>✓  ✓</td>
</tr>
<tr>
<td></td>
<td>P1-P2</td>
<td>✓  x</td>
<td>✓  ✓</td>
</tr>
<tr>
<td>( \lambda_{F_2} )</td>
<td>P1-P3</td>
<td>✓  ✓  ✓</td>
<td>✓  ✓</td>
</tr>
<tr>
<td></td>
<td>P2-P3</td>
<td>✓  ✓  ✓</td>
<td>✓  ✓</td>
</tr>
</tbody>
</table>
6.1. HIDDEN MARKOV MODEL IMPLEMENTATION

<table>
<thead>
<tr>
<th>Pair</th>
<th>Input of HMMF</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1-P2</td>
<td><img src="image" alt="Sampled r₁" /></td>
</tr>
<tr>
<td>P1-P3</td>
<td><img src="image" alt="Sampled r₂" /></td>
</tr>
<tr>
<td>P2-P3</td>
<td><img src="image" alt="Sampled r₃" /></td>
</tr>
</tbody>
</table>

Table 6.8. Sampled signal of the estimator for each pair.
CHAPTER 6. TRACKING THE NUMBER OF AGENTS

Output of HMM$_F$ using $\lambda_{F_1}$

<table>
<thead>
<tr>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
</tr>
<tr>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_2$</td>
</tr>
<tr>
<td>$S_3$</td>
</tr>
</tbody>
</table>

Output sequence of HMM$_F$ for both parameter sets.

Table 6.9.

Output of HMM$_F$ using $\lambda_{F_2}$

<table>
<thead>
<tr>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
</tr>
<tr>
<td>$S_1$</td>
</tr>
<tr>
<td>$S_2$</td>
</tr>
<tr>
<td>$S_3$</td>
</tr>
</tbody>
</table>
6.1. HIDDEN MARKOV MODEL IMPLEMENTATION

The chosen set of parameters was $\lambda_{F_2}$ because it identified correctly two of the three sequences, and was able to identify the final state correctly all three times. The parameters for HMM$_F$ are:

$$\pi = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix}
0.9665 & 0.03348 & 0 & 0 \\
0 & 0.9577 & 0.0423 & 0 \\
0 & 0 & 0.9629 & 0.0371 \\
0 & 0 & 0 & 1
\end{bmatrix}$$

<table>
<thead>
<tr>
<th>State</th>
<th>$\mu$</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$\begin{bmatrix} -0.0139 \ 0.4994 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0104 &amp; 0.0 \ 0.0 &amp; 0.01 \end{bmatrix}$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$\begin{bmatrix} -0.1055 \ 0.4937 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0107 &amp; 0.0001 \ 0.0001 &amp; 0.01 \end{bmatrix}$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$\begin{bmatrix} -0.2179 \ 0.4584 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0109 &amp; 0.0003 \ 0.0003 &amp; 0.01 \end{bmatrix}$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$\begin{bmatrix} -0.0597 \ 0.49 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0109 &amp; 0.0001 \ 0.0001 &amp; 0.01 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

Table 6.10. Observation pdf parameters of $\lambda_{F_2}$. 
HMM for agents not applying force: $\text{HMM}_{NF}$

From the results of the estimator in Tables 6.4, 6.5 and 6.6, we see that there are more visible changes in the $y$ component than the $z$ component of vector $\hat{r}_s$. Therefore, this HMM will take as input the $x$ and $y$ components after sampling. The sampled signal for each pair is presented in Table 6.8.

There were four different parameter sets obtained for the case where the agents do not apply any force towards the sensor. These were obtained by training with different datasets and each of the set of parameters was tested for all three pairs. The goal of testing them was to see how well each parameter set of HMM$_{NF}$ could identify the states correctly in the training data. The four sets of parameters are referred as:

- $\lambda_{NF1}$: Trained with P1-P2
- $\lambda_{NF2}$: Trained with P1-P2, P2-P3
- $\lambda_{NF3}$: Trained with P1-P2, P2-P3, P1-P3
- $\lambda_{NF4}$: Trained with P1-P2, P1-P3

The results of these are presented in Table 6.11, where each parameter set will be evaluated on how many states it can identify correctly.

<table>
<thead>
<tr>
<th>Parameter set</th>
<th>Pairs</th>
<th>Results</th>
<th>Correct states</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$S_0$</td>
<td>$S_1$</td>
</tr>
<tr>
<td>$\lambda_{NF1}$</td>
<td>P1-P2</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>P1-P3</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>P2-P3</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\lambda_{NF2}$</td>
<td>P1-P2</td>
<td>✓</td>
<td>x</td>
</tr>
<tr>
<td>P1-P3</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>P2-P3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$\lambda_{NF3}$</td>
<td>P1-P2</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>P1-P3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>P2-P3</td>
<td>✓</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>$\lambda_{NF4}$</td>
<td>P1-P2</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>P1-P3</td>
<td></td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>P2-P3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 6.11. Results of the parameter sets for all three pairs when no force is applied towards the sensor by the agents.
### Table 6.12. Sampled signal of the estimator for each pair.
CHAPTER 6. TRACKING THE NUMBER OF AGENTS

<table>
<thead>
<tr>
<th>Output of HMM$<em>{NF}$ using $\lambda</em>{NF}$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P1-P2</td>
<td><img src="image1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>P1-P3</td>
<td><img src="image2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>P2-P3</td>
<td><img src="image3.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

**Table 6.13.** Output sequence of $\lambda_{NF1}$ and $\lambda_{NF2}$. 

52
### Output of HMM\(_{NF}\) using \(\lambda_{NF,3}\)

<table>
<thead>
<tr>
<th>P1-P2</th>
<th>Estimated sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(S_1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P1-P3</th>
<th>Estimated sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(S_1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P2-P3</th>
<th>Estimated sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(S_1)</td>
</tr>
</tbody>
</table>

### Output of HMM\(_{NF}\) using \(\lambda_{NF,4}\)

<table>
<thead>
<tr>
<th>P1-P2</th>
<th>Estimated sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(S_1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P1-P3</th>
<th>Estimated sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(S_1)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P2-P3</th>
<th>Estimated sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S_0)</td>
<td>(S_1)</td>
</tr>
</tbody>
</table>

Table 6.14. Output sequence of \(\lambda_{NF,3}\) and \(\lambda_{NF,4}\).
The chosen set of parameters was $\lambda_{NF_3}$ because with it, the HMM identified all states for one of the pairs, and it was able to identify the final state on another pair. Although with $\lambda_{NF_2}$ the HMM also identified all states for one pair, it failed to identify any state for another pair. Thus, the model parameters for HMM$_{NF}$ are:

$$\pi = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

$$A = \begin{bmatrix} 0.9676 & 0.0324 & 0 & 0 \\ 0 & 0.9512 & 0.0488 & 0 \\ 0 & 0 & 0.9549 & 0.4512 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<table>
<thead>
<tr>
<th>State</th>
<th>$\mu$</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_0$</td>
<td>$\begin{bmatrix} 0.05 \ -0.1596 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0102 &amp; -0.0006 \ -0.0006 &amp; 0.0115 \end{bmatrix}$</td>
</tr>
<tr>
<td>$S_1$</td>
<td>$\begin{bmatrix} 0.1232 \ -0.2467 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0108 &amp; -0.0003 \ -0.0006 &amp; 0.0105 \end{bmatrix}$</td>
</tr>
<tr>
<td>$S_2$</td>
<td>$\begin{bmatrix} 0.0908 \ -0.2775 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0106 &amp; 0.0002 \ 0.0002 &amp; 0.0104 \end{bmatrix}$</td>
</tr>
<tr>
<td>$S_3$</td>
<td>$\begin{bmatrix} 0.021 \ -0.2761 \end{bmatrix}$</td>
<td>$\begin{bmatrix} 0.0108 &amp; -0.0002 \ -0.0002 &amp; 0.0104 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

Table 6.15. Observation pdf parameters of $\lambda_{NF_3}$.
Chapter 7

Experiments

The experiments were performed to evaluate the performance of the proposed method. For the experimental tasks, four humans were asked to participate. These were different from the humans involved in producing the training sequences in the previous chapter. Recall, that two HMMs were trained separately: one HMM for the cases where humans apply a slight force towards the robot while grasping the object, HMM$_F$; and another HMM for cases where the humans do not apply any force towards the robot, HMM$_{NF}$. Both of the trained HMMs were tested for the predefined transition sequence, as well as for a deviation of the sequence. This deviation is explained in the following sections. The results of these experimental tasks are presented and explained at the end of this chapter.

7.1 Description of the experiments

7.1.1 Experimental setup

To perform the transition sequence, the PR2 was set up to maintain a fixed position. The right arm of the robot was positioned to grasp the cardboard at the gripper point, the same position used for tuning the system, see Figure 7.1. The height of the gripper was adjusted so the agents were able to grasp in a natural way. The sensor was calibrated before the agents performed each task to avoid any drift that might be present from the force/torque sensor, see Chapter 3 where this procedure is explained in detail.

The parameters for the system used were the ones obtained in Chapters 5 and 6. For the grasping point estimator, the noise parameters of the Kalman filter were $\sigma_r = 0.01$ and $\sigma_m = 50$, which correspond to the process noise and measurement noise respectively. The model used for HMM$_F$ is $\lambda_{F_2}$, shown in Table 6.10 and the model used as HMM$_{NF}$ is model $\lambda_{NF_3}$, shown in Table 6.15.
7.1.2 Experimental tasks

In total, four humans participated in the experiments to perform the experimental tasks. They were split in two groups and each group performed two tasks. The humans will be referenced as Person A, Person B, Person C and Person D in the following sections.

Group 1

The first group was formed by three people out of the four available for the experiments. The people in this group had no prior knowledge of the tasks they would perform. In this group, three different pairs were made and are shown Table 7.1, where the letter $P$ represents the word Person, e.g., $PA$ refers to Person A. In the table, a person is underlined if that person played the role of Agent 1 in the corresponding pair. For Task 2, the humans that took the role as Agent 2 in Task 1 changed their role to Agent 1. The transition sequences of this group will be used to test $HMM_{NF}$.

<table>
<thead>
<tr>
<th>Group 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA - PB</td>
</tr>
<tr>
<td>PA - PC</td>
</tr>
<tr>
<td>PB - PC</td>
</tr>
</tbody>
</table>

Table 7.1. Pairs in group 1

Group 2

The second group was formed by all four people available for the experiments. The fourth person added to group 1 had some knowledge of the research project and knew that forces had to be applied towards the sensor for better performance of the system, therefore, this person only belonged to group 2. From this group, six different pairs were made and they are presented in Table 7.2, where the notation is the same as in Table 7.1. For Task 2, the humans that took the role as Agent 2 in Task 1 changed their role to Agent 1. The transition sequences of this group will be used to test $HMM_{F}$.

<table>
<thead>
<tr>
<th>Group 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>PA - PB</td>
</tr>
<tr>
<td>PA - PC</td>
</tr>
<tr>
<td>PA - PD</td>
</tr>
</tbody>
</table>

Table 7.2. Pairs in group 2.
7.1. DESCRIPTION OF THE EXPERIMENTS

The tasks

Both groups were asked to perform two tasks:

- **Task 1:** Perform the transition sequence described in Chapter 3.

- **Task 2:** Perform a transition sequence similar to task 1. Instead of grasping the *right* and *left* points on the cardboard, the agents were asked to grasp the *right corner* and *left corner* points respectively, see Figure 7.1.

To perform the tasks, both groups were explained the purpose of this research and received the following instructions:

- They were asked to perform the transition sequence which was explained to them and to make it completely clear, the images in Figure 3.8 were shown to them.

- They were told that they would receive instructions on when to change to the next step, and that the changes would be every 5 seconds.

- They were asked to grasp only at the marked points on the cardboard using one hand.

- Finally, they were asked to try to keep the cardboard flat.

![Figure 7.1. The cardboard used for manipulation.](image)

After group 1 performed tasks 1 and 2, group 2 was asked to perform the tasks again except, this time, they were explained where the sensor was, and that they should apply a slight force towards the robot.
CHAPTER 7. EXPERIMENTS

7.2 Results

The results of the experimental tasks will be presented as follows. First, the results of HMM$_F$ for group 2 are presented followed by the result of HMM$_{NF}$ on group 1. The results of the grasping point estimator for both groups and the tasks they performed are presented in Appendix A.

7.2.1 Results for model HMM$_F$

Task 1

The HMM$_F$ will be evaluated on how many states were identified correctly, and if state $S_3$ was identified correctly. A state is considered to be identified correctly if the current identified state matches the current step the agents are in, excluding the parts of the observation where the estimator has not settled. The time it takes the system to detect a state transition will also be evaluated, from the time the change occurred to when it was actually identified by the HMM. The analysis of the results for the six different pairs formed from group 2 are shown in Table 7.3. The times marked as bold are the longest state transition times for each pair, and the time in red shows the absolute longest state transition of all state transitions for all pairs.

<table>
<thead>
<tr>
<th>Pairs</th>
<th>States detected</th>
<th>Detected final state</th>
<th>Reaction time to changes (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) PA - PB</td>
<td>3</td>
<td>✓</td>
<td>$S_0 \rightarrow S_1$</td>
</tr>
<tr>
<td>2) PA - PC</td>
<td>2</td>
<td>x</td>
<td>$S_1 \rightarrow S_2$</td>
</tr>
<tr>
<td>3) PA - PD</td>
<td>4</td>
<td>✓</td>
<td>$S_2 \rightarrow S_3$</td>
</tr>
<tr>
<td>4) PB - PC</td>
<td>4</td>
<td>✓</td>
<td>$S_0 \rightarrow S_1$</td>
</tr>
<tr>
<td>5) PB - PD</td>
<td>4</td>
<td>✓</td>
<td>$S_1 \rightarrow S_2$</td>
</tr>
<tr>
<td>6) PC - PD</td>
<td>4</td>
<td>✓</td>
<td>$S_2 \rightarrow S_3$</td>
</tr>
</tbody>
</table>

Table 7.3. Results of model HMM$_F$ for task 1. (*) The time is taken from $S_0$ to $S_2$.

For this task, the model was able to identify all states correctly 4 times out of 6 (pairs 3-6), and it was able to identify the final state 5 out of 6 times. Overall, the longest time it took the model to detect a transition from one state to the next was 1.21 seconds.

For pair 3, the input and output of HMM$_F$ are shown in Table 7.4. Analysing this case since all states were identified correctly, the change from step 1 to step 2 occurs after sample 25, so after this sample it is expected that the HMM will identify the following states as $S_1$, which according to Table 7.4, happens immediately at sample 26. The transition to step 3 happens after sample 49, and the HMM identifies sample 50 as $S_2$, which corresponds to step 3. Finally, the transition to step 4 happens after sample 74, the HMM identifies $S_3$ at sample 78 and remains
7.2. RESULTS

dere. For pairs 4-6, the identified transition sequence is similar to the one of pair 3. Therefore, the result of these pairs are not shown here, but in Appendix B.1.1 including the sampled output of the estimator.

<table>
<thead>
<tr>
<th>Input of HMM: Pair 3) PA-PD</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Graph 1" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output of HMM: Pair 3) PA-PD</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2" alt="Graph 2" /></td>
</tr>
</tbody>
</table>

Table 7.4. Results of the sampled output of the Kalman filter and the identified path for pair 3.

In the case of pairs 1 and 2, only 3 states were identified correctly, and state $S_3$ was identified correctly only for pair 1. The input and output of the model HMM$_F$ for pair 1 is shown in Table 7.5. The transition to step 2 happens after sample 25, however, the HMM failed to identify the samples after this as state $S_1$. This is because after some time, $\hat{r}_x$ converges to a similar value as state $S_0$. The transition from step 2 to step 3 occurs after sample 48, where the HMM identified correctly the following samples as state $S_2$. Note that, to go from $S_0$ to $S_2$, given the state transition matrix $A$, the system must go through $S_1$, so there is one sample that is identified as $S_1$. Finally, after the transition to step 4 occurred, the final state was identified as $S_3$.

The result of pair 2 is presented here since $S_3$ was not successfully identified. The input and output of the HMM are shown in Table 7.6. For this pair, after the change from step 3 to step 4 occurred the HMM is not able to identify it as a state transition, even though there was a visible change in $\hat{r}_x$ and $\hat{r}_z$ after sample 75.
Table 7.5. Results of the sampled output of the Kalman filter and the identified path for pair 1.

Table 7.6. Results of the sampled output of the Kalman filter and the identified path for pair 2.
### 7.2. RESULTS

#### Task 2

The HMM$_F$ for task 2 is evaluated in the same way as task 1. The analysis of the results for the six pairs are compiled in Table 7.7. The four states were identified correctly for 5 out of the 6 transition sequences performed (pairs 2 to 6). For pair 1, the system failed to identify $S_2$, but identified the final state correctly. The longest time it took the system to detect a transition was 0.95 seconds.

<table>
<thead>
<tr>
<th>Pairs</th>
<th>States detected</th>
<th>Detected final state</th>
<th>Reaction time to changes (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PA - PB$</td>
<td>3</td>
<td>✓</td>
<td>$S_0 \rightarrow S_1$</td>
</tr>
<tr>
<td>$PA - PC$</td>
<td>4</td>
<td>✓</td>
<td>$S_1 \rightarrow S_2$</td>
</tr>
<tr>
<td>$PA - PD$</td>
<td>4</td>
<td>✓</td>
<td>$S_2 \rightarrow S_3$</td>
</tr>
<tr>
<td>$PB - PC$</td>
<td>4</td>
<td>✓</td>
<td>$S_1 \rightarrow S_2$</td>
</tr>
<tr>
<td>$PB - PD$</td>
<td>4</td>
<td>✓</td>
<td>$S_2 \rightarrow S_3$</td>
</tr>
<tr>
<td>$PC - PD$</td>
<td>4</td>
<td>✓</td>
<td>$S_1 \rightarrow S_2$</td>
</tr>
</tbody>
</table>

Table 7.7. Results of model HMM$_F$ for task 2.($^*$) The time is taken from $S_1$ to $S_3$.

The input and output of the HMM for pair 1 is shown in Table 7.8 since it was the only data of this task for which only 3 states were identified correctly. The unidentified state was $S_2$, however, the final state was identified correctly. The input signal and the identified sequence for pairs 2-6 are found in Appendix B.1.2.

#### Input of HMM$_F$: Pair 1) PA-PB

#### Output of HMM$_F$: Pair 1) PA-PB

Table 7.8. Results of the sampled output of the Kalman filter and the identified path for pair 1.
7.2.2 Results for model HMM$_{NF}$

**Task 1**

The analysis of the results for the three different pairs of Group 1 are shown in Table 7.9. The evaluation is the same as for HMM$_F$, except the response time will not be considered since it is not really measurable from state to state.

<table>
<thead>
<tr>
<th>Pairs</th>
<th>States detected</th>
<th>Detected final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) PA - PB</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>2) PA - PC</td>
<td>2</td>
<td>✓</td>
</tr>
<tr>
<td>3) PB - PC</td>
<td>1</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 7.9. Results of model HMM$_{NF}$ for task 1.

The results of HMM$_{NF}$ for pairs 1 and 3 are the same, there is no change detected at all during the whole transition, the model remained in $S_0$. The results for pair 1, for example, are presented in Table 7.10 where there are noticeable changes in the input signal, however they are not detected. The result of pair 3 is shown in B.2.

**Input of HMM$_{NF}$: Pair 1) PA-PB**

![Input Graph]

**Output of HMM$_{NF}$: Pair 1) PA-PB**

![Output Graph]

Table 7.10. Results of the sampled output of the Kalman filter and the identified path for pair 1.

For pair 2, only the initial and final state are identified correctly, and $S_1$ is identified as $S_0$ while $S_2$ is identified as $S_3$. See the input and output in Table 7.11.
7.2. RESULTS

Table 7.11. Results of the sampled output of the Kalman filter and the identified path for pair 2.
CHAPTER 7. EXPERIMENTS

Task 2

The analysis of the results for the three pairs of group 1 are shown in Table 7.12. The evaluation is the same as for the previous task. The model HMM$_{NF}$ failed to detect any change during the transition sequence. This is because the signal deviates too much from the signal used for training.

<table>
<thead>
<tr>
<th>Pairs</th>
<th>States detected</th>
<th>Detected final state</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) PA - PB</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>2) PA - PC</td>
<td>1</td>
<td>x</td>
</tr>
<tr>
<td>3) PB - PC</td>
<td>1</td>
<td>x</td>
</tr>
</tbody>
</table>

Table 7.12. Results of model HMM$_{NF}$ for task 2.

The input and output of the HMM is shown here for pair 1 in Table 7.13. The plots for pairs 2 and 3 are presented in Appendix B.2.

Table 7.13. Results of the sampled output of the Kalman filter and the identified path for pair 1.
Chapter 8

Conclusions

8.1 Summary

This report describes a method to determine a change in the number of people jointly grasping an object with a robot using only force and torque information. The proposed solution is based on a transition sequence of four steps that humans should perform to go from the initial scenario to the final one. The force and torque information is used to estimate the grasping point of the agents with a Kalman filter. While the humans are going from one step of the transition sequence to the next, the estimated point changes according to the step the humans are in. These changes are used to track the steps in the sequence using a hidden Markov model (HMM). Using HMMs to track the steps in the transition sequence helps to disambiguate steps that provide similar observations, which is the case for the initial and final step. In this thesis, two different models were trained: one HMM for a case when humans perform the transition sequence just holding the object (without applying any force towards the robot), and the other HMM for when the humans hold and apply a slight force towards the robot. To evaluate the method, humans that were not involved in the training of the HMM were asked to perform two tasks: 1) perform the previously defined transition sequence; 2) perform a deviation of the predefined transition sequence. The humans performed both tasks twice. The first time they performed the tasks, they did not apply any force towards the robot. The second time, they were asked to apply a slight force towards the robot.

The results of the HMM for cases when humans apply force towards the robot are satisfactory. For the first task, the final step of the sequence was identified correctly 5 out of 6 times, and tracking all of the steps during the transition sequence were identified correctly 4 out of 6 times. For the second task, the final step was identified correctly all 6 times, and the whole transition sequence was tracked correctly 5 out of 6 times. The reason for this, is that for both tasks, the humans provide forces that make the estimator give similar signals, even though the forces are not applied from the same points in the different tasks.
For the HMM that was trained for cases when humans grasp without applying force towards the robot, the results were not satisfactory. In task 1, only 1 out of 3 times did the HMM detect the final state and the complete sequence was not identified at all. For task 2, the HMM was not able to detect any change at all. The reason for this, is that without applying any force, the estimator cannot converge to the same place, therefore, the output of the estimator for task 1 was sometimes too different. The output of the estimator for task 2 differed significantly from the training data, thus there was no change detected at all.

The results of this thesis suggest that this method is suitable for cooperative manipulation tasks where forces will be applied towards the robot by the humans. In situations where the humans will not apply any force, this method will not provide satisfactory estimation of the agents grasping the object.

8.2 Future Work

The solution proposed in this report consists on having one HMM for one transition sequence, where Agent 2 always joins grasping at the right side of the robot. To make the solution more robust, different HMMs could be trained for different scenarios involving different changes and use the solution to problem 1, defined in Chapter 4, to choose the output of the sequence of the parameter set that better matches the observations. The HMM in the method could also be trained for sequences where more than two agents are involved and explore how far apart agents have to be to distinguish the steps in the sequence.

Additionally, if this method were to be used with a vision based system for the same purpose, it could make for a more reliable detection of the number of agents and could also provide reliable information about where the agents are grasping.
Bibliography


Appendix A

Grasping point estimator results

A.1 Group 1

A.1.1 Task 1

Figure A.1. Output of the grasping point estimator of task 1 performed by pair 1.
APPENDIX A. GRASPING POINT ESTIMATOR RESULTS

Figure A.2. Output of the grasping point estimator of task 1 performed by pair 2.

Figure A.3. Output of the grasping point estimator of task 1 performed by pair 3.
A.1. GROUP1

A.1.2 Task 2

Figure A.4. Output of the grasping point estimator of task 2 performed by pair 1.
APPENDIX A. GRASPING POINT ESTIMATOR RESULTS

Figure A.5. Output of the grasping point estimator of task 2 performed by pair 2.

Figure A.6. Output of the grasping point estimator of task 2 performed by pair 3.
A.2. GROUP 2

A.2 Group 2

A.2.1 Task1

Figure A.7. Output of the grasping point estimator of task 1 performed by pair 1.
Figure A.8. Output of the grasping point estimator of task 1 performed by pair 2.

Figure A.9. Output of the grasping point estimator of task 1 performed by pair 3.
A.2. GROUP 2

Figure A.10. Output of the grasping point estimator of task 1 performed by pair 4.

Figure A.11. Output of the grasping point estimator of task 1 performed by pair 5.
Figure A.12. Output of the grasping point estimator of task 1 performed by pair 6.
A.2. GROUP 2

A.2.2 Task 2

Figure A.13. Output of the grasping point estimator of task 2 performed by pair 1.
Figure A.14. Output of the grasping point estimator of task 2 performed by pair 2.

Figure A.15. Output of the grasping point estimator of task 2 performed by pair 3.
A.2. GROUP 2

Figure A.16. Output of the grasping point estimator of task 2 performed by pair 4.

Figure A.17. Output of the grasping point estimator of task 2 performed by pair 5.
Figure A.18. Output of the grasping point estimator of task 2 performed by pair 6.
Appendix B

HMM results

B.1 HMM$_F$

B.1.1 Task 1

### Input of HMM$_F$: Pair 4) PB-PC

![Sampled $J_a$](image)

### Output of HMM$_F$: Pair 4) PB-PC

![Estimated sequence](image)

**Table B.1.** Results of the sampled output of the Kalman filter and the identified path for pair 4.
### APPENDIX B. HMM RESULTS

#### Table B.2. Results of the sampled output of the Kalman filter and the identified path for pair 5.

<table>
<thead>
<tr>
<th>Samples</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>S_1</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>S_2</td>
<td>S_3</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>S_4</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>S_5</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>S_6</td>
</tr>
</tbody>
</table>

#### Table B.3. Results of the sampled output of the Kalman filter and the identified path for pair 6.

<table>
<thead>
<tr>
<th>Samples</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>S_1</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>S_2</td>
<td>S_3</td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>S_4</td>
</tr>
<tr>
<td>80</td>
<td></td>
<td>S_5</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>S_6</td>
</tr>
</tbody>
</table>
B.1. HMM$_F$

### B.1.2 Task 2

<table>
<thead>
<tr>
<th>Input of HMM$_F$: Pair 2) PA-PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Graph of sampled $f_a$]</td>
</tr>
<tr>
<td>![Graph of sampled $f_b$]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output of HMM$_F$: Pair 2) PA-PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Graph of estimated sequence]</td>
</tr>
</tbody>
</table>

**Table B.4.** Results of the sampled output of the Kalman filter and the identified path for pair 2.
### APPENDIX B. HMM RESULTS

#### Input of HMM\(_F\): Pair 3) PA-PD

![Graph showing sampled output for PA-PD](image)

<table>
<thead>
<tr>
<th>Sampled ( j_k )</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

#### Output of HMM\(_F\): Pair 3) PA-PD

![Graph showing estimated sequence for PA-PD](image)

\( S_1, S_2, S_3, S_4 \)

#### Table B.5. Results of the sampled output of the Kalman filter and the identified path for pair 3.

#### Input of HMM\(_F\): Pair 4) PB-PC

![Graph showing sampled output for PB-PC](image)

<table>
<thead>
<tr>
<th>Sampled ( j_k )</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>68</td>
</tr>
<tr>
<td></td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>100</td>
</tr>
</tbody>
</table>

#### Output of HMM\(_F\): Pair 4) PB-PC

![Graph showing estimated sequence for PB-PC](image)

\( S_1, S_2, S_3, S_4, S_5 \)

#### Table B.6. Results of the sampled output of the Kalman filter and the identified path for pair 4.
B.1. HMM$_F$

Table B.7. Results of the sampled output of the Kalman filter and the identified path for pair 5.

Table B.8. Results of the sampled output of the Kalman filter and the identified path for pair 6.
APPENDIX B. HMM RESULTS

B.2 HMM\textsubscript{NF}

Task 1

<table>
<thead>
<tr>
<th>Input of HMM\textsubscript{NF}: Pair 3) PB-PC</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image of sampled output" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output of HMM\textsubscript{NF}: Pair 3) PB-PC</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image2.png" alt="Image of estimated sequence" /></td>
</tr>
</tbody>
</table>

Table B.9. Results of the sampled output of the Kalman filter and the identified path for pair 3.
B.2. $\text{HMM}_{\text{NF}}$

Task 2

**Table B.10.** Results of the sampled output of the Kalman filter and the identified path for pair 2.
APPENDIX B. HMM RESULTS

Input of HMM$_{NF}$: Pair 3) PB-PC

Output of HMM$_{NF}$: Pair 3) PB-PC

Table B.11. Results of the sampled output of the Kalman filter and the identified path for pair 3.