A Proximity Sensor Network for Autonomous Grasping

Iyad Abdul-baki

Department of Electrical Engineering
McGill University, Montreal
July, 1997

A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfilment of the requirements of the degree of Master of Engineering

©Iyad Abdul-baki, 1997
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-37253-7
Abstract

Object localization and tracking are important requirements for the autonomous grasping of moving objects. Autonomous grasping requires continuous information gathering by sensors. In this thesis, I examined the use of miniature amplitude based infra-red proximity sensors which are useful in this context since they can be placed directly in the robot's end-effector and thus guarantee unobstructed sensing. Unfortunately, using these sensors in a network leads to problems of local interference. To eliminate such problems, I synchronized and modulated the frequency of commercial sensors to enable their use in a network. In addition, I have developed two approaches which provide useful tools for autonomous grasping of a planar object with known geometry and surface properties. The first approach utilizes a network of three sensors and an on-line extended Kalman filter to localize the object and estimate it's six states (two positions, one orientation and three velocities). The second approach is heuristic and uses four binary sensors to localize and grasp a stationary object without the need for scanning. Both approaches have been tested experimentally, and in each case the objective was successfully achieved.
Résumé

La saisie autonome d'un objet mobile nécessite sa localisation ainsi que le suivi de sa trajectoire. La saisie nécessite une rétroaction continue à partir de mesures provenant d'un ou de plusieurs capteurs. Des capteurs de proximité à infra-rouge s'avèrent très utiles dans ce contexte. En effet, grâce à leur taille miniaturée, ces capteurs peuvent être placés au niveau de l'organe terminal effecteur éliminant ainsi les problèmes d'occultation. L'utilisation de ces capteurs dans un réseau entraîne des problèmes d'interférence. Pour éliminer ces problèmes, on propose dans cette thèse une synchronisation et une modulation des fréquences de ces capteurs. Dans cette thèse, on développe et on met en pratique deux techniques permettant des outils nécessaires à une saisie autonome d'un objet plan. Dans ces deux techniques, on suppose que la géométrie de l'objet ainsi que les propriétés de sa surface sont connues. La première technique vise la localisation de l'objet. Basée sur les mesures d'un réseau de trois capteurs, on utilise un filtre de Kalman étendu pour déterminer la dynamique de l'objet. La seconde technique est plutôt heuristique. Elle vise principalement des objets stationnaires et elle a pour objective la réalisation de la tâche saisie sans balayage. Quatre capteurs à sorties binaires sont utilisés dans cette technique.
Acknowledgements

I would like to acknowledge the financial support offered to me by the Natural Sciences and Engineering Council of Canada (NSERC). Moreover, I am greatful for the support from M.P.B. technologies Inc. and for the cooperation and interest of Dr. Ian Sinclair.

A very special thank you also goes to the people who made my stay in the lab enjoyable. In particular, Marc Leblanc for his general help. Mojtaba Ahmadi for his continuous concern and for being a model when it comes to passing knowledge from one generation to the other. Anca Cocosco for her cheerfulness and remembering everyone’s birthday. Ken Yamazaki for reviewing the manuscript overnight. John Damianakis and Greg Petryk for introducing me to the work. I also to thank two students who are not members of the lab but frequently answered my technical questions, namely Michael Glaum and Kaouther Benameur. In addition, I extend my gratitude to Akio Yoshinaka and Paul Hubbard for their help in proof reading the manuscript.

Professor Martin Buehler is thanked for giving me the opportunity to work in his lab, and for supervising me throughout the work. The vast experience that I acquired while working under his supervision is greatly appreciated.

The two years that I spent at the center for intelligent Machines were very joyful and productive. I think that I owe it to the staff, secretaries and fellow students for making this stay very enjoyable.

Last but not least, I thank my parents and brothers for their support.
## Contents

1 Introduction ................................................................. 2
   1.1 Motivation .................................................................. 2
   1.2 Problem Formulation .................................................. 4
   1.3 Literature Review ...................................................... 6
   1.4 Experimental Setup .................................................... 9
   1.5 Author's Contributions ............................................... 10
   1.6 Thesis Organization ................................................... 11

2 Proximity Sensors ............................................................. 14
   2.1 Requirements .......................................................... 14
   2.2 The STM Sensor ....................................................... 15
   2.3 Sensor Multiplexing .................................................... 17
       2.3.1 Sensors Synchronization Via Software ...................... 19
   2.4 Sensor Frequency ...................................................... 20

3 Object Localization: Extended Kalman Filter ......................... 22
   3.1 Introduction ........................................................... 22
       3.1.1 System Formulation ............................................. 22
       3.1.2 Kalman Filter .................................................... 24
       3.1.3 The Sensor Network ............................................ 25
   3.2 Filter Formulation ..................................................... 26
       3.2.1 Object Model ................................................... 28
CONTENTS

3.2.2 Sensor Model ........................................... 29
3.2.3 Jacobian Derivation .................................... 33
3.2.4 Observability Analysis .................................. 39
3.2.5 Extended Kalman Filter Equations ....................... 42
3.3 Filter Implementation Issues ............................... 43
  3.3.1 State Vector Initialization ............................. 43
  3.3.2 Filter Fine Tuning ................................... 44
  3.3.3 State Covariance Matrix Behaviour .................... 46
  3.3.4 Divergence and Observability Checks .................. 46
  3.3.5 Effect of Sampling Period ............................. 46
3.4 Discussion of Results ..................................... 47
  3.4.1 Filter Performance .................................... 47
  3.4.2 Filter Fine tuning .................................... 52
  3.4.3 Autonomous Object Localization ....................... 52
3.5 Conclusion .................................................. 55

4 Object Grasping: Heuristic Approach ............................ 58
  4.1 Introduction ............................................. 58
  4.2 Conceptual Formulation of the Approach ..................... 59
    4.2.1 Assumptions ........................................ 59
    4.2.2 The Sensor Network ................................ 60
    4.2.3 An Example ........................................ 66
  4.3 Implementation of the approach ............................ 66
  4.4 Conclusion ............................................... 72

5 Conclusion .................................................. 74
  5.1 Future Recommendations ................................... 74
List of Figures

1.1 *Gripper pads and tool interface used by Hydro Quebec* .................. 5
1.2 *Planar view of gripper and the “T” shaped object* ..................... 6
1.3 *Actuated PPR robot with gripper* ...................................... 12
1.4 *Unactuated RRR robot holding Hydro Quebec tool interface* ......... 13

2.1 *The STM sensor head* .................................................. 15
2.2 *Modifications on the STM sensor electronics* ......................... 18
2.3 *Timing diagram for A/D and sensor pulse signal* ..................... 19

3.1 *The proposed sensor network* ......................................... 27
3.2 *Raw sensor output as a function of distance and angle* .............. 30
3.3 *Fitted sensor output as a function of distance and angle* .......... 31
3.4 *Error in curve fitting raw data as a function of angle and distance* ........ 32
3.5 *Gripper-object detailed view* .......................................... 34
3.6 *Effect of sampling period on the filter performance* ................. 48
3.7 *Object states estimates using the extended Kalman filter* .......... 50
3.8 *Error in state estimate using the extended Kalman filter* .......... 51
3.9 *Effect of changing R on the filter performance: pose* ............... 53
3.10 *Effect of changing R on the filter performance: velocity* .......... 54
3.11 *Autonomous object localization* ....................................... 56

4.1 *Planar view of the tool interface* .................................... 59
4.2 *The proposed sensor network* ......................................... 61
LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Idealized sensor domains for the proposed sensor configuration</td>
<td>62</td>
</tr>
<tr>
<td>4.4</td>
<td>Gripper commands per sensor domains</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>Actual sensor domains</td>
<td>69</td>
</tr>
<tr>
<td>4.6</td>
<td>Autonomous grasping experiment: case 1</td>
<td>70</td>
</tr>
<tr>
<td>4.7</td>
<td>Autonomous grasping experiment: case 2</td>
<td>71</td>
</tr>
</tbody>
</table>
# List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td><em>Recent research in the use of electro-optical sensors</em></td>
<td>8</td>
</tr>
<tr>
<td>1.2</td>
<td><em>Specifications of the actuated PPR robot</em></td>
<td>9</td>
</tr>
<tr>
<td>2.1</td>
<td><em>Effect of frequency on the STM sensor output</em></td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td><em>Initial estimation of the sensors parameters</em></td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td><em>The confidence level of each domain</em></td>
<td>64</td>
</tr>
<tr>
<td>4.2</td>
<td><em>The gripper command for each domain</em></td>
<td>68</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Motivation

Autonomous grasping of moving objects is an important task in robotics. Examples include a manufacturing process where robots grasp objects from a moving conveyer belt, or from other robots, and remote teleoperation where robots replace humans in hazardous environments like space, deep sea, nuclear waste sites, or high voltage wiring. In these environments, the robot may have to grasp and use various tools which are not necessarily attached to the end effector. Hence, the capability of a robot to grasp an object (e.g. a tool) is essential to its successful use.

Safe and efficient grasping can be defined as the ability of the robot to grasp an object without the colliding with it or losing it. Safe and efficient grasping requires continuous and unobstructed sensing of the object. Therefore the sensing system should be reliable for all the positions of the object prior to grasping.

In traditional remote teleoperation the vision system does not provide unobstructed sensing of the object; the operator, usually located in the safety and comfort of a control room, receives information about the real environment through various vision systems. Two cameras are usually used for depth perception (stereo vision). The cameras have to be far enough from the object so as to provide a global view of the
environment. Using the global system for autonomous grasping does not provide the unobstructed sensing required. This problem is pronounced the most just prior to grasping when the object is located very close to the gripper. The object is usually occluded from the camera by the robotic links or end effector. Therefore, there is a need for local sensing, one that provide information about the object during the stage when the camera cannot provide such information. Moreover, using a standard teleoperation system to grasp an object is rendered difficult due to the limited bandwidth of the whole system (especially in space operations where the operator is very far away from the site). These limitations can be overcome to an extent with better hardware and/or a very skilled operator. However, the resultant increase in cost can be prohibitive.

Thus, there is a need for a complementary system which can detect the object locally when it is very close to the gripper.

An alternative, low cost solution is to compliment the vision system with a local network of miniature electro-optical sensors. The advantages of using such a network are many: firstly the sensors are very small, available sensors are 2 mm in diameter [9], and can be embedded right into the robotic gripper thus eliminating any extra space requirements. Secondly, the sensors are rugged and robust, so one does not have to worry about torque limitations on the end effector. Thirdly, the sensors can be grouped into a network running at high frequencies (e.g., 500 Hz) compared to a camera, and used in a loop to provide feedback for robotic tracking. This loop could be running locally requiring very little input from the control room and thus enabling tracking to be achieved with a rich feedback.

The ability of humans to grasp objects of various geometries can be attributed to a sophisticated vision and object recognition system [13]. Robots, on the other hand, have difficulty in recognizing different objects and grasping them. One way to achieve such dexterity is to connect various objects to a standard interface. The problem of grasping various objects then simplifies to that of grasping the standard interface
with its known geometry and surface properties (color, texture, etc.).

Grasping can be categorized into smooth and non-smooth. A smooth grasp is achieved when the robotic gripper matches the object's position and velocity at time of grasp, and hence there will be no impact between the two. A non-smooth grasp is one where a match of velocity is not necessary and an impact or collision is acceptable.

The task of grasping, whether it involves a smooth or non-smooth grasp, encloses various subtasks or steps:

1. Sensing: finding or developing the proximity sensors which meet the required specifications. A list of the specifications is provided in section 2.1.

2. Localization: developing an approach that translate the sensor output into information about the object (e.g., position, velocity, ...etc).

3. Tracking: developing an approach for tracking of the object. Tracking is required so that the object is kept in the “view of the sensors” and brought to the area where it can be grasped.

4. Supervising: developing a supervisory level routine that can decide on the pre-grasping motion. For example, the routine has to decide on when to start tracking the object, when it is safe to approach the object, when to backup and re-attempt ...etc.

Items 1-3, namely finding the appropriate sensors for the task, and providing approaches to localize and track the object before grasping are the subject of this thesis. The supervisory level routine, item 4, which is responsible for making the decisions on grasping is beyond the scope of this thesis.

### 1.2 Problem Formulation

The specific problem that I deal with in this thesis is that of grasping a standard tool interface. One such example of a standard interface is the one developed by the
research team at the company Hydro Quebec for use in teleoperation. In particular, the company's research is focused on the use of teleoperation in live power line maintenance, where a high voltage power line can be repaired or replaced under harsh weather conditions and without any power interruption. The standard tool interface, presented in figure 1.1, can be attached to any tool needed. By knowing the offset of each tool from the tool interface, the operator is able to perform various tasks with the tool.

Since the geometry of the tool interface is fairly complicated, I have used a simplified planar version. If we assume that the gripper and the tool interface are restricted to planar motion, then the intersection of the tool interface with the said plane is a "T" shaped object. Thus, a "T" shaped object is developed and used for this work. Figure 1.2 shows a sketch of the planar version of the object and the gripper.

Figure 1.1: *Gripper pads and tool interface used by Hydro Quebec*
Figure 1.2: A planar view of gripper and the "T" shaped object (tool interface). The vertical section of the object corresponds to the back plate of the tool interface presented in figure 1.1. The section of the tool behind this plate is attached to the tool.

1.3 Literature Review

Electro-optical sensors can be classified by their mode of operation into three categories; triangulation based [12, 22], phase difference based [19, 23] and intensity based [3, 16, 18, 11, 2]. Triangulation based sensors are the most accurate but due to geometric constraints imposed by the method, the setup is large and would not satisfy the space constraints of our application. Proximity sensors using the phase difference as a mode of operation are presented in [3, 23, 19]. The sensor itself contains six LEDs placed in a cross shaped pattern with the receiver in the middle. The phase shift of the received signal is used to calculate the distance, angle, or orientation between the sensor and the object. The sensor has a large size to accommodate the various emitters and thus again would not meet the size constraints of our application. However, the third type of proximity sensors, the intensity based ones, are very small; 2 mm in diameter being commercially available [9]. The intensity based sensor can have
a minimum of one emitter and one receiver with the intensity of the received signal being used as an output. The output is a nonlinear function of many variables; these include the distance between the sensor and the object, the angle between the sensor and the normal to the surface of the object, and the object's reflectance properties (color, texture, surface finish, etc.). Moreover, the output is also a function of the ambient light conditions, except for a pulsed intensity based sensor.

One of the first to use electro-optical proximity sensors in robotics was Heinrich A. Ernst at M.I.T. in 1961 [7]. He equipped a mechanical hand with proximity sensors which were then used in performing some pick and place operations. The sensors were used in a binary mode, indicating the presence or absence of an object. In 1977, Johnston [11] and Bejczy [2] used intensity based proximity sensors. The analog output of the sensors was used to estimate object distance.

Since those earliest applications, various researchers have used intensity based sensors in robotics. For instance, in 1980 Espiau and Catros [8] examined various filtering approaches to estimate the distance between the sensor and object. They acknowledged the effect of the angle and surface properties but did not include them in their sensor model. The sensor output was modeled as a nonlinear function of distance only. In 1984, Wampler [30] assumed no angle dependency and a known constant reflectance gain to come up with a distance-voltage calibration table. In 1991, Li [14] assumed a negligible angle dependence and known reflective properties and used Extended Kalman filtering to estimate the distance to a flat object. The same concept is used again by Li [15] in 1993 where the distance between three sensors placed in a triangular pattern and a flat object was calculated using three independent extended Kalman filters. Using geometric relations, two of the objects orientation angles as well as the distance between the object and the sensors were calculated. A later study by Y. Li [17] acknowledged and used the angle dependency but assumed a constant reflectance gain in the sensor model. Most recently (1997), Petryk and Buehler [24] assumed that the output is a nonlinear function of both the distance between the
sensor and object, and the angle between the sensor and the normal to the surface of the object, and a linear function of the reflectance gain. With these assumptions, the authors [24] used an extended Kalman filter to estimate not only the position and velocity of a planar object, but also the reflectance gain. A summary of the above review is presented in table (1.1).

<table>
<thead>
<tr>
<th>Object Geometry</th>
<th>Estimation Scheme</th>
<th>States Estimated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Espiau &amp; Catros [8] (1980)</td>
<td>flat, EKF</td>
<td>1D position</td>
</tr>
<tr>
<td>Li [14] (1991)</td>
<td>flat, EKF</td>
<td>1D position</td>
</tr>
<tr>
<td>Li [15] (1993)</td>
<td>flat, 3 indep. EKFs</td>
<td>1D position, 2D orientation</td>
</tr>
<tr>
<td>Li [16, 17] (1994)</td>
<td>flat, EKF</td>
<td>1D position</td>
</tr>
<tr>
<td>Present thesis (1997)</td>
<td>&quot;T&quot; shaped, EKF</td>
<td>2D position, 2D velocity, 1D orientation, 1D angular velocity</td>
</tr>
</tbody>
</table>

Table 1.1: Recent research in the use of electro-optical sensors

As can be seen from the first and second columns of the table, most of the previous work is concerned with a flat planar object but none has yet succeeded in estimating all six states of a non-flat planar object.
CHAPTER 1. INTRODUCTION

The work presented in this thesis is a logical continuation of the previous work of all the aforementioned researchers; six states of a planar object are estimated, with a more complicated object geometry.

As it is noted from table (1.1), the Extended Kalman filter is one of the most popular estimation schemes. The topic of the Kalman filter and the extended Kalman filter is discussed in detail later in chapter 3. Moreover, the topic is present in almost any book on control, and has been addressed in various publications [1, 5, 4, 6, 10, 21, 28].

1.4 Experimental Setup

The test bed used for the experiments in this thesis consists of two planar custom made robots built in McGill's Autonomous Manipulation Laboratories (AML). The first is an actuated PPR robot (figure 1.3) while the second is an unactuated RRR robot (figure 1.4). Both robots have simple inverse kinematics and their end effectors can move in two translational and one rotational degree of freedom. The actuated PPR robot can travel anywhere in a workspace of $600mm \times 300mm$ and its end effector can rotate $360^\circ$. Further specifications on the actuated robot are provided in table (1.2). The unactuated robot has two links of length 400mm and 200mm. The last link lacks length because it is used for pure orientation of the object. Rotation of the links is read via three high optical encoders with a resolution of $7 \times 10^{-3}$ degrees.

<table>
<thead>
<tr>
<th></th>
<th>Travel</th>
<th>Drive train</th>
<th>Peak Motor Torq.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$-Stage</td>
<td>600mm</td>
<td>Belt (1 rev = 90 mm)</td>
<td>4.1Nm</td>
</tr>
<tr>
<td>$Y$-Stage</td>
<td>300mm</td>
<td>Ball Screw (20mm lead)</td>
<td>1.8Nm</td>
</tr>
<tr>
<td>$\alpha$-Stage</td>
<td>$360^\circ$, 127mm $\phi$</td>
<td>Worm Gear (36:1)</td>
<td>.35Nm</td>
</tr>
</tbody>
</table>

Table 1.2: Specifications of the actuated PPR robot. Courtesy: G. Petryk [25]

The gripper, along with the sensors, is attached to the end effector of the actuated robot, while the object is attached to the end effector of the unactuated robot. A
handle is used to move the object around. A custom made input/output electronics board provides the interface between the robots and the computer. The computer is a 166MHz PC running under QNX [27], a real-time operating system.

Information about the sensor hardware, and their interface via 12 bit analog to digital converters is explained later in chapter 2.

1.5 Author’s Contributions

The author’s contributions can be summarized into three points;

- Hardware modification of commercial pulse-based infra-red proximity sensors to achieve two goals; first synchronization so as to minimize any local interference, and second frequency modulation such that the sensors' frequency is controlled by computer. The first feature is essential for using the sensors in a network while the second feature is very important in “intelligent sensing”.

- The development of a three degree of freedom object localization approach for a “T”-shaped object. The estimation scheme is performed by a real-time extended Kalman filter which estimates the six states of a planar object, namely two positions, two velocities, one orientation and one angular velocity. According to the author’s knowledge, this is the first time that an extended Kalman filter has been used with pulse-based infra-red proximity sensors to estimate all three degrees of freedom of the planar object. Moreover, the geometry of the object presented, and consequently the derivation of the Jacobian is more involved than those presented in previous work.

- The development of a heuristic approach for non-smooth grasping of a stationary object. The approach uses the sensors in a binary mode, indicating the presence or absence of the object. The space in front of the gripper is divided into various regions and the gripper moves in such a way that the object is continuously
brought closer to the region where it is grasped. The approach requires no scanning of the object. According to the author's knowledge, this approach is novel in the sense that it uses proximity sensors in a binary mode for autonomous grasping.

1.6 Thesis Organization

The thesis is divided into five chapters. Chapter 1 is a general introduction which includes sections on literature review, problem formulation, experimental setup and the thesis contributions. In chapter 2 the reader finds information about the proximity sensors, and the modifications that enabled their use in a network. Chapter 3 presents an approach for object localization whereby an extended Kalman filter is used to estimate the pose and velocity of the "T" shaped object. Chapter 4 presents a heuristic approach for non-smooth grasping of the stationary object. Finally, chapter 5 contains the conclusion and recommendations for future work.
Figure 1.3: Actuated PPR robot with gripper holding a liquid paper container. The prismatic joint running from the lower left side of the figure is the X-stage, the other one is the Y-stage. The gripper is mounted on the α-stage.
Figure 1.4: Unactuated RRR robot holding Hydro Quebec tool interface
Chapter 2

Proximity Sensors

The present chapter introduces the proximity sensors. Section 2.2 has a general description of the sensors that match most of the requirements while section 2.3 provides a description of the modifications implemented on the sensors to permit their use in a network. Finally section 2.4 has information on the frequency characteristics of the modified sensors.

2.1 Requirements

The sensors that would be appropriate for our application should have the following characteristics:

- fast with sampling in the order of 100Hz.
- small sensors with a diameter less than 7mm.
- ambient light rejection
- range of about 0-100mm.
- no interference or cross-talk between sensors.
A commercial sensor that would match all of the above requirements was not found at the time of the search (1996). However, a sensor manufactured by the company "Sensor Technologies Munich" (STM [9]) was found to match most of the requirements. The STM sensor is described in detail in the following section.

2.2 The STM Sensor

The STM sensor is a pulse-based infra-red proximity sensor. The sensor consists of one emitter and one receiver which are housed together in a cylindrical steel casing (figure 2.1). The transmitter is an LED (Light Emitting Diode) which operates in the infrared region with a peak wavelength of 880 nm and a spectral radiation bandwidth of 80 nm. The receiver is a photo-transistor with maximum sensitivity at a wavelength of 850 nm. The small size of the housing, a cylinder of 13 mm length and 5 mm diameter, makes it suitable for use in gripper pads and the end effector in general.

Both the receiver and emitter are connected via a flexible electrical cable to the sensor's electronics box. The STM sensor operates as follows; the LED is pulsed or

![Figure 2.1: The STM sensor head](image)
CHAPTER 2. PROXIMITY SENSORS

turned on for a certain period of time during which it emits infra-red light. Whenever this light hits an object, it reflects and/or diffuses back and is picked up by the receiver. The signal out of the receiver is then filtered, and processed as the analog output of the sensor. The STM sensors are pulsed so as to filter out ambient light. “The light from the emitter is pulsed with a certain frequency and the electronics in the receiver only registers light with this frequency. The effect of light with any other frequency (ambient light) is suppressed” [9].

When the sensors were tested for interference, it was observed that “sometimes” the presence of one sensor close to a second one results in cross-talk. This form of interference between the sensors makes their outputs unreliable at best and could hinder our application. Therefore the sensor circuitry was analysed to understand the reasoning behind this interference.

The electronics circuitry responsible for pulsing the LED and receiver is analysed; a chip generates a 500 Hz pulse with a duration of 76 μsec. This pulse is used to control both the LED and the receiver which are turned on every time the pulse signal goes high, and shut off when the pulse signal is low. Hence the LED and receiver are turned on for a period of 76 μsec and turned off thereafter.

Given the mode of operation of the STM sensor described in section 2.2, we note that each sensor is being pulsed independently from the others. A problem could arise when more than one sensor are used together in a network. Suppose that at least two sensors happen to be turned on at the same time. The light emitted from the LED of one sensor could be picked up by the receiver of the second sensor, thus contaminating the output of the second and rendering the sensors unreliable.

A solution to this problem is to multiplex the sensors of the network. Multiplexing is achieved by synchronizing the time when a particular sensor is turned on, and ensuring that all the other sensors are turned off during the same period. This synchronization guarantees that the receiver of a particular sensor will register only the light emitted by its emitter. The hardware modifications needed to implement multiplexing is
discussed in the following section.

2.3 Sensor Multiplexing

Multiplexing many sensors is achieved by synchronizing the pulses controlling them. This is done by bypassing the built-in pulse generator of each sensor, and instead supplying the latter with a synchronized pulse whose width is 76 µsec. There are various ways to achieve this multiplexing. One way is to use a “multiplexer” chip. Another way is to use the computer and provide the pulses by software via a digital line. The second approach is favoured over the first because not only does it provide control over the frequency of sensors operation but it also makes the addition or deletion of sensors a very simple software task. A Max 187 12 bit analog to digital converter (figure 2.2) is used to convert the analog signals of the sensors to the digital signal required by the computer. The chip has one analog input, two digital control inputs, and one digital output for the converted signal. The chip operates in the following way; one of the digital inputs, called the Chip Select or CS, is high by default. Once the CS goes low, the analog to digital conversion is started. The chip’s analog input is read and converted into a 12 bit number, which is transferred over to the computer one bit at a time. The second digital input, the Serial Clock (Sclk), is used to control the outputing of the bits through the digital output line (DOUT). The (Sclk) signal goes high/low 12 times during which the 12 bits are transferred through the (DOUT) channel to the computer. After all the bits are transferred the CS is set high again signalling the termination of the conversion. This occurs within one cycle.

The CS digital signal is used to activate a “one-shot” chip (74122) which produces a pulse of 76 µsec. This pulse triggers the appropriate sensor. A schematic diagram of the modified circuit is shown in figure 2.2.

A timing scheme of the A/D operations and sensor pulses is provided in figure 2.3. At time 1 the CS goes from low to high signalling the end the A/D conversion.
Figure 2.2: The circuit diagram showing the modifications done on the STM sensor electronics. The Max 187 is the A/D converter, the LS74122 is the “one-shot” pulse-width regulator and LT1368 is the simple gain amplifier used to amplify the sensor output so that it matches the input requirements of the A/D. Values of the resistors and capacitors are given below; $R_1=R_2=10K$, $R_3=20K$, $R_4=50\Omega$, $C_1=0.1\mu F$, $C_2=10nF$, $C_3=0.1\mu F$, $C_4=22\mu F$
Meanwhile the sensor pulse signal goes high, signalling that the sensor is on. After 76 µsec (time = 2) the sensor pulse signal goes low and the sensor output is produced. At time = 3 the $\overline{CS}$ goes from high to low and the analog to digital conversion is started. When the conversion is finished, (the $\overline{CS}$ goes from low to hi) and the sensor is turned on again. This is repeated for each cycle.

![Diagram](image)

Figure 2.3: The timing diagram for the A/D converter and the sensor pulse signal. At time 1 the sensor is turn on and at time 2 it is turned off. At time 3 the A/D conversion starts and at time 4 the digital output has been transfered to the computer.

### 2.3.1 Sensors Synchronization Via Software

Finally, we need to provide the synchronized $\overline{CS}$ by software. First, let us assume that we have "n" sensors numbered from 1 to n and connected to the computer via an I/O board. A function which requires the sensor number as an input takes care of reading the $A/D$ of the corresponding sensor. Synchronization is achieved by calling this function periodically (the frequency will be discussed later in section 2.4). Sensor
number 1 is read first, then 2, ... n then back to 1 ...etc. This way each sensor is being turned on and read while the others are turned off.

2.4 Sensor Frequency

Though the sensors were originally designed to operate at a fixed frequency of 500 Hz, it is now controlled by computer. It is observed that the sensors running at frequencies less than 500 Hz still produce adequate results. For example, table (2.1) shows the outcome of experiments where the frequency of the sensor is varied while an object is placed in front of it. The average and standard deviation of the sensor output over the period of the run is reported because the instantaneous value is too noisy. It is observed that down to frequencies of 5 Hz the sensor's output is not affected by the frequency at which the LED and the receiver are turned on.

<table>
<thead>
<tr>
<th>Frequency (Hz)</th>
<th>Average sensor output (counts)</th>
<th>Standard deviation (counts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>366</td>
<td>2</td>
</tr>
<tr>
<td>208</td>
<td>367</td>
<td>2</td>
</tr>
<tr>
<td>100</td>
<td>367</td>
<td>2</td>
</tr>
<tr>
<td>50</td>
<td>367</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>367</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2.1: The effect of frequency on the output of the STM sensor for a stationary object

The fact that the sensor output is not affected by its frequency is of great advantage for mobile robots with limited on board power. When the LED is turned on, the power consumption of the sensor is at its peak. When the LED is off, the power consumption of the sensor drops accordingly. Thus having the sensors run at a low frequency, which
is now done very easily by software, will reduce the total power consumption. The idea that the sensors can be operated at different frequencies makes them very useful in an “intelligent sampling” system. For example, the sensors could be used on remote robots with limited on board power (examples include space and deep sea exploration). Initially the sensors would be running at low frequencies to provide protection from collision with any obstacle. Once an object is detected, then the frequency can be increased to obtain more updates as needed.

Moreover, these modifications facilitate the addition or deletion of sensors in the sensor network. The sensor is simply plugged into the appropriate channel and can be used with minimal software modification.

In the experiments performed for this thesis, the sensors were running at a fixed frequency of 250 Hz.
Chapter 3

Object Localization: Extended Kalman Filter

3.1 Introduction

In this chapter, an extended Kalman filter uses the output of the STM sensors to estimate the states of the "T" shaped object. The estimated states are two positions, two velocities, one orientation and one angular velocity. This localization is a required step to achieve autonomous grasping as explained in section 1.2. We start this chapter with a general introduction to the system and the extended Kalman filter topic. Section 3.2 includes all the formulations required for the Kalman filter, namely an object model, a sensor model, the Jacobian derivation, and the observability analysis. Section 3.3 deals with filter implementation issues and section 3.4 is a presentation and discussion of the results. Finally, section 3.5 is a conclusion of the chapter.

3.1.1 System Formulation

The free floating object is modeled as a first-order linear dynamic system. In the discrete time domain this model is:

\[ x(k+1) = \Phi(k)x(k) + \Gamma(k)u(k) + w(k), \]  

(3.1)
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

where $x$ is the state vector containing the states of the object. $\Phi$ and $\Gamma$ are the state transition matrix and input matrix respectively. The input vector $u$ is used to introduce the known inputs of the object while $w$ is the model noise used to allow for any unmodeled dynamics of the system.

The object is monitored by $n$ sensors and the output equation for the $i^{th}$ sensor is given by:

$$ y_i = h_i(x) + v_i, \quad (3.2) $$

where $h_i$ is a non-linear function relating the sensor output to the states and $v_i$ is the sensor model noise. When using a network of sensors, we can use a vector format of $y$:

$$ y = h(x) + v, \quad (3.3) $$

where $v$ is the vector containing the sensor noise,

$$ y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \quad (3.4) $$

and

$$ h(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_n(x) \end{bmatrix}. \quad (3.5) $$

The object localization problem is how to estimate the state vector $x$, whose dynamic model is given by equation (3.1), and it is being observed by $n$ sensors modelled by equation (3.3). The estimator should be able to use both the dynamic model and the sensor model to come up with an estimation of the states. The most common estimation scheme for systems whose output is a non-linear function of the states is the extended Kalman filter. However, since the extended Kalman filter is a special form of the Kalman filter, the latter is introduced first.
3.1.2 Kalman Filter

The Kalman filter is an optimal recursive data processing algorithm for linear systems [20]. It is optimal in that it incorporates all the information that is provided to it, regardless of their precision, to estimate the current value of the state. The latter is obtained by combining a prediction of the state, computed from history based on a given model, and the current weighted measurement data in such a way that the error is minimized statistically (minimizing the covariance of the state). The Kalman filter is recursive in the sense that although it incorporates the history into the present, it does not require all previous data to be kept in storage and reprocessed at every iteration. In particular, the Kalman filter addresses the general problem of estimating the state $x$ of a first-order, discrete-time process which is governed by the linear difference equation,

$$
x(k + 1) = \Phi(k)x(k) + \Gamma(k)u(k) + w(k),$$  \hfill (3.6)

$$
y(k) = h(x(k)) + v(k),$$  \hfill (3.7)

where $x(k)$ is the state vector at time $k$. The vectors $w(k)$ and $v(k)$ are the object model and the sensor noise respectively. Both noises are assumed to be Gaussian. $\Phi$ is the state transition matrix which relates the state at time step $k$ to the state at step $k + 1$ in the absence of any inputs or object model noise. The $\Gamma$ matrix relates the input vector $u$ to the state $x$. The vector $y(k)$ is the output at time $k$, while $h(x(k))$ is the vector whose elements are the “linear” functions relating the output of the sensors to the state vector. “linear” is a key word here because the Kalman filter requires that the system measurement model be linear. For nonlinear measurement models, different approaches have been developed. The most popular one is the extended Kalman filter.

The extended Kalman filter is a special form of the Kalman filter that linearizes a non linear measurement model with respect to the present states. The linearization
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

is a first order Taylor expansion of the measurement equation (3.7);

\[ y(k) = H(x(k))\delta x + v(k), \]  

(3.8)

where

\[ H(x) = \left[ \frac{\partial h(x)}{\partial x} \right] \]  

(3.9)

is the Jacobian of the non-linear measurement functions, \( h(x) \).

3.1.3 The Sensor Network

In choosing the number of sensors and their positions the following restrictions were applied;

- the minimum number of sensors are used
- the configuration of the sensors corresponds to having an "observable system". This topic is discussed in detail in section 3.2.4.
- the placement of the sensors is in such a way that simplifies the geometry of the problem. This is important for the Jacobian derivations.

Our proposed sensor network is presented in figure 3.1. The network consists of three sensors; sensors 1 and 2 are mounted parallel to the y-axis of the gripper while sensor 3 is mounted parallel to the x-axis. The coordinates of the \( i^{th} \) sensor \( (s_{ix}, s_{iy}) \) are expressed in the gripper's reference frame. \( d_i \) is the distance between the \( i^{th} \) sensor and the object and \( \theta_i \) is the angle between \( d_i \) and the normal to the surface at that point. Note that we have made the assumption that the field of vision of each of the sensors is a straight line. In other words, the sensor's incident light beam projects a point on the object. This assumption was made to simplify the geometric analysis.

The orientation of the sensors is chosen according to the geometry of the object in such a way that all the angles \( \theta_1, \theta_2, \) and \( \theta_3 \) are equal. This enables the indices to be dropped. Any of the angles will be referred to hereafter as \( \theta \),

\[ \theta_1 = \theta_2 = \theta_3 = \theta. \]  

(3.10)
This choice of the orientation of the sensors with respect to the gripper simplifies the geometry of the problem and the Jacobian derivation (section 3.2.3).

The states of the object that we wish to estimate are the pose (position, orientation) and the velocities. The pose of a solid body can be determined by knowing the position of a point on that body and the orientation of a specific line passing through that point. If we consider the point B and the x-axis of the object's frame, then the problem of finding the object's pose is equivalent to finding the coordinates of point B and the angle between the B and G frames, \([\theta_B]_G\).

Let \([B_x]_G\) and \([B_y]_G\) be the x and y components of the point B, and \([\theta_S]_G\) be the relative angle between the object's and the gripper's reference frame. The state vector expressed in the gripper's reference frame can be written as:

\[
x = \begin{bmatrix}
x_1 \\
x_1 \\
x_2 \\
x_2 \\
x_3 \\
x_3
\end{bmatrix},
\]

where

\[
x_1 = [B_x]_G \\
x_2 = [B_y]_G \\
x_3 = [\theta_S]_G.
\]

### 3.2 Filter Formulation

The extended Kalman filter requires an object model, a sensor model and the Jacobian of the measurements equation. These requirements are addressed in sections 3.2.1, 3.2.2, and 3.2.3 respectively. Moreover, the observability analysis of our system is discussed in section 3.2.4.
Figure 3.1: The proposed sensor network
3.2.1 Object Model

The dynamic object model in discrete time domain is

\[ x(k + 1) = \Phi x(k) + \Gamma u(k) + w, \tag{3.12} \]

where \( k \) is the time step and \( x(k) \) is the state vector defined in section 3.1.3. Note that the state vector and consequently equation (3.12) is given in the gripper's reference frame. \( \Phi \) and \( \Gamma \) are the state transition matrix and input matrix respectively, while \( w \) is the error due to unmodeled object dynamics. Note that we are assuming the noise to be time invariant.

Since we have no knowledge about the object’s input (forces to a point mass in the plane), we assume a zero input. The input vector \( u \) is set to zero \((u = 0)\). Therefore, the object is assumed to be at steady state conditions and accelerations are modeled as disturbances. The dynamic object model equation can be rewritten as:

\[ x(k + 1) = \Phi x(k) + w. \tag{3.13} \]

Since we don’t know the object’s model, the state transition matrix is chosen to be:

\[ \Phi = \begin{bmatrix} 1 & \delta t & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.14} \]

and \( \delta t \) is the sampling period of the Kalman filter. This choice of the state transition matrix corresponds to the following:

\[ x_i(k + 1) = x_i(k) + \dot{x}_i(k)\delta t + w_i \tag{3.15} \]

\[ \dot{x}_i(k + 1) = \dot{x}_i(k) + w_i, \tag{3.16} \]
which is assuming no acceleration of the object. Moreover, the object model noise is assumed to be stationary, Gaussian and not correlated in time. The acceleration of the object is not necessarily Gaussian but this assumption is needed for the implementation of the Kalman filter. Mathematically this translates to:

\[
E[w(n) w(m)] \approx \begin{cases} Q & n = m \\ 0 & n \neq m \end{cases}
\]  

(3.17)

where \( Q \) is the diagonal system covariance matrix.

### 3.2.2 Sensor Model

A sensor model is needed to relate the output of the sensors to the states. This relationship is found in two stages; first a model that relates the sensor output to both the distance and the angle between the sensor and the object is found. Second, the said distance and angle are related geometrically to the states. The former part of the relationship is developed in this section while the latter is derived in section 3.2.3.

To find a model that relates the output of the sensors to the distance and the angle between the sensor and the object, sensor calibration has to be performed to obtain the raw data. A model can then be fitted to the data.

### Sensor Calibration

The calibration of the sensors is performed experimentally in the following steps;

1. The sensor to be calibrated is placed in a fixture which is located at a known position and orientation.

2. A rectangular object, which is large compared to the sensor diameter, is covered with white paper. This object is then mounted on the robot's end effector.

3. The object is moved in front of the sensor such that both its location and orientation are changed.
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

4. The states of the object are recorded along with the sensor output. Then, by using geometry, the relative distance and angle between the object and sensor are calculated.

5. The sensor output is plotted as a function of the distance and angle.

Figure 3.2 shows a typical sensor output as a function of distance and angle. The output itself has units of counts (12 bits corresponding to 4096 counts), while the units of distance and angle are mm and degrees respectively. The relations between the sensor output and the angle shows symmetry about the zero angle. Moreover, there is an inversely proportional relationship between the sensor output and the distance. Based on these observations, the following sensor model is proposed:

\[ h_i = \frac{\lambda \beta_{1,i}}{(d_i + \beta_{4,i})^\beta_{3,i}} \cos(\beta_{2,i} \theta_i), \quad (3.18) \]

where \( h_i \) is the \( i^{th} \) sensor output, and \( \lambda \) is the reflectance gain coefficient which contains information on the surface properties of the object (e.g., color, level of reflection, surface finish, etc.). In this thesis, the object surface properties are assumed known.
Figure 3.3: Fitted sensor output as a function of distance and angle. The β coefficients are obtained by using the recursive least square method on the raw data.

so λ is set to 1. \( d_i \) and \( \theta_i \) are the distance and the angle between the object and the sensor respectively, while \( \beta_{1,i} \) to \( \beta_{4,i} \) are constants to be determined.

The \( \beta_{1,i} \) to \( \beta_{4,i} \) constants are determined by using the recursive least square method [26, 29] to find the values that minimize the least square error between the model and the raw data. Figure 3.3 shows a plot of the sensor model fit using the recursive least square method, and figure 3.4 shows the error between the model and the raw data as a function of both the angle and the distance. The upper plot shows that the error is uniform across the whole range of the angle with a zero mean and a standard deviation of about 15%. However, the error between the model and the raw output is not uniform with distance; the mean of the error is always around zero but the standard deviation increases for larger distances. This result is intuitively correct; for large distances, the sensor output is low. The effect of noise (usually a constant value) is more pronounced for small outputs than it is for large ones because the signal to noise ratio is lower. Moreover, the sensor output curve is very flat for large distances which means that any small error in the sensor output will translate into a large error.
Figure 3.4: The error in curve fitting the raw sensor output; the upper plot shows the error as a function of angle while the lower plot shows the error as a function of distance, both in percentage error of current value.
3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

in distance. This is the reason why for large distances large errors are produced.

3.2.3 Jacobian Derivation

The Jacobian matrix $\mathbf{H}$, of the measurement functions, $h_1$, $h_2$, $h_3$ with respect to the state vector can be divided into three submatrices:

\[
\mathbf{H} = \begin{bmatrix}
\mathbf{H}_1 \\
\mathbf{H}_2 \\
\mathbf{H}_3
\end{bmatrix} = \begin{bmatrix}
\frac{\partial [h_1]}{\partial [x]} \\
\frac{\partial [h_2]}{\partial [x]} \\
\frac{\partial [h_3]}{\partial [x]}
\end{bmatrix}.
\] (3.19)

Using the chain rule, we can write,

\[
\mathbf{H}_i = \frac{\partial [h_i]}{\partial [x]} = \frac{\partial [h_i]}{\partial [p_i]} \cdot \frac{\partial [p_i]}{\partial [x]},
\] (3.20)

where $p_i$ is a parameter vector containing the sensor parameters that we need to estimate,

\[
p_i = \begin{bmatrix}
d_i \\
\theta
\end{bmatrix}.
\] (3.21)

Using the sensor model presented in section 3.2.2 we can calculate

\[
\frac{\partial [h_i]}{\partial [p_i]} = \begin{bmatrix}
-\beta_{1,i} \beta_3, i \lambda \cos(\beta_2, i \theta) \\
(d_i + \beta_4, i \lambda \sin(\beta_2, i \theta)) (d_i + \beta_4, i \lambda)
\end{bmatrix}.
\] (3.22)

Therefore, we are only missing the term $\frac{\partial [p_i]}{\partial [x]}$. To calculate this term we need to find a geometric relation between the states of the object and the sensor parameters $(d, \theta)$.

Figure 3.5 shows the object and the gripper; frames $\mathcal{G}$ and $\mathcal{B}$ are attached to the gripper and object respectively. The coordinates of the $i^{th}$ sensor, $(s_{ix}, s_{iy})$ are given in the gripper's reference frame. Line $L$ passes through point B and is parallel to the y-axis of the $\mathcal{G}$ frame. $d_3$ intersects the object at point A and when extended intersects $L$ at point C. Also, line $L'$ passes through the point E, defined as the point on the object seen by sensor 1, and is parallel to the x-axis of the $\mathcal{G}$ frame.
Figure 3.5: Gripper-object detailed view; Frame B, with a center B, is attached to the object while frame G, with a center G, is attached to the gripper as shown. Line L passes through point B and is parallel to the y-axis of the G frame, while line L' passes through the point on the object where sensor 1 shines and is parallel to the x-axis of the gripper. $d_3$ intersects the object at point A and when extended intersects L at point C.
As we recall from section 3.1.3, the states are

\[
x_1 = [B_x]_g \\
x_2 = [B_y]_g \\
x_3 = [\theta_B]_g.
\]

Consider figure 3.5, from the triangle formed by \(d_2 - d_1\), \(L'\) and the object,

\[
\begin{align*}
\theta &= [\theta_B]_g = x_3 \quad (3.23) \\
\theta &= \arctan \left( \frac{d_2 - d_1}{s_{2x} - s_{1x}} \right). \quad (3.24)
\end{align*}
\]

The equality of equation (3.23) can be derived from the geometry, but determining the signs of the angles is more involved. By definition, the angle between two frames is the angle required to turn the reference frame until it has the same direction as the other frame. The sense is positive for a counter clockwise rotation. From equation (3.24), the sign of \(\theta\) is determined by the sign of \(d_2 - d_1\). The object-gripper configuration shown in figure 3.5 corresponds to a positive \([\theta_B]_g\) which translates to \(d_2 > d_1\) and from equation (3.24) a positive \(\theta\). For the configuration where \([\theta_B]_g\) is negative, we have \(d_2 < d_1\) and again equation (3.24) gives a negative \(\theta\). Hence, the equality of equation (3.23) is verified. Note here that as far as the sensor model is concerned, the sign of the angle \(\theta\) is irrelevant because the model shows symmetry about \(\theta = 0\).

Now from the geometry of the figure we have,

\[
\begin{align*}
x_1 &= s_{3x} - (d_3 + \overline{AC}) \quad (3.25) \\
x_2 &= s_{3y} + \overline{CB}, \quad (3.26)
\end{align*}
\]

where \(\overline{AC}\) and \(\overline{CB}\) are to be determined.

From the right-angle triangle \(CBA\) we have,

\[
\begin{align*}
\angle CBA &= \theta \\
\overline{AC} &= \overline{AB} \sin(\theta) \\
\overline{CB} &= \overline{AB} \cos(\theta),
\end{align*}
\]
and the problem comes down to finding $\overline{AB}$.
If we define $\text{Pro}_{A,B}(\overline{PQ})$ as the projection of $\overline{PQ}$ on the line formed by the two points $A$ and $B$ then the segment $\overline{AB}$ is equal to,

$$\overline{AB} = \text{Pro}_{A,B}(d_2) + \text{Pro}_{A,B}(s_{2y} - s_{3y}) - \text{Pro}_{A,B}(d_3). \quad (3.27)$$

Substituting the above equation,

$$\overline{AB} = d_2 \cos(\theta) + (s_{2y} - s_{3y}) \cos(\theta) - d_3 \sin(\theta).$$

Now that we have an expression for $\overline{AB}$, then substituting up the chain,

$$x_1 = s_{3x} - (d_3 + [d_2 \cos(x_3) + (s_{2y} - s_{3y}) \cos(x_3) - d_3 \sin(x_3)] \sin(x_3)) \quad (3.28)$$

$$x_2 = s_{3y} + [d_2 \cos(x_3) + (s_{2y} - s_{3y}) \cos(x_3) - d_3 \sin(x_3)] \cos(x_3) \quad (3.29)$$

$$x_3 = \arctan \left( \frac{d_2 - d_1}{s_{2x} - s_{1x}} \right). \quad (3.30)$$

Expanding and rearranging the above equations we get,

$$x_1 = s_{3x} - d_3 \cos^2(x_3) - (d_2 + s_{2y} - s_{3y}) \cos(x_3) \sin(x_3) \quad (3.30)$$

$$x_2 = s_{3y} - d_3 \sin(x_3) \cos(x_3) + (d_2 + s_{2y} - s_{3y}) \cos^2(x_3) \quad (3.31)$$

$$x_3 = \arctan \left( \frac{d_2 - d_1}{s_{2x} - s_{1x}} \right), \quad (3.32)$$

which gives a relation between $x_1$, $x_2$ and $x_3$, and $d_1$, $d_2$, $d_3$. However, for the Jacobian calculation, we need to find $d_1$, $d_2$, $d_3$ as functions of $x_1$, $x_2$ and $x_3$, so multiplying equation (3.30) by $\tan(x_3)$ and subtracting equation (3.31) we get an expression for $d_2$,

$$d_2 = x_2 + (s_{3x} - x_1) \tan(x_3) - s_{2y}. \quad (3.33)$$

Notice that because of geometric constraints $x_3$ cannot reach $\pm \pi/2$ while all the three sensors are still viewing the object. Consequently, $\tan(x_3)$ is never zero which justifies the validity of multiplying by $\tan(x_3)$. 

In a similar fashion, multiplying equation (3.31) by \( \tan(x_3) \) and adding it to equation (3.30) gives,
\[
d_3 = -x_1 + (s_{3y} - x_2) \tan(x_3) + s_{3x}.
\] (3.34)
Substituting \( d_2 \) into equation (3.32),
\[
d_1 = x_2 + (s_{3x} - s_{2x} + s_{1x} - x_1) \tan(x_3) - s_{2y}.
\] (3.35)

Finally rewriting the above equations we have,
\[
\begin{align*}
d_1 &= x_2 + (s_{3x} - s_{2x} + s_{1x} - x_1) \tan(x_3) - s_{2y} \quad (3.36) \\
d_2 &= x_2 + (s_{3x} - x_1) \tan(x_3) - s_{2y} \quad (3.37) \\
d_3 &= -x_1 + (s_{3y} - x_2) \tan(x_3) + s_{3x} \quad (3.38)
\end{align*}
\]
where again \( s_{ix} \) and \( s_{iy} \) are the coordinates of the \( i^{th} \) sensor expressed in the \( \mathcal{G} \) frame.
The above equations are derived for the case where \( x_3 \) is positive, but the derivation has been verified to hold for the negative angles as well.

Given the above equations, the missing term of the Jacobian, namely \( \frac{\partial[p_1]}{\partial[x]} \), can readily be calculated as:

\[
\frac{\partial[p_1]}{\partial[x]} = \begin{bmatrix}
\tan(x_3) & 0 & 0 & (s_{3x} - s_{2x} + s_{1x} - x_1) \sec^2(x_3) & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]
\[
\frac{\partial[p_2]}{\partial[x]} = \begin{bmatrix}
\tan(x_3) & 0 & 0 & (s_{3x} - x_1) \sec^2(x_3) & 0 \\
0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]
\[
\frac{\partial[p_3]}{\partial[x]} = \begin{bmatrix}
-1 & 0 & -\tan(x_3) & 0 & (s_{3y} - x_2) \sec^2(x_3) & 0 \\
0 & 0 & 0 & 0 & 1 & 0
\end{bmatrix}
\]

Multiplying the three matrices yields the Jacobian,
\[
H = \begin{bmatrix}
H_1 \\
H_2 \\
H_3
\end{bmatrix}
\] (3.39)
where

\[
H_1 = \begin{bmatrix}
-\Omega_1 \tan(x_3) \\
0 \\
\Omega_1 \\
0 \\
\Omega_1 (s_{3x} - s_{2x} + s_{1x} - x_1) \sec^2(x_3) + \Theta_1 \\
0
\end{bmatrix}^T,
\]

\[
H_2 = \begin{bmatrix}
-\Omega_2 \tan(x_3) \\
0 \\
\Omega_2 \\
0 \\
\Omega_2 (s_{3x} - x_1) \sec^2(x_3) + \Theta_2 \\
0
\end{bmatrix}^T,
\]

\[
H_3 = \begin{bmatrix}
-\Omega_3 \\
0 \\
\Omega_3 \tan(x_3) \\
0 \\
\Omega_3 (s_{3y} - x_2) \sec^2(x_3) + \Theta_3 \\
0
\end{bmatrix}^T.
\]

\[\Omega_i = \frac{-\beta_{1,i} \beta_{3,i} \lambda \cos(\beta_{2,i} x_3)}{(d_i + \beta_{4,i})^{\beta_{3,i} + 1}},\]

\[\Theta_i = \frac{-\beta_{1,i} \beta_{2,i} \lambda \sin(\beta_{2,i} x_3)}{(d_i + \beta_{4,i})^{\beta_{3,i}}},\]

and,

\[d_1 = +x_2 + (s_{3x} - s_{2x} + s_{1x} - x_1) \tan(x_3) - s_{2y}\]

\[d_2 = +x_2 + (s_{3x} - x_1) \tan(x_3) - s_{2y}\]

\[d_3 = -x_1 + (s_{3y} - x_2) \tan(x_3) + s_{3x}.\]
Finally, the sensor model equation can be written as:

\[ y(k) = H(x(k)) \delta x + v, \] (3.40)

where \( H \) has been calculated and \( v \) is assumed to be stationary, Gaussian and not correlated in time. Mathematically this translates to:

\[ E[v(n) v(m)] \approx \begin{cases} R & n = m \\ 0 & n \neq m \end{cases} \] (3.41)

where \( R \) is the diagonal measurement covariance matrix.

### 3.2.4 Observability Analysis

The concept of observability addresses the ability of the sensor to capture the dynamic behaviour of the system. If the system is unobservable, then the sensor output cannot provide any useful information about the states of the object. For our system, we start by presenting the general form of the observability matrix,

\[ O = \begin{bmatrix} H \\ H\Phi \\ H\Phi^2 \\ H\Phi^3 \\ H\Phi^4 \\ H\Phi^5 \end{bmatrix}, \] (3.42)

where \( H \) and \( \Phi \) are the Jacobian and state transition matrices, respectively. Since the Jacobian is a function of the state, the system could become unobservable in certain object-gripper configurations. Before we substitute the values of \( H \), and \( \Phi \) in equation (3.42), we need to write the \( H \) matrix in a more compact form,

\[ H = \begin{bmatrix} M_1 & 0 & N_1 & 0 & F_1 & 0 \\ M_2 & 0 & N_2 & 0 & F_2 & 0 \\ M_3 & 0 & N_3 & 0 & F_3 & 0 \end{bmatrix}, \] (3.43)
where

\[ M_i = -\Omega_i \tan(x_3) \]

\[ N_i = \Omega_i \]

\[ F_i = \Omega_i (s_{3x} - s_{2x} + s_{1x} - x_1) \sec^2(x_3) + \Theta_i, \]

and the variables \( \Theta \) and \( \Omega \) are defined in section (3.2.3).

Substituting the above value of \( H \) and the value of \( \Phi \) in equation (3.14), we get the observability matrix for our system,

\[
O = \begin{bmatrix}
M_1 & 0 & N_1 & 0 & F_1 & 0 \\
M_2 & 0 & N_2 & 0 & F_2 & 0 \\
M_3 & 0 & N_3 & 0 & F_3 & 0 \\
M_1 & M_1 \delta t & N_1 & N_1 \delta t & F_1 & F_1 \delta t \\
M_2 & M_2 \delta t & N_2 & N_2 \delta t & F_2 & F_2 \delta t \\
M_3 & M_3 \delta t & N_3 & N_3 \delta t & F_3 & F_3 \delta t \\
M_1 & 2M_1 \delta t & N_1 & 2N_1 \delta t & F_1 & 2F_1 \delta t \\
M_2 & 2M_2 \delta t & N_2 & 2N_2 \delta t & F_2 & 2F_2 \delta t \\
M_3 & 2M_3 \delta t & N_3 & 2N_3 \delta t & F_3 & 2F_3 \delta t \\
M_1 & 3M_1 \delta t & N_1 & 3N_1 \delta t & F_1 & 3F_1 \delta t \\
M_2 & 3M_2 \delta t & N_2 & 3N_2 \delta t & F_2 & 3F_2 \delta t \\
M_3 & 3M_3 \delta t & N_3 & 3N_3 \delta t & F_3 & 3F_3 \delta t \\
M_1 & 4M_1 \delta t & N_1 & 4N_1 \delta t & F_1 & 4F_1 \delta t \\
M_2 & 4M_2 \delta t & N_2 & 4N_2 \delta t & F_2 & 4F_2 \delta t \\
M_3 & 4M_3 \delta t & N_3 & 4N_3 \delta t & F_3 & 4F_3 \delta t \\
M_1 & 5M_1 \delta t & N_1 & 5N_1 \delta t & F_1 & 5F_1 \delta t \\
M_2 & 5M_2 \delta t & N_2 & 5N_2 \delta t & F_2 & 5F_2 \delta t \\
M_3 & 5M_3 \delta t & N_3 & 5N_3 \delta t & F_3 & 5F_3 \delta t
\end{bmatrix}
\]  

Observability theory states that if the rank of \( O \) is less than the number of states being observed, the system is unobservable [1]. Therefore, we have to analyse the rank of \( O \).
Let us divide the observability matrix into 6 \((3 \times 6)\) sub-matrices. It is observed that the third, fourth, fifth and sixth sub-matrices are all linear combinations of the first and the second sub-matrix. Therefore, by subtracting the first sub-matrix from the second and performing some basic linear algebra row reduction, we can rewrite the observability matrix as

\[
O = \begin{bmatrix}
M_1 & 0 & N_1 & 0 & F_1 & 0 \\
M_2 & 0 & N_2 & 0 & F_2 & 0 \\
M_3 & 0 & N_3 & 0 & F_3 & 0 \\
\quad & 0 & M_1 \delta t & 0 & N_1 \delta t & 0 & F_1 \delta t \\
\quad & 0 & M_2 \delta t & 0 & N_2 \delta t & 0 & F_2 \delta t \\
\quad & 0 & M_3 \delta t & 0 & N_3 \delta t & 0 & F_3 \delta t \\
\quad & 0 & 0 & 0 & 0 & 0 & 0 \\
\quad & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\quad & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}.
\]  
(3.45)

The above matrix is composed of three sub-matrices; the first three rows, the second three rows, and the last twelve rows full of zeros. Since these rows don't change the rank of the matrix they can be simply dropped. Moreover, since the entries in the two matrices are similar (the entries of the second matrix are the same as the first but multiplied by \(\delta t\) and the columns are shifted to the right), it is sufficient to analyse only one sub-matrix. The rank of the \(O\) will be 2 times the rank of the first sub-matrix.

Consider the first sub-matrix,

\[
\begin{bmatrix}
M_1 & 0 & N_1 & 0 & F_1 & 0 \\
M_2 & 0 & N_2 & 0 & F_2 & 0 \\
M_3 & 0 & N_3 & 0 & F_3 & 0 \\
\end{bmatrix}.
\]  
(3.46)

To guarantee a rank of 3, the following conditions must be met,

\[
M_3 N_2 - M_2 N_3 \neq 0
\]  
(3.47)
\[ M_3(N_2F_1 - N_1F_2) + M_2(N_1F_3 - N_3F_1) + M_1(N_3F_2 - N_2F_3) \neq 0. \quad (3.48) \]

Substituting for equation (3.47), we get
\[ x_3 \neq \pm \frac{\pi}{4} \text{rad}. \quad (3.49) \]

This condition of observability is always met because due to geometric constraints of the object and the gripper; the object cannot be at \( \theta = \pm \frac{\pi}{4} \text{rad} \) and still be viewed by all three sensors simultaneously. In other words, the object would collide with the gripper before it is at \( \pm \frac{\pi}{4} \text{rad} \) relative to the latter.

As for the second condition, substituting in symbolically and simplifying leads to,
\[ \Omega_1\Omega_2\Omega_3(S_{3y} - x_2 - S_{1z} + S_{2x}) + \cos^2(x_3)(\Theta_3\Omega_1\Omega_2 + \Theta_2\Omega_1\Omega_3 - \Theta_1\Omega_2\Omega_3) \neq 0. \quad (3.50) \]

The physical significance of the above equation is not readily understood; however, this condition is always checked on line and never occurred throughout all the experiments.

The analysis of the observability matrix reveals that the system is observable if three conditions are satisfied simultaneously; first the object is viewed by all three sensors. Second, the angle between the object's and the gripper’s reference frames is not equal to \( \pm \pi/4 \text{rad} \). Third, equation (3.48) is satisfied.

### 3.2.5 Extended Kalman Filter Equations

Finally the extended Kalman filter equations for our system are:

**Prediction**
\[
\hat{x}(k + 1|k) = \Phi \hat{x}(k) \\
P(k + 1|k) = \Phi P(k) \Phi^T + Q
\]

**Kalman gain**
\[
K = P(k + 1|k)H(k + 1)^T[H(k + 1)P(k + 1|k)H(k + 1)^T + R]^{-1} \quad (3.51)
\]
Update

\[
\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K[y(k+1) - \hat{y}(k+1|k)]
\]
\[
P(k+1|k+1) = [I - KH(k+1)]P(k+1|k)[I - KH(k+1)]^T + KRK^T,
\]

where \( y(k+1) \) is the measured sensor output (equation 3.40) and \( \hat{y}(k+1|k) \) is the predicted sensor output:

\[
\hat{y}(k+1|k) = H(\hat{x}(k+1|k))\delta x.
\] (3.52)

For the first time the Kalman filter is run \((k = 0)\), the prediction part of the filter requires an initial value for \( x \) and \( P \). Moreover, the \( Q \) and \( R \) matrices need to be specified initially, and their value do not change with time. These and other filter implementation issues are discussed in the following section.

### 3.3 Filter Implementation Issues

To implement the extended Kalman filter, many practical issues concerning the filter were dealt with and analyzed. In this section, we address these issues.

#### 3.3.1 State Vector Initialization

Choosing good initial values for the states is a key element for the convergence of the extended Kalman filter. A good initialization of the state vector (close to actual state) will "help" the filter to converge fast, while a poor initialization could cause the filter to diverge. We have no previous knowledge about the object's states before it is seen by the sensors. Therefore, a method has to be developed to initialize the states from the sensor output.

The idea of the initialization method is to use the sensor output and some of the geometric relationships obtained in section 3.2.3 to calculate the sensor parameters from which the states can be readily calculated (equations (3.30, ..., 3.32)).
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

The $i^{th}$ sensor model relating the output to the sensor parameters is:

$$h_i(d_i, \theta) = \frac{\lambda \beta_{i,i}}{(d_i + \beta_{i,i})^3 \cos(\beta_{2,i} \theta)}$$

If we recall from section 3.2.3, equation (3.24) presents a relationship between $\theta$ and $d_1$ and $d_2$,

$$\theta = \arctan \left( \frac{d_2 - d_1}{s_{2x} - s_{1x}} \right).$$

The above four equations (three sensor outputs and the geometric relation equation) contain four unknowns $(d_1, d_2, d_3, \theta)$. The exact analytical solution involves a complicated cubic equation. An analytical solution was not attempted because very good results were obtained using an iterative method.

In the first iteration we assume that the angle $\theta$ is equal to zero. Using this information, we solve for $d_i$ given the $i^{th}$ sensor output. This is done for sensors 1 and 2 and the two distances $(d_1, d_2)$ are calculated.

In the second iteration, we use the values of $(d_1, d_2)$ in equation (3.54) to find an estimate of $\theta$. This estimate is then used in equation (3.53), and a better estimate of the distances is obtained. The estimates obtained after the second iteration are good enough that no further iteration is required.

Table (3.1) shows an example of using this approach. It is observed that after only two iterations, we have a very good estimation of the distances and the angle. Note that angles larger than $30^\circ$ are not possible because by then the object would have collided with the gripper.

Given these estimates of the distances and the angle, we can calculate an estimate of the object's position and orientation using the geometric relations derived earlier in section 3.2.3. The object's velocities are initialized to zero.

### 3.3.2 Filter Fine Tuning

The extended Kalman filter calculates the best estimate of the present states as a weighted linear combination of the predicted states obtained from the dynamic model,
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

Table 3.1: Initial estimation of the sensors parameters. The actual distances are provided based on the sensor model. After only two iterations the estimates are reasonably close to the actual values.

<table>
<thead>
<tr>
<th>Sensor parameter</th>
<th>Actual value</th>
<th>Value after 1st iteration</th>
<th>Value after 2nd iteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta$ (deg)</td>
<td>20</td>
<td>0</td>
<td>20.78</td>
</tr>
<tr>
<td>$d_1$ (mm)</td>
<td>5</td>
<td>5.56</td>
<td>4.95</td>
</tr>
<tr>
<td>$d_2$ (mm)</td>
<td>26</td>
<td>27.46</td>
<td>25.87</td>
</tr>
<tr>
<td>$d_3$ (mm)</td>
<td>10</td>
<td>10.57</td>
<td>9.95</td>
</tr>
</tbody>
</table>

and the present measured states obtained from the sensor output. Fine tuning the filter changes the weight.

Fine tuning the filter is done by initializing the measurement covariance matrix $R$, and the object model covariance matrix $Q$. Increasing $Q$ relative to $R$ means that we have less confidence in the object model and more confidence in the sensor measurement. As a result, the filter gains will increase, thereby weighting the measurement more heavily; this is verified intuitively since it is better to rely less on the noisy object model and more on the less noisy measurement updates. On the other hand, decreasing $Q$ compared to $R$ indicates that the measurements are more noisy than the prediction from the object’s model. This will decrease the Kalman gain so that the new state is more dependent on the less noisy object model. A comparison between two filters with different $R$ values is presented in figures 3.9 and 3.10 with the experimental results section.

Experiments were performed with $R = \text{diag}(4, 4, 4)$, and $Q = \text{diag}(0.001, 0.02, 0.001, 0.02, 0.001, 0.02)$ unless otherwise specified.
3.3.3 State Covariance Matrix Behaviour

The state covariance matrix, $P$, weights the states by their uncertainty; the elements of the diagonal matrix are the covariances of the states. For problems with time-invariant system models and stationary statistics, the trace of $P$ tends to a positive steady state value when the filter converges [20]. Thus $P$ can be used as a measure of the progress of the filter.

Moreover, the calculation of the Kalman gain ($K$) involves inverting a matrix which is itself a function of $P$. If the $P$ matrix is badly conditioned, numerical errors could occur and cause the filter to diverge. Therefore, the condition number of $P$ provides an indication of the numerical stability of the filter.

3.3.4 Divergence and Observability Checks

Since the extended Kalman filter is running in real-time, the validity of its estimates is always verified on line. At each iteration of the filter, two checks are executed; first the two conditions of observability obtained in section 3.2.4 are checked to make sure that the system is observable. Second, the filter's output is always checked for divergence. Divergence can occur if at least one state has a value that is physically impossible, such as a state indicating the object is occupying the same space as the gripper. Note that during all the runs performed on the system, the filter never reached a state where the system became unobservable.

3.3.5 Effect of Sampling Period

The Kalman filter is running in the discrete time domain while the object is moving in the continuous time domain. The sampling period of the filter is an important factor in its performance.

The effect of the sampling period of the extended Kalman filter on the estimates of the states is examined experimentally. Figure 3.6 shows the outcome of three ex-
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

Experiments where the filter was running at different sampling periods; the first column shows the results of the filter running at 0.004 sec, the second column shows the filter running at 0.04 sec while the third one shows the filter running at 0.4 sec. The filter running at periods of 0.004 tracks the states smoothly. The one running at 0.04 seconds is not as smooth as the former. The filter running at 0.4 seconds has a very "rough" tracking and was stopped at around 2.5 seconds. The filter diverged based on the tests of divergence performed at each iteration (section 3.3.4). The estimated $x_2$ component of position was a physically impossible value and indicated that the object had penetrated the gripper. From the figure we conclude that the performance of the filter degrades with larger sampling period.

The sampling period of the Kalman filter was fixed to 0.004 for all experiments unless otherwise specified.

3.4 Discussion of Results

3.4.1 Filter Performance

Object localization experiments were performed to test the Extended Kalman filter. One of the experiments involved bringing the object in front of the sensor where it remained stationary for 2 seconds, then moved around for 4 seconds and brought back to rest for 2 seconds. The results are presented in figures 3.7 and 3.8. Figure 3.7 shows a comparison between the actual object states and the estimated ones. The actual trajectories are shown as solid lines while the estimated trajectories are shown as dashed lines. Figure 3.8 shows the error plots of the various states.

The error plot has interesting results; the error in $x_1, x_2, x_3$ does not go to zero at steady state conditions. What is even more interesting is the fact that the value of the steady state error for the $x_2$ component, changed from -1 mm for the first 2 seconds to -0.25 mm for the last two seconds. From figure 3.7 we observe that during the first two seconds, the object was at a distance of 34 mm from the gripper while it
Figure 3.6: The effect of sampling period on the filter performance. The set of plots on the left show the results for the filter running at 0.004 seconds, the one in the middle for 0.04 seconds and the one on the right for 0.4 seconds. The solid lines show the actual trajectory of the object while the dashed line show the estimated trajectory.
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

approached to 20 mm for the last 2 seconds. This observation means that the steady state error decreased when the distance between the object and the gripper (or the sensors) decreased.

Referring back to figure 3.4 in section 3.2.2, we observe that the error in the sensor model decreases as the distance between the object and the sensor decreases. Hence, there is a possibility that the non zero steady state error is due to the sensor model.

Moreover, all the geometric relations between the object and the sensors are derived based on the assumption that the sensor projects a point on the object. In reality, the sensor paints a circle on the object where the radius decreases as the distance between the two decreases [15]. Therefore, this assumption is more valid for small distances than for large distances.

Note that the error in the $x_1$ component of pose does not exhibit the same behaviour at steady state. A possible reason could be that the difference in $x_1$ in the beginning and the end of the experiment (7 mm) is not as large as that in $x_2$ (15 mm). It is believed that the change of 7 mm is not large enough to make the difference noticeable.

The second interesting thing to note in figure 3.8 is that the error in velocity is about an order of magnitude higher than that in the position. Moreover, the highest error occurs when the object changes direction of velocity (accelerates), then it drops off to smaller values when the object is at steady state conditions. This result is logical because the dynamic model assumes that the object is at steady state conditions, and accelerations are modeled as noise. The effect of acceleration (unmodeled noise) is more pronounced on the velocity (first integral of acceleration) than it is on position (second integral of acceleration).

Moreover, the maximum error in position is about 3 mm and occurs instantaneously. Similarly, the maximum error in velocity is about 30 $mm/sec$ and also occurs instantaneously. These errors are acceptable given that the exact object's dynamics and inputs are unknown. A better result could be obtained if they were known, but we have demonstrated that the approach works for the "worst case scenario", that is
Figure 3.7: Object states estimates using the extended Kalman filter. The actual object trajectory is plotted as solid lines while the estimated object trajectory is plotted as dashed lines.
Figure 3.8: Error in state estimate using the extended Kalman filter
unknown object's dynamics and inputs.

### 3.4.2 Filter Fine tuning

The effects of tuning the Kalman filter is demonstrated experimentally; figures 3.9 and 3.10 show the actual and the estimated states for two Kalman filters. One filter has \( R_1 = \text{diag}(4, 4, 4) \) while the second has \( R_2 = \text{diag}(4000, 4000, 4000) \). For both filters \( Q \) is set to \( \text{diag}(0.001, 0.02, 0.001, 0.02, 0.001, 0.02) \). In the first case, the filter converges very fast (0.05 sec) and the tracking performance is very good. In the second case, the filter convergence is much slower (0.5 sec) and the lag of the estimated state is more pronounced. What is not shown in the figure is that for noisier sensors, the steady state performance of the filter in case 2 would have been much better than that of the first case. In our case, the sensor noise is not very large and that is the reason why it did not show in the figure.

### 3.4.3 Autonomous Object Localization

This section demonstrates the ability of the extended Kalman filter to be used in autonomous object localization which is a first step in autonomous grasping. The extended Kalman filter is started autonomously on-line in the following fashion; initially, the object is moved around in space. Meanwhile the method explained in section 3.3.1 is used to find a "rough" estimate of the pose of the object from the sensor's output. If the estimate indicates that the object has entered the area where it can be viewed by all sensors, then the system becomes observable and the extended Kalman filter can be started. The latest estimate of the object's state is used to initialize the state matrix of the filter.

An example of such an experiment is presented in figure 3.11. The figure shows four plots. The first plot in the figure is the trace of \( P \) which is used here to show the time the filter is started. In the next three plots, a comparison between the object's actual and estimated pose is presented. The actual trajectories are shown as solid lines and
CHAPTER 3. OBJECT LOCALIZATION: EXTENDED KALMAN FILTER

Figure 3.9: The effect of changing the $R$ on the filter performance. For the column of plots to the left has $R_1 = \text{diag}(4, 4, 4)$, while that to the right has $R_2 = 1000 \times R_1$. 
Figure 3.10: The effect of changing the $R$ on the filter performance. For the column of plots to the left has $R_1 = \text{diag}(4, 4, 4)$, while that to the right has $R_2 = 1000 \times R_1$. 
the estimated trajectories are presented as dashed lines. Initially at time 0 sec, the object is far from the gripper. During the first 5 seconds, the object is brought closer to the gripper. Meanwhile the sensor output is monitored and a "rough" estimate of the object's position is calculated based on the method explained in section 3.3.1. At time = 5 seconds, the "rough" estimate of the object's state indicate that it has entered the area where it can be viewed by all sensors. This condition is checked on line by ensuring that both the estimated object's position and orientation are reasonable and lie within certain limits. For example, a reasonable estimate of the location of the object provided by the sensors can not include a location which corresponds to a distance that is larger than the range of the sensors.

At time = 5 seconds the latest estimate of the object's state are used to initialize the state matrix of the extended Kalman filter and the latter is started. For the period from 5 seconds to 15 seconds, the filter tracks the object.

The figure shows some interesting results for the autonomous initialization of the filter; during the first 5 seconds, the estimates of the $x_1$ and $x_2$ states show that the object is far away. Moreover, all the estimates are very noisy because for distant objects, the sensor output is low and thus the signal to noise ratio becomes large. As the object comes closer to the gripper, but not yet in the area where all three sensors can view it, the $x_2$ estimate starts converging to the real value. However, the estimate of the $x_1$ component is still indicating a far object. When both the $x_1$ and $x_2$ estimate are reasonable, the $x_3$ estimate is checked. When all three estimates are judged reasonable, the extended Kalman filter is started.

3.5 Conclusion

We have presented and validated experimentally one basic tool that is needed for autonomous grasping of dynamic objects, namely object localization. In particular, we have demonstrated that an extended Kalman filter running in real-time and using infra-red proximity sensors as sensing devices can be autonomously started to localize
Figure 3.11: An experiment showing autonomous object localization. The top plot shows the trace of $P$ which gives an indication of when the filter is started. Before time $= 5$ seconds the "rough" estimate of the object's states indicates that the system is unobservable. At time $= 5$ seconds the object becomes in view of all sensors, the system becomes observable, and the Kalman filter is started automatically.
a moving object. An estimate of the states of the object is checked. Once the output indicates that the object’s location corresponds to an observable system, the extended Kalman filter is started. The extended Kalman filter then tracks the six states of the object with errors less than 3 mm for position and 30 mm/sec for velocities. These results are obtained assuming no previous knowledge about the object dynamics.
Chapter 4

Object Grasping: Heuristic Approach

4.1 Introduction

In this chapter a heuristic approach is developed for non-smooth grasping of a stationary object. The objective of the approach is to bring the end effector into a position where the object can be grasped, without colliding with the object or losing it.

The concept of the heuristic approach is to use a network of binary sensors, which simply indicate the presence or absence of the object, to divide the area in front of the gripper into smaller areas or domains. A domain is defined as an enclosed area in which the object can move while still producing the same binary response from all the sensors. Each domain is assigned a confidence level which is a relative indication of how close the object is to being grasped and how many sensors are monitoring it. The process behind the choice of confidence levels is explained in section 4.2.

Autonomous grasping is achieved as follows; once the object is “spotted” in one domain, the gripper moves to bring the object into a domain of equal or higher confidence. This sequence is continued until the object reaches the domain of highest confidence where it is grasped.
The conceptual formulation of the approach is presented in section 4.2 and the implementation of the approach is presented in section 4.3.

4.2 Conceptual Formulation of the Approach

Before proceeding to the details of this approach, we state the assumptions that were made on the object and the sensors.

4.2.1 Assumptions

Figure 4.1: Planar view of the tool interface.

Figure 4.1 shows a sketch of the object and the gripper. The gripper is assumed to move only in the planar x and y direction. Considering the gripper's reference frame, the object's position can be determined by knowing the coordinates of any point on it. For example, point P is picked as the point to represent the object as this choice
will simplify the later analysis.

The gripper has two parallel jaws which can move from 0 mm to 35 mm apart. The STM sensors, discussed in detail in chapter 2, are used in the binary mode; if the output is above a certain threshold, the binary output is 1 (object exists) and vice versa. A cheaper proximity sensor with a binary output could equally have been used.

Moreover, for the purpose of simplifying the conceptual formulation of the approach, the field of vision of each of the sensor is assumed to be a straight line. In other words, the LED emits an incident light beam that paints a point on the object. The effects of making this simplification will be discussed in section 4.3. Finally, each sensor is assumed to have a fixed range of 50 mm. This range limitation is achieved by choosing an appropriate threshold.

4.2.2 The Sensor Network

The proposed sensor network is presented in figure 4.2. There is a total of 4 sensors used; two sensors are placed in the vertical direction, parallel to the gripper jaws, while the other two are placed at 45 degrees with respect to the vertical. The dotted lines coming from the sensors represent the field of vision of each of the sensors, while the length of the line is an indication of the range.

The sensor domains corresponding to this sensor configuration are shown in figure 4.3. The various dashed areas correspond to different domains, and are labelled according to the number of sensor that sees the object when it is in that domain. For example, consider the rectangle on the top left hand side, labelled $D_1$. If $P$, the point representing the object, happens to be within the boundaries of the rectangle, then the object is detected only by sensor number 1. The object can be anywhere in $D_1$ and the response from all the sensors stays the same; sensor 1 indicates the presence of the object while sensors 2, 3, and 4 indicate its absence. The domain $D_{1234}$ corresponds to the object's position being centered and well penetrated into the
gripper. This domain is the ideal domain in which the object can be grasped.

The white region along the sides of the jaws and below regions $D_{134}, D_{234}$ and $D_{1234}$ is defined as the physically impossible region. This definition is because in that region the object and the jaws are occupying the same space. This region comes about due to representing the object as a point. When the object collides sideway with a pad, the point P is still 1/2 the width of the object away from the pad. Thus, the physically impossible region is simply a side effect of this representation and does not affect the discussion.

Figure 4.2: The proposed sensor network

The reasoning behind the placement of the sensors is as follows; sensors 1 and 2 are needed to detect the presence of the object in front of the gripper. The placement
Figure 4.3: Idealized sensor domains for the proposed sensor configuration
of sensors 3 and 4 at 45° serves a dual purpose. First, the sensors provide better coverage in the front of the gripper and second they provide detection of the "T" shaft between the gripper pads.

Having sensors 3 and 4 at 45° is desirable; mounting these sensors at angles less than 45° causes the domains $D_3$ and $D_4$ to move down closer to the gripper and hence the area in front of the gripper which is covered by at least one sensor is reduced. On the other hand, mounting sensors 3 and 4 at angles greater than 45° produces domains which are not connected. In other words, there will be gaps between the domains in which the object is not detected. These gaps can be eliminated by bringing the gripper jaws closer together. However, this action will reduce the total area in front of the gripper which is covered by at least one sensor, and consequently reduce the chance to initially view the object. To maximize this chance, the sensors 3 and 4 have to be set at the optimal angle of 45° and the distance separating the gripper jaws should be $50 \times \cos(45°) = 35.4$ mm.

The main purpose of having all these domains is to divide the space in front of the gripper into various regions where each region can be assigned a certainty or confidence level. The confidence level is chosen as the sum of two variables. The first variable is the number of sensors that can view the object in a certain domain. The second variable is a measure of the "closeness" of the object to the domain where it can be grasped ($D_{1234}$). The domain which is the "furthest" from $D_{1234}$ is given a proximity level of 0, and the other domains are assigned values accordingly. Table (4.1) shows how each domain is assigned a confidence level. For example, an object in $D_1$ is viewed by one sensor. Moreover, $D_1$ is one of the furthest domains from $D_{1234}$, so the measure of "closeness" is set to 0. The confidence level, being the sum of the two variables is 1. On the other hand, the $D_{1234}$ is domain where the object is grasped so it is assigned the highest closeness level, and there are four sensors monitoring the object. As such, the confidence level in $D_{1234}$ is set very high. Note that domains $D_4$, $D_{14}$, $D_3$, and $D_{23}$ have equal confidence levels.
CHAPTER 4. OBJECT GRASPING: HEURISTIC APPROACH

The confidence level is only a relative measure to determine which domains are favoured over the others. This information is then used to decide what action should be taken by the gripper to bring the object into a domain of higher confidence. The object is continuously brought into domains of equal or higher confidence until the former reaches $D_{1234}$ where it is grasped.

The gripper commands that result in the object moving from one domain to another of equal or higher confidence are presented in figure 4.4.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Number of sensors</th>
<th>Proximity level from $D_{1234}$</th>
<th>Confidence level (sum)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$D_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$D_3$</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$D_4$</td>
<td>1</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>$D_{14}$</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>$D_{23}$</td>
<td>2</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>$D_{34}$</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>$D_{134}$</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>$D_{234}$</td>
<td>3</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>$D_{1234}$</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 4.1: The confidence level of each domain

**Initial Object Position**

Figure 4.4 shows the gripper commands per sensor domain. The arrows inside each domain point in the direction of the gripper command in that domain. The two solid lines intersecting at $90^\circ$ in the middle define the boundaries where the object would be grasped without collision. For any initial object position below the two lines, for example the lower end of $D_{14}$ or $D_{23}$, the gripper would collide with the object.
To avoid this collision from occurring, a condition is imposed. The autonomous grasping approach is not started if the object is *initially* located either in $D_{23}$ or $D_{14}$. By doing so, we guarantee that the object is always inside the boundaries of the cone.

Figure 4.4: *Gripper commands per sensor domains; the arrow in each domain points in the direction of the gripper command. The 90° cone drawn in the middle represents the boundaries of the object position outside which the approach will fail; for any object position outside the cone, the object will collide with gripper pads.*
CHAPTER 4. OBJECT GRASPING: HEURISTIC APPROACH

4.2.3 An Example

The following is an example to demonstrate the usage of the approach. The object is initially assumed to be outside any of the domains. The operator moves the gripper close to the object until the latter enters one of the domains, \( D_1, D_2, D_3, D_4, \) or \( D_{34} \) (note that \( D_{23} \) and \( D_{14} \) are excluded because some parts of them fall inside the cone). At this time, the operator is notified that the object is detect and the autonomous grasping approach is started. In particular, let us assume that the object is in \( D_2 \).

The gripper command at \( D_2 \) is to move up. The grippers motion causes the object to reach \( D_{23} \) where the command is to move at an angle of 45°. As the gripper is moving in that direction, the object is brought to domain \( D_3 \) where the gripper keeps on going in the same direction as before until the object reaches \( D_{34} \). In this domain, the gripper moves up to bring it over to \( D_{1234} \) where it is grasped.

Although this approach is derived for a stationary object, we will conduct an analysis which involves restricted object motion to demonstrate the robustness of the method. Assume that we have the object already in \( D_{1234} \) ready to be grasped. If meanwhile the object moves sideway, then it enters domains \( D_{234} \) or \( D_{134} \) and the gripper will move left or right (respectively) to bring the object back to \( D_{1234} \). Similarly, if the object backs up, it enters \( D_{34} \) and the gripper moves forward to bring it back to \( D_{1234} \).

4.3 Implementation of the approach

The STM sensors (chapter 2) are used in the binary mode; if the output is above a certain threshold, the binary output is 1 (object exists) and vice versa. A cheaper proximity sensor with a binary output could equally have been used.

The actual domains are obtained experimentally by moving the object in front of the sensors and recording their binary output as a function of the object position. The sensors' binary output is obtained by comparing each sensor output (digital 0 -
4095) to a particular threshold.

The threshold is selected for each sensor based on two requirements; first, the threshold is used to distinguish the output due to the presence of an object from that due to the base noise level. The base noise level is defined as the maximum value (in volts or ticks) that the sensor would produce when no object is placed in its field of vision. For an ideal sensor the base noise level should be 0. However, this is never achieved in reality because of imperfections in the electronics (e.g. inaccurate resistors, saturation of amplifiers ...etc). The second requirement of the threshold is to adjust the domains of the sensors so that domains $D_1$ and $D_3$ are symmetric to $D_2$ and $D_4$ respectively.

After scanning the object and recording the sensor's binary output, we end up with a 3-D plot showing the sensors binary output as a function of the objects position (x and y). Then, the contour of this 3-D plot is used as a representation of the real boundaries of the domains.

Figure 4.5 shows the real domains obtained experimentally. The origin of the plot corresponds to the location where the object is centered between the jaws and is touching the pads at sensors 1 and 2. The various regions are indicated in the figure and the physically impossible region is not shown because it is not required for the analysis.

A comparison between this figure and figure 4.3 reveals some differences; first, the boundaries of the actual domains are not as straight as those developed conceptually. In particular, $D_{23}$ shows irregular boundaries, and $D_1$ has an "island" in the top left side. These irregularities are due to the noise in the sensor measurement. Note that the irregularities occur at the limit of the range of the sensors. At that limit, the effect of noise is more pronounced than at closer distances. However, these irregularities did not influence the performance of the approach. Second the shape of the domains are different although the general trend is preserved; in particular, the edges are more "rounded" in the actual domains than those in the conceptual one. Moreover, the
actual domains are "wider" than the idealized ones. This is pronounced in domains $D_{234}$ and $D_{134}$ where the shape of the domain has changed. These differences are due both to sensor noise and the assumption that the sensor projects a point on the object. In reality, the sensor projects a circle where the center has the high intensity of light which gradually decreases with increasing radius. Consequently, if the object enters the field of vision of a sensor, the latter's output shows a gradual change and not a sudden one. Similarly, the actual domains do not exhibit sharp corners or edges which define the boundaries, but rather show a gradual change.

Given the changes in the shape of domains $D_{234}$ and $D_{134}$, we changed the gripper commands in these domains accordingly. Table (4.2) presents a comparison of the gripper commands for the idealized and the actual domains.

<table>
<thead>
<tr>
<th>Domain</th>
<th>Gripper command for idealized domains</th>
<th>Gripper command for actual domains</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>move up (90°)</td>
<td>move up (90°)</td>
</tr>
<tr>
<td>$D_2$</td>
<td>move up (90°)</td>
<td>move up (90°)</td>
</tr>
<tr>
<td>$D_3$</td>
<td>move up and left (135°)</td>
<td>move up and left (135°)</td>
</tr>
<tr>
<td>$D_4$</td>
<td>move up and right (45°)</td>
<td>move up and right (45°)</td>
</tr>
<tr>
<td>$D_{14}$</td>
<td>move up and right (45°)</td>
<td>move up and right (45°)</td>
</tr>
<tr>
<td>$D_{23}$</td>
<td>move up and left (135°)</td>
<td>move up and left (135°)</td>
</tr>
<tr>
<td>$D_{34}$</td>
<td>move up (90°)</td>
<td>move up (90°)</td>
</tr>
<tr>
<td>$D_{134}$</td>
<td>move right (0°)</td>
<td>move up and right (45°)</td>
</tr>
<tr>
<td>$D_{234}$</td>
<td>move left (180°)</td>
<td>move up and left (135°)</td>
</tr>
<tr>
<td>$D_{1234}$</td>
<td>grasp</td>
<td>grasp</td>
</tr>
</tbody>
</table>

Table 4.2: The gripper command for each domain. The commands are given in gripper local coordinates. The angles are measured from the $+x$ direction.

Presentation of Results
Figure 4.5: The actual sensor domains (obtained experimentally).
Figure 4.6: Autonomous grasping experiment case 1: object entering $D_1$. The figure is shown in the gripper's reference frame, so the object is seen as moving. The path followed by the object is shown as a dashed line. Note that the path corresponds to the relative motion of the object with respect to the gripper.
Figure 4.7: Autonomous grasping experiment case2: object entering $D_2$. The figure is shown in the gripper's reference frame, so the object is seen as moving. The path followed by the object is shown as a dashed line. Note that the path corresponds to the relative motion of the object with respect to the gripper.
Figures 4.6 and 4.7 show two typical experiments performed to demonstrate the operation of sensor domain. The figures show the sensor domains and a dashed line. The dashed line corresponds to the object's trajectory as seen by the gripper. In other words, from the gripper's point of view, the sensors and their domains are stationary and the object is moving. This configuration was used to illustrate the gripper motion in each domain. For example, in figure 4.6 the object enters domain $D_1$ at the top left. The gripper command is to move up, so the object as viewed by the gripper is moving down. Then the gripper's motion causes the object to enter domain $D_{14}$ where the gripper's motion is in the $+135^\circ$. Again this translates to the object moving in the $+45^\circ$ angle. This motion is continued even when the object enters $D_{134}$. Finally the object enters $D_{1234}$ where it is grasped by closing the gripper pads. In summary the following path is followed: $D_1, D_{14}, D_{134}, D_{1234}$. From Table (4.1), we obtain the following confidence levels (correspondingly): 1, 4, 6, 8. It is interesting to note that the object moves from a domain to another of equal or higher confidence until it is in $D_{1234}$.

Figure 4.7 shows a similar experiment whereby the object enters $D_2$ and is brought to $D_{1234}$ via the following path: $D_2, D_{23}, D_3, D_{34}, D_{234}, D_{1234}$. Looking up Table (4.1), we obtain the following confidence levels (correspondingly): 1, 4, 4, 5, 6, 8. It is observed that the technique works exactly as planned by bringing the object from one domain to another of equal or higher confidence level.

It is interesting to note that the approach does not require any scanning of the object. The moment the latter is spotted, immediate action is taken leading to grasping it.

### 4.4 Conclusion

We have developed and implemented a heuristic approach for non-smooth grasping of a stationary object. The approach is simple, fast and does not require any object scanning prior to grasping. It makes use of strategic placement of sensors and
known object geometry to divide the space around the gripper into sensing domains. Moreover, the geometry of the sensor network is chosen such that only one gripper command is required per domain to bring the object into another domain of higher confidence. This sequence is continued until the object reaches the last domain where it can be grasped.
Chapter 5

Conclusion

We have presented three tools for on-line autonomous grasping of a planar object.

First, we have multiplexed and frequency modulated commercially available infra-red proximity sensors to eliminate cross talk, enabling the sensors to be used in a network.

Second, we have used a network composed of three sensors and an extended Kalman filter to autonomously initiate the localization of the object. The filter estimates six states of the object, pose and velocities, with a maximum error of 3 mm in position and 30 mm/sec in velocity. This approach could be used for smooth grasping where the object's position and velocity are matched at time of contact.

Third, we have developed a network of four binary sensors for non-smooth grasping of a stationary object. The approach is very fast and does not require any object scanning prior to grasping.

5.1 Future Recommendations

The extended Kalman filter developed works well in localizing the object. The next logical step would be to test its performance when used for tracking the object. In other words, the filter output would be used as an input to a controller for the gripper's
CHAPTER 5. CONCLUSION

motion.

The system has only been tested on a planar testbed. An area of future focus is to extend the work into a 6 degree of freedom testbed. In particular, the Jacobian derivations could be challenging.

The sensor domain approach was developed for a stationary object with a fixed known orientation. The next logical step will be extending this work to allow for localizing an object which could be at any orientation with respect to the gripper. In this case, the domain will become a volume instead of an area, and the domain analysis will have to be done in 3D space.

Moreover, a supervisory level is still needed to make decisions on various stages of autonomous grasping, and has the ability to reset the system, back off and re-attempt a grasp, if necessary.
Bibliography


