LAPPEENRANTA
UNIVERSITY OF TECHNOLOGY

Olli Alkkiomäki

SENSOR FUSION OF PROPRIOCEPTION, FORCE AND VISION IN ESTIMATION AND ROBOT CONTROL

Thesis for the degree of Doctor of Science (Technology) to be presented with due permission for public examination and criticism in the Auditorium 1382 at Lappeenranta University of Technology, Lappeenranta, Finland on the 3rd of December, 2009, at noon.

Acta Universitatis
Lappeenrantaensis
358

Supervisors Professor Ville Kyrki
Machine Vision and Pattern Recognition Laboratory
Department of Information Technology
Faculty of Technology Management
Lappeenranta University of Technology
Finland

Professor Heikki Kälviäinen
Machine Vision and Pattern Recognition Laboratory
Department of Information Technology
Faculty of Technology Management
Lappeenranta University of Technology
Finland

Reviewers Professor Danica Kragić
School of Computer Science and Communication
Royal Institute of Technology
Sweden

Professor Darius Burschka
Department of Informatics
Technische Universität München
Germany

Opponents Professor Danica Kragić
School of Computer Science and Communication
Royal Institute of Technology
Sweden

Professor Juha Röning
Department of Electrical and Information Engineering
University of Oulu
Finland

ISBN 978-952-214-835-3
ISBN 978-952-214-836-0 (PDF)
ISSN 1456-4491

Lappeenrannan teknillinen yliopisto
Digipaino 2009

Acknowledgements

This thesis draws from the research work carried out in TacVision project at Lappeenranta University of Technology (LUT), Finland, during the years 2005-2009. The project was a joint effort of Machine Vision and Pattern Recognition Laboratory in the Department of Information Technology and Mechatronics research group in the Department of Mechanical Engineering. Several people have contributed in the work and it is a pleasure to thank all those who made this thesis possible.

First I owe my deepest gratitude to my supervisors Professor Ville Kyrki and Professor Heikki Kälviäinen. This thesis would not have been possible without your guidance. For financial support Tekes and Suomen Akatemia are gratefully acknowledged.

I thank warmly my co-authors Professor Heikki Handroos and Dr. Young Liu for their contribution and expertise in mechatronics. Special thanks to Professor Huapeng Wu and the laboratory staff of the Department of Mechanical Engineering for guidance in mechatronics machinery and helping in tinkering various gadgets for the robots. Also the developers of the itlabcpp library, especially Simo Heiliö, Ville Kyrki, Janne Laaksonen and Pekka Paalanen, are thankfully acknowledged.

The project steering group meetings gave me valuable insights into the industrial aspects of robotics. I like to thank Heikki Aalto from Delfoi, Mika Laitinen from Master Automation Group and Timo Laurila from Tekes for enlightening discussions during the meetings.

I am grateful to the reviewers Professor Danica Kragić and Professor Darius Burschka for their valuable comments in improving my thesis. I also owe big gratitude to the anonymous reviewers of the numerous articles sent for peer review during the project.

I thank my colleagues and fellow students at LUT for inspiring academic and social environment and discussions during the coffee breaks and at #itlab and #lager. Also thanks to the laboratory staff and secretaries at the Department of Information Technology.

Finally, I would like to thank my family for encouraging, supportive atmosphere, and Reija, thank you for your care, patience and understanding.

Lappeenranta, 2009

Olli Alkkiomäki

Abstract

Olli Alkkiomäki
Sensor Fusion of Proprioception, Force and Vision in Estimation and Robot Control
Lappeenranta, 2009
109 p.

Acta Universitatis Lappeenrantaensis 358
Diss. Lappeenranta University of Technology

ISBN 978-952-214-835-3
ISBN 978-952-214-836-0 (PDF)
ISSN 1456-4491

Sensor-based robot control allows manipulation in dynamic environments with uncertainties. Vision is a versatile low-cost sensory modality, but low sample rate, high sensor delay and uncertain measurements limit its usability, especially in strongly dynamic environments. Force is a complementary sensory modality allowing accurate measurements of local object shape when a tooltip is in contact with the object. In multimodal sensor fusion, several sensors measuring different modalities are combined to give a more accurate estimate of the environment. As force and vision are fundamentally different sensory modalities not sharing a common representation, combining the information from these sensors is not straightforward.

In this thesis, methods for fusing proprioception, force and vision together are proposed. Making assumptions of object shape and modeling the uncertainties of the sensors, the measurements can be fused together in an extended Kalman filter. The fusion of force and visual measurements makes it possible to estimate the pose of a moving target with an end-effector mounted moving camera at high rate and accuracy. The proposed approach takes the latency of the vision system into account explicitly, to provide high sample rate estimates. The estimates also allow a smooth transition from vision-based motion control to force control. The velocity of the end-effector can be controlled by estimating the distance to the target by vision and determining the velocity profile giving rapid approach and minimal force overshoot. Experiments with a 5-degree-of-freedom parallel hydraulic manipulator and a 6-degree-of-freedom serial manipulator show that integration of several sensor modalities can increase the accuracy of the measurements significantly.

Keywords: sensor fusion, force, machine vision, robotic manipulation, extended Kalman filter, uncertainty modelling

UDC 621.865.8 : 681.532.6 : 681.532.8 : 681.527.5 : 004.93'1

Symbols and abbreviations

η	noise
τ	latency
$\boldsymbol{\phi}(\cdot)$	function converting a rotation matrix to immediate angles
$\boldsymbol{A}$	a linear mapping from image plane parameters to homogeneous plane parameters
$\boldsymbol{F}$	state transition matrix
$\boldsymbol{G}$	control input matrix
$\boldsymbol{H}_e$	Hessian
$\boldsymbol{H}$	observation matrix
$\boldsymbol{J}$	Jacobian
$\boldsymbol{K}$	Kalman gain
$\boldsymbol{P}$	covariance of the state
$\boldsymbol{Q}$	covariance of the process noise
$\boldsymbol{R}(\cdot)$	function converting immediate angles to a rotation matrix
${}^{W}\boldsymbol{R}_{O}$	rotation from the world frame to the object frame
$\boldsymbol{S}$	covariance of the measurement
${}^{EE}\boldsymbol{T}_{C}$	transformation from the end-effector frame to the camera frame

1-D	1-dimensional
2-D	2-dimensional
3-D	3-dimensional
CAD	computer aided design
CCD	charge-coupled device
DOF	degree-of-freedom
EKF	extended Kalman filter
IBVS	image-based visual servoing
KF	Kalman filter
P	proportional
PI	proportional-integral
PBVS	position-based visual servoing
RGB	red green blue
SO(3)	space of orientations

CONTENTS

Chapter I

Introduction

> I purpose to show that, however impossible it may now seem, an automaton may be contrived which will have its “own mind,” and by this I mean that it will be able, independent of any operator, left entirely to itself, to perform, in response to external influences affecting its sensitive organs, a great variety of acts and operations as if it had intelligence.
>
> Nikola Tesla, *Century Illustrated Magazine* June 1900

1.1 Background

Industrial robots have long been used in heavy, repetitive tasks to produce the same routine again and again without variation. Car industry is a good example of traditional robotized mass production where large series of identical products are manufactured. When the size of the series is big, the robot programming costs per produced unit are low. However, if smaller series or customized products are produced, more flexible robot control is desirable.

One of the greatest challenges in robotics is to make the robots work in a robust way in an uncertain environment. Sensor-based robotics attempts to answer this question by allowing the robot to react to the feedback from the world immediately, instead of requiring a perfect model of the world which would allow perfect planning of actions in advance. Simple motion control is not adequate, when the position of the workpiece is unknown during manipulation, as the uncertainties present in the task prevent the robot from following a preprogrammed trajectory. Sensor-based manipulation allows a robot to adapt to a dynamic and uncertain environment which changes over time, independent of the actions of the robot. With sensors, the uncertainties of the environment can be modeled and the robot can take actions based on the sensory input. Sensor-based automation was envisioned long before the first industrial manipulators, but sensor-based robotics has become feasible only recently. If the robot can react to its surroundings and change its behavior on the basis of sensory input, a wide variety of tasks can be performed. Sensor-based robotic manipulation promises increase in the productivity, flexibility, and robustness of manipulation.

Visual sensing allows the robot to examine its surroundings and adapt to the environment. Force offers a complementary sensory modality allowing accurate measurements when in contact with the object. In multimodal sensor fusion, several sensors measuring different modalities are combined to give a more accurate estimate of the environment. In Fig. 1.1, a robotic manipulator uses several sensor modalities together to examine its surroundings. Vision gives a coarse estimate of the position and rotations (pose) of the target object regardless of whether the tool is in contact with the object or not. The joint sensors of the robotic manipulator measure the pose of the end-effector and can give additional information of the object pose when the tool is in contact with the object. A force sensor mounted on the wrist of the manipulator measures contact and friction forces and can extract object properties, such as the deformability and friction coefficient. Combining visual and force sensing is currently one of the most promising approaches to sensor-based manipulation, as combining the sensors can speed up the manufacturing process and allow more accurate robot control.

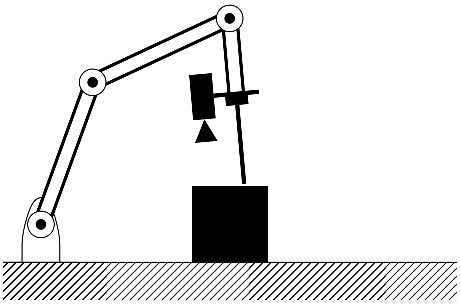


Figure 1.1: Robotic manipulator examining its environment.

A robot can adapt to an unknown environment in two ways. In hierarchical paradigm the robot observes the environment through sensors and builds a model that describes the unknown environment. After the sensors have constructed a model of the environment, a planning algorithm devises a path to follow in order to accomplish the desired action. Another approach is to react to the sensor feedback immediately through feedback control. In reactive paradigm it is possible to use very simple planning, giving only a velocity command or position increment to the robot. The two paradigms can also be composed into hybrid architecture, where knowledge of the environment is combined with reactive control [11]. In this thesis a hybrid paradigm is adopted, where the workspace of the robot is assumed to be free of obstacles, and the controller generates the path online on the basis of the multimodal sensory input. By combining information from several sensor modalities, uncertainties in the environment can be modeled and an accurate estimate of the surroundings of the robot can be created. Traditional sensor based robotics exploits sensor measurements directly in control. In this thesis, sensor fusion techniques are proposed which combine the sensor information before the control loop to achieve high-rate, accurate and robust control.

Framework models for sensor-based robot control define methods for task-based manipulation. Mason's compliance frame [67] or task frame formalism [21] presents a framework where different controllers are assigned to different translational and rotational axes of the task frame. The approach is also known as hybrid control, where different sensors control different perpendicular directions. The formalism has been in extensive use in combining different sensor modalities, such as force and vision [94, 15, 14, 13]. Task frame formalism has limitations, however. The framework does not define interaction between frames and thus only one frame, the task frame, can be used in the control. This limits the controlled axes to be perpendicular to each other.

Constraint-based programming offers a more general framework allowing several feature frames to be used simultaneously in the control. In the task function approach [86] the task is divided into priority and secondary tasks, allowing several competing control objectives to be fullfilled simultaneously. The method is often used in visual servoing, but also other sensor modalities, such as force can be fused in this framework [25]. The approach is also known as impedance control, where the weighted sum of individual task functions are combined in the control [65]. A similar formalism, constraint-based task specification [30] estimates the state, taking into account also the geometric uncertainties of the environment.

Although the above models offer great help in developing the system, they do not specify what kind of sensors are used, how they are calibrated and how the sensor uncertainties are modelled. The modelling of the uncertainties is not straightforward, especially in the case of multi-degree-of-freedom sensors, such as vision. These questions are considered in this thesis, and examples are given for different visual sensors, as well as wrist mounted 6-DOF force sensors.

The research has been conducted in the “TacVision” project, which is a part of a larger “Kitara” technology programme focusing on the Application of Information Technology in Mechanical, Civil and Automation Engineering funded by Academy of Finland and Tekes (Finnish Funding Agency for Technology and Innovation). The goal of the TacVision project is to develop methods that use both visual and tactile feedback to control robotic manipulators and increase the robustness and efficiency of robot control with integration of multiple sensors. The web pages of the project are available at [91].

1.2 Objectives

The objective of this thesis is to develop estimation methods for combined vision and force control. The focus is on estimating the uncertainties of sensor measurements and combining them in one model to provide a robust estimate of a dynamic environment with known uncertainties. It is investigated whether the estimate can be used in robotic manipulation to provide versatile and accurate robot control. Also methods for transition from vision-based control to combined vision and force control are developed. One of the challenges in force controlled manipulation is that when a machining tool first touches the object, a momentary force peak occurs. It is investigated if it is possible to reduce the force peak by estimating the position of the object and reducing the impact velocity before the contact occurs. The main research question in the thesis is whether modelling the dynamic uncertain environment and fusing several sensor modalities together in one model can provide better control stability and performance than using the sensor measurements directly in control.

1.3 Contribution and publications

The main contribution of this thesis is developing estimation methods for robotic manipulation. An unknown environment is modeled by fusing information from visual, force and robot joint sensors to provide an estimate of the pose of an object for the robot controller.

A major part of the conducted research has been published in scientific conferences and in a journal article. The research began by investigating methods for smooth transition from vision-based motion control to force control. When the end-effector of a robotic manipulator touches the object, a momentary force peak may occur. By estimating the pose of the object, the force peak can be reduced by decreasing the velocity of the robotic manipulator before the contact occurs. A method for approaching a target object quickly and taking contact with no force overshoot was published in [2]. The article describes a novel velocity profile approach, which is shown with experiments to be more efficient than the previous image-based approaches.

Robot control in dynamic environments also requires an accurate estimate of the pose of the target object. A method for tracking a moving object with an end-effector mounted moving camera has been developed using an EKF (Extended Kalman filter) model to estimate the pose of an object based on visual and proprioceptive information of the robot end-effector. Theory and simulated results for a full 6-degree-of-freedom (DOF) case and experiments with 1-DOF hydraulic servo bench are presented in [5]. The article includes a novel sensor delay compensation technique for 6-DOF pose tracking and a method for filtering rotational measurements in the EKF. Experiments with contacting the moving target and keeping a desired contact force were also conducted. Force overshoot was reduced by estimating the object pose with the previous estimation method, and the results were published in [62]. Later the visual system, as well as the experiments for moving target tracking were extended to cover a more general case [6], and also force control experiments were conducted with a 5-DOF parallel hydraulic manipulator [63].

Force offers a complementary sensor modality for determining the object pose. A method for fusing proprioceptive information of the end-effector with force and visual measurement in the EKF has been proposed in three publications [7, 3, 4]. The publications propose a novel sensor integration method where tooltip measurement is fused with visual measurement to estimate the object pose. The uncertainties of the visual measurements are modeled in detail, allowing the tooltip measurement to compensate for the uncertain pose estimate acquired by the vision sensor. The latest work introduces a structured light pose estimation method and hybrid control experiments with a 6-DOF serial manipulator [4]. The article proposes a novel laser sensor and a toolpoint calibration method essential for position-based control. The thesis also includes unpublished original work in shared control, utilizing novel estimation methods.

The author made a major contribution to the development and experimental work in [62] and participated in the development and experiments in [63]. In the rest of the papers the author is the first author and had a primary role in developing the system, writing, and conducting the experimental work [2, 5, 6, 4, 3, 7].

1.4 Outline of the thesis

The relations between the chapters presented in this thesis are shown in Fig. 1.2. Chapter 2 introduces vision-based robot control using pose estimation algorithms. Pose estimation is later used in Chapter 3, where an extended Kalman filter (EKF) combining visual and encoder information is presented, and in Chapter 4, where contact information is combined in the EKF estimate. Chapter 5 focuses on combined control integrating force and vision sensors by using the previously introduced estimation methods, and finally, Chapter 6 summarizes the achievements of this thesis, and suggests topics for future research.

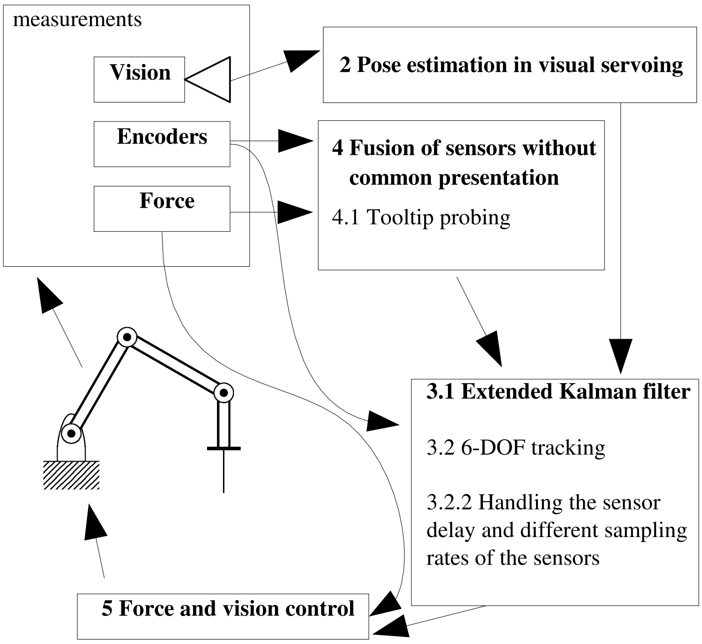


Figure 1.2: Structure of the theoretical work.

CHAPTER II

Pose estimation in visual servoing

In this chapter vision-based robot control, visual servoing is introduced. It is shown how the pose of an object can be estimated in the world coordinates and how the estimate can be used in control. First, the relative pose of the object with respect to the camera must be determined. Then the pose can be transformed to the world frame by using hand-eye calibration and the proprioceptive information of the robot end-effector. To extract the relative pose of the object with respect to the camera, model-based pose estimation methods are often used. Several methods for stable and real-time 3-D pose tracking exist in the literature. Most current 3-D pose tracking systems are based on tracking object boundaries [35, 48], but examples of appearance-based 3-D tracking [44] also exist. Methods based on non-linear pose computation [29], texture tracking [100] and model free cues [51] have been presented. The model can be a CAD-model of the object [35] or simply a model describing the 3-D coordinates of markers glued on the object. Also structured light sensors can be used to define the pose of an object. In robotics, laser striping is often used to extract geometrical properties of the target object [28], but also information of the object pose can be obtained if the geometry of the object is assumed to be known.

Two different pose estimation methods are proposed in this chapter. The first method is a model-based pose estimation method using markers as image features. The method can extract the full 6-DOF pose of an object from a single image. A novel marker system is presented, where perspective projection and feature correspondence problems are solved in a computationally efficient way. The second method is a laser-triangulation-based structured light method. Methods based on image features require distinguishable features on the target object, such as markers. Active lighting methods do not require any features and also work on smooth surfaces without texture. The proposed method can estimate the pose of a planar object and extract the 5-DOF pose of two intersecting planes. A novel laser sensor calibration method is also proposed, automating the laborious calibration procedure completely.

2.1 Visual servoing

Vision-based robot control, visual servoing, is traditionally classified into two types of systems [43]: image-based and position-based systems. In image-based visual servoing the control error is expressed directly in the image plane. This has the advantage that a 3-D model of the target is not necessary. However, a disadvantage is that the Cartesian error of the robot is not easily related to the image error, which also explains some of the convergence problems related to image-based visual servoing (IBVS) [24, 66]. In IBVS, image features are used to control the robot directly. The visual sensor gives relative commands to the robot controller in order to maintain the desired tool position and orientation with respect to the target object. The calibration of the sensor is simple, as the sensor only needs to give relative information about where the end-effector should move in order to achieve the correct tool pose. The control will converge to the correct tool pose as long as the direction of the relative information given by the sensor is correct. However, there are limitations in this approach, as the desired pose must be shown to the system. Also, the control is performed in image coordinates which makes the Cartesian tool trajectory unpredictable.

In the Position-based visual servoing (PBVS), the control is based on 3-D pose of the object estimated in the Cartesian space using vision. This approach suffered earlier from high computational complexity, but currently real-time pose estimation and tracking is possible for different types of objects, see e.g. [35, 93, 51]. PBVS allows control in Cartesian coordinates, and therefore the generated robot trajectory is known. In addition to the two control approaches above, another approach, called “hybrid visual servoing”, is to divide the control between the image space and Cartesian space [64].

Different camera configurations can be used in visual servoing. The camera can be affixed to the end-effector in eye-in-hand [13] configuration, or the camera can observe both the hand and the object [83]. The effect of camera calibration errors has been studied for different configurations [37] and between position-based and image-based visual servoing [32]. Several camera-object configurations have been proposed for different applications: moving camera/static object (e.g. visual navigation), static camera/moving object (e.g. surveillance), and moving camera/moving object (e.g. moving target indication).

Real-time pose estimation of 3-D objects enables such tasks as grasping moving objects and localization of a mobile robot. In visual servoing, visual measurements are needed to provide a feedback for control. To attain controller stability, the sample rate needs to be high enough and the sensor delay low. This can be circumvented by using low feedback gains, which consequently causes slow convergence. Low gains are typical for much of the research in visual servoing. To use vision to control a fast moving platform, specialized hardware is often [47] required. To achieve efficient and robust control, the frequency of the control loop must be high enough. Complex algorithms, such as model-based 3-D tracking restrict the sampling rates typically to under 100Hz, and also considerable sensor delay exists due to image processing.

2.2 Marker-based pose estimation

To simplify the pose estimation procedure, a marker-based method is introduced. The model can be predefined, or if the 3-D locations of the markers are unknown, they can be determined by moving the robot end-effector to known locations and storing the image coordinates of the markers. From these pixel coordinates it is possible to reconstruct the 3-D coordinates of the markers, as long as the intrinsic camera parameters are known. However, predefined models offer better accuracy if they are available. In this thesis, marker-based tracking with a predefined 3-D marker model is used.

The marker system was designed so that perspective projection does not cause inaccuracies in determining the marker location. Marker systems based on determining the centroid of a circle cannot determine the marker center reliably. The perspective projection distorts the circle into an ellipse therefore giving inaccurate measurement for the marker centroid. This problem can be avoided by computationally intense ellipse fitting techniques [40]. However, in real-time tracking, simple point features are desirable.

In the approach the marker features are points and do not suffer from perspective projection. Each marker consist of three corners which can be recognized with corner extraction methods [39]. Depending on the resolution of the image and frame rate of the camera, the system can be built either to detect corners in the whole image or in the neighborhood of the corners in the previous image. A sample of a marker is shown in Fig. 2.1.

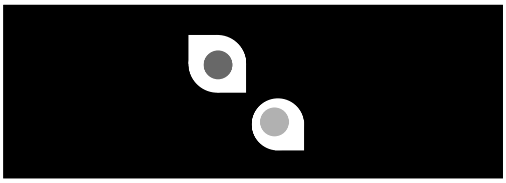


Figure 2.1: Marker with three high contrast corners.

The corners of a single marker lie on the same line, and due to the linearity of the perspective projection, the corners are collinear also in image coordinates. The corners are spaced equally, and between two corners there is a color encoded dot which is used for determining the label of the marker. Lens distortion can cause the corners to diverge from the line, but as the markers occupy only a small area in the image plane, and the dot between the corners is relatively large, the labeling is robust even when a small lens distortion exists. Each marker has unique color encoding, and therefore correspondence between 2-D and 3-D points can be made. The marker system and the coordinate axes of the estimated relative pose are shown in Fig. 2.2, and an algorithm for extracting the marker features is described in Alg. 1.

With model-based pose-estimation the pose of the object relative to the camera ${}^{C}\boldsymbol{T}_{O}$ can be determined if the intrinsic camera parameters are known. Pose estimation methods require at least three 2-D–3-D feature pairs that are not on the same line. An initial guess for the pose is calculated using DeMenthon-Davis's method [31]. However, DeMenthon-Davis's method is not as accurate as non-linear pose estimation methods [41] and therefore a conjugate gradient approach [53] is used to minimize the image plane reconstruction errors, to determine the maximum likelihood pose. If the motion between two consecutive images is small, the previous pose can be given as a starting point for the gradient descent. Otherwise the initial pose must be estimated in every frame. An algorithm for the pose estimation is given in Alg. 2.

Algorithm 1 Algorithm for extracting the feature coordinates.

```
find the corner points from the image
for all corner point $c_i$ do
    find the two closest corners $c_l$ and $c_r$
    if $c_i$, $c_l$ and $c_r$ are on the same line and $c_i$ is in the middle then
        measure the color of the dot between $c_i$ and $c_l$
        measure the color of the dot between $c_i$ and $c_r$
        determine the color encoding for the marker
        get the 3-D object coordinates of the marker corner points from a database based on the color encoding
        store the 2-D image plane coordinates and the 3-D object coordinates in a list
    else
        proceed to the next corner
    end if
end for
return a list of corresponding 2-D–3-D coordinate pairs of the feature points
```

Algorithm 2 Algorithm for 6-DOF pose estimation.

```
calculate an initial pose estimate based on the 2-D–3-D feature pairs using DeMenthon-Davis's [31] method
repeat
    back project the 3-D features on the image plane using the estimated pose and camera calibration parameters
    compute the sum of squared errors of back projected and measured image feature coordinates
    calculate the gradient of the projection error function
    perform a line search along the gradient to find a new pose estimate
until error is under a threshold
```

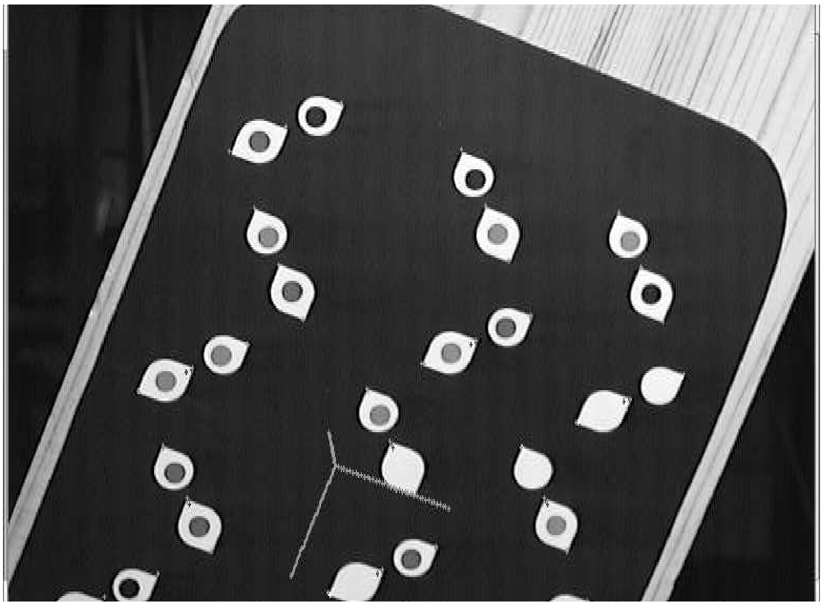


Figure 2.2: Marker system.

In the proposed approach, the camera is attached rigidly to the end-effector. To estimate the world frame pose of a target object, the transformation from the camera to the object ${}^{C}\boldsymbol{T}_O$ is first estimated using vision, as described above. The translation and rotation of the camera with respect to the end-effector ${}^{EE}\boldsymbol{T}_C$ must be determined in a calibration step in order to get the pose of the object relative to the end-effector

$$ {}^{EE}\boldsymbol{T}_O = {}^{EE}\boldsymbol{T}_C {}^{C}\boldsymbol{T}_O. \tag{2.1} $$

The absolute pose of the object with respect to the world frame is then

$$ {}^{W}\boldsymbol{T}_O = {}^{W}\boldsymbol{T}_{EE} {}^{EE}\boldsymbol{T}_O, \tag{2.2} $$

where ${}^{W}\boldsymbol{T}_{EE}$ is the end-effector pose in the world frame determined with forward kinematics.

Several hand-eye calibration techniques exist in the literature. The translation and rotation can be solved independently [97, 77] or simultaneously [42]. Calibrating the hand-eye configuration of parallel manipulators with lower mobility imposes endogenous difficulties. If the actuators have less than six degrees of freedom, it is not possible to control all the translations and rotations. The relative rotation between the camera and the end-effector can be determined with translational movement of the end-effector. Rotations of the end-effector are required to measure the translation. In lower mobility manipulators, some degrees of freedom are unavailable, which weakens the accuracy of the calibration. Also the work space of the parallel manipulator is limited, further deteriorating the calibration, as accurate hand-eye calibration requires large movements of the end-effector.

2.3 Pose estimation by structured light sensor

The visual system consists of a diffractive optics laser cross hair module and a fire-wire camera. The laser cross hair module generates two laser planes perpendicular to each other in the world coordinates. The laser pattern is projected on the target object and the camera captures an image of the pattern from a different viewpoint. Correspondence between an illuminated point on the camera image and a three-dimensional world point can be made by triangulation. An image processing algorithm extracts four lines from the image and calculates the parameters of the lines. Gaussian filtering is first applied to the image in order to smooth the speckle pattern of the laser light. Next, two illuminated areas on each column of the image are detected by using thresholding. The mean of each area is calculated and the y-coordinates of the mean of upper and lower areas are stored in separate arrays. Least squares line fit is then used to extract four lines from the arrays. Also the standard deviations of the line parameters are obtained by the least squares fit. Fig. 2.3 shows an image from the laser sensor. The line parameters are visualized by drawing two dots on each line. The second dot is chosen to be on the crossing point of two lines, and therefore three dots define a plane where the laser pattern is projected. The three dots on the left side of the image define three points on the first plane and the three dots on the right side of the image define three points on the second plane.

A narrow-field-of-view lens is attached to the camera in order to minimize the lens distortion effects. If the lens distortion is not considered, there exists a linear transformation $\boldsymbol{U} = \boldsymbol{A}\boldsymbol{L}$ from the parameters of the two lines in image plane $\boldsymbol{L} = \begin{pmatrix} a_1 & b_1 & a_2 & b_2 & 1 \end{pmatrix}^T$ to homogeneous plane parameters $\boldsymbol{U} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$. It is also possible to compensate for the lens distortion by calculating the lens distortion off-line and transforming the image plane coordinates through the lens distortion model during calibration and tracking. The extracting of the homogeneous plane parameters is presented in Alg. 3. The algorithm extracts the plane parameters of a plane on the left side of the image. The procedure is repeated to extract the plane parameters for a plane on the right side of the image by processing the arrays in reverse direction.

Algorithm 3 Algorithm for estimating the pose of a homogeneous plane.

```
filter the image with Gaussian filter
for all image plane $x$-coordinate do
    find the two largest connected components from the single image column
    compute the mean of the first connected component's pixel values and store in an array
    compute the mean of the second connected component's pixel values and store in a second array
end for
extract line parameters $a_1$ and $b_1$ from the beginning of the first array using least squares line fit
extract line parameters $a_2$ and $b_2$ from the beginning of the second array
store the line parameters into vector $\boldsymbol{L} = \begin{pmatrix} a_1 & b_1 & a_2 & b_2 & 1 \end{pmatrix}^T$
compute the homogeneous plane parameters $\boldsymbol{U} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T = \boldsymbol{A}\boldsymbol{L}$
return the homogeneous plane parameters of the first plane
```

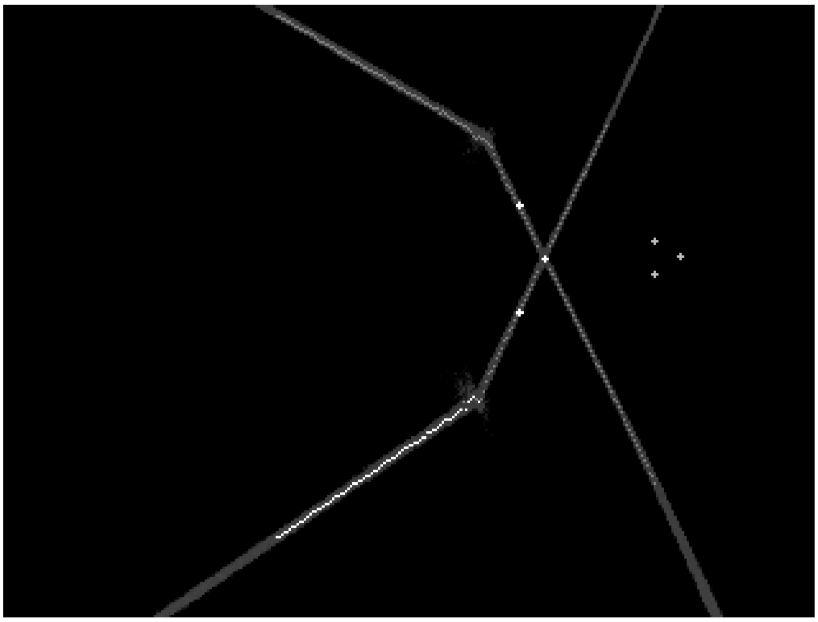


Figure 2.3: Image from the laser sensor.

One challenge with position-based visual feedback is the calibration of the visual sensor. The visual sensor must provide information of the absolute position of the object in all configurations, which requires very accurate calibration. One possibility is to use a camera fixed to the world frame, but the resolution of the camera restricts the accuracy of the measurement, and this configuration allows the target object to be observed from a single viewpoint only. When the camera is attached to the end-effector of the robot, a simple laser projection-based visual sensor can extract the geometry of the object when the sensor is moved by the robot during the machining process. When calibrating the sensor, also the pose of the sensor with respect to the end-effector must be determined. If the sensor has fixed optics and sturdy design, it is often enough to calibrate the intrinsic parameters of the sensor once. The hand-eye calibration on the other hand must be repeated if the position or orientation of the visual sensor changes with respect to the end-effector. The hand-eye calibration is often a laborious process prone to measurement errors, which must be performed by the robot end user. A method for automatic laser-sensor calibration is proposed below, which solves the intrinsic parameters of the sensor as well as the hand-eye calibration in a single calibration step.

Calibrating the laser sensor system often requires three different calibration steps [109, 105, 90]. First the intrinsic camera parameters are calibrated, then the hand-eye calibration is determined, and finally the laser planes are calibrated. Calibration errors propagate, and therefore each step adds more uncertainty to the estimate. The calibration pattern used for determining the intrinsic parameters is not exact, as it is printed with an ordinary laser printer. The robot calibration is not perfect, which increases the uncertainty in the hand-eye calibration. Lastly, the laser plane calibration suffers from the uncertainties of the previous steps, as well as imperfections in the calibration object.

Methods for determining all the required parameters in one step are also available. The calibration of the laser plane as well as the intrinsic and hand-eye parameters of the camera can be obtained in a single calibration procedure. Chen et al. [26] present a method which resolves a linear transformation $\boldsymbol{X}_w = \boldsymbol{T}_{cb}\boldsymbol{X}_i$ from the homogeneous image plane coordinates $\boldsymbol{X}_i$ to the homogeneous world coordinates $\boldsymbol{X}_w$. The linear transformation $\boldsymbol{T}_{cb}$ is a 4×3 collineation matrix describing a map from one projective space to another. The elements of the matrix are explained by Agin [1], but it is not necessary to know what the elements describe, as the linear map can be solved by linear least squares fit. Reid [84] has extended the method for plane-image point corresponds, where the original method used line-image point corresponds. It should be noted that Reid's method does not require any features on the calibration plane other than the projected laser line. The calibration is based on world plane to image point correspondences, rather than world point to image point correspondences.

For the calibration of the laser sensor, the method presented by Reid [84] has been modified. The original method requires several calibration planes to be known in the world coordinates, and not all planes all allowed to be parallel. In this section, a method requiring only one calibration plane is proposed. The robot end-effector is rotated and translated into different poses in order to view the calibration plane from different viewpoints. Fig. 2.3 shows how the robotic manipulator is transformed to different poses during the calibration procedure. The laser crosshair module projects two lines on the calibration plane, and the camera captures an image of the lines. The parameters of the two lines are obtained in image plane coordinates and stored for each pose $\boldsymbol{T}$. Homogeneous presentation for the line parameters is adopted and thus there exists a linear transformation $\boldsymbol{A}$ from the line parameters $\boldsymbol{L} = \begin{pmatrix} a_1 & b_1 & a_2 & b_2 & 1 \end{pmatrix}^T$ to homogeneous plane parameters $\boldsymbol{U} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T$. The 5×4 conversion matrix $\boldsymbol{A}$ converts the line parameters to relative plane parameters

$$\rho \boldsymbol{U} = \boldsymbol{A}\boldsymbol{L}, \tag{2.3}$$

where ρ is a scaling factor.

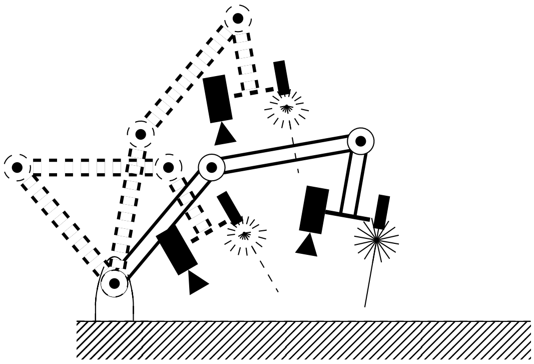


Figure 2.4: Different poses during the laser sensor calibration procedure.

The parameters of the calibration plane $\boldsymbol{U_c}$ are assumed to be known in the world coordinates, and also the pose of the end-effector can be obtained from the joint sensors of the robot. Therefore several corresponds between line parameters $\boldsymbol{L}_i$ and relative plane parameters $\boldsymbol{U}_i$ can be obtained by moving the end-effector to different poses $\boldsymbol{T}_i$, where i is a running index of the pose. The relative plane parameters can be obtained by transforming the calibration plane to the end-effector coordinates $\boldsymbol{U}_i = \boldsymbol{T}_i^{-1}\boldsymbol{U_c}$, and thus the transformation equation (2.3) then becomes

$$\rho \boldsymbol{T}_i^{-1}\boldsymbol{U_c} = \boldsymbol{A}\boldsymbol{L}_i. \tag{2.4}$$

There are 19 unknowns in the transformation matrix $\boldsymbol{A}$, as one of the parameters is a scaling factor. Therefore at least five different poses are required to determine the projectivity. The system of linear equations can then be solved by singular value decomposition to obtain the transformation matrix $\boldsymbol{A}$. An algorithm for the calibration procedure is presented in Alg. 4.

Algorithm 4 Algorithm for laser sensor calibration.

```
for i=1 to number of poses do
    capture an image
    store the line parameters as the $i$:th row of a matrix $\boldsymbol{L}_m$
    transform the calibration plane $\boldsymbol{U_c}$ to the end-effector coordinates $\boldsymbol{U}_i = \boldsymbol{T}_i^{-1}\boldsymbol{U_c}$
    store the relative calibration plane parameters $\boldsymbol{U}_i$ as the $i$:th row of a matrix $\boldsymbol{U}_m$
    move the end-effector to the next pose
end for
compute pseudoinverse $\boldsymbol{L}_m^+$ using singular value decomposition
$\boldsymbol{A} = \boldsymbol{L}_m^+\boldsymbol{U}_m$
```

To get reliable results, the poses used for the calibration should cover the work-space used during robot control, and the number of poses should be much higher than the required five. The calibration process can be fully automated and repeated during the robot control in case the calibration becomes inaccurate. During the calibration process, the tool center point was rotated around the x-axis to four different poses and to eight different poses around y-axis. For each rotation, the end-effector was also translated along the z-axis to six different heights, yielding in total $4 \times 8 \times 6 = 192$ poses. The calibration errors are presented in Fig. 2.5. When the end-effector was rotated to different poses, it was also translated in order to keep the tool centerpoint stationary. The plot describes the error of the plane distance parameter d for each individual end-effector position in the x- and y-coordinates. It can be seen that the calibration is more accurate in the center of the calibration area where the relative rotation is smaller, and that the variance of the error becomes higher when the relative rotations are large. The histogram of the calibration errors shown in Fig. 2.6 gives an estimate of the expected errors during robot control. The error distribution is close to Gaussian, and the magnitude of the errors is in the order of few millimeters.

2.4 Summary and discussion

Two different pose estimation methods were proposed in this chapter, a marker-based pose estimation method and a structured light laser sensor method. A novel laser sen-

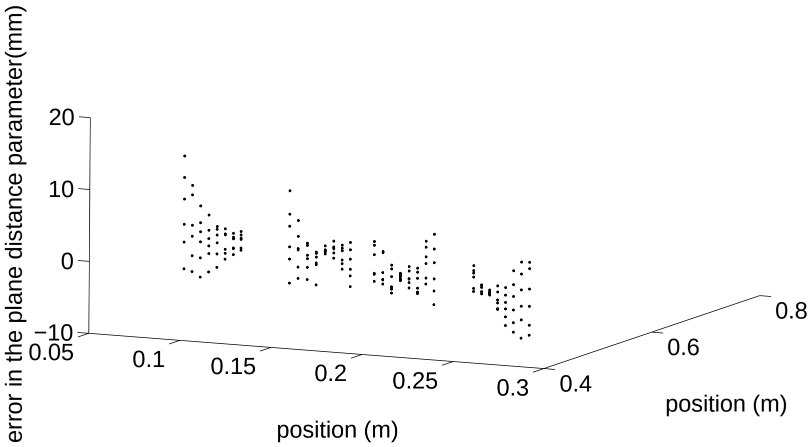


Figure 2.5: Calibration errors for each pose.

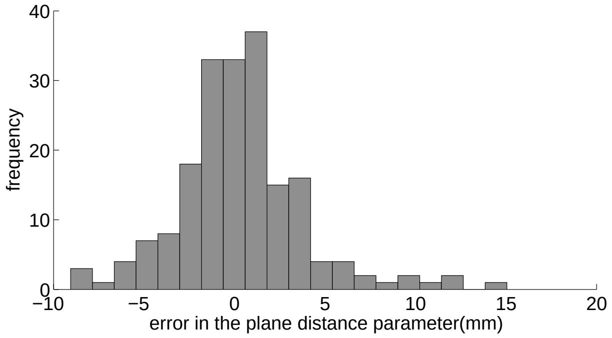


Figure 2.6: Histogram of calibration errors.

sor calibration method was also proposed, which automates the calibration procedure. Experimental results and error analysis of the laser sensor calibration procedure were shown.

Model-based pose estimation methods require a model of the target object. This can be for example a CAD-model of the object or a list of marker features in 3-D coordinates. Marker-based pose estimation methods are often used in industrial applications as they simplify the pose estimation procedure. In this chapter, a robust marker system was proposed, which utilizes color encoding to determine the correspondence of the markers. The marker system is based on point features instead of circular markers and therefore does not suffer from perspective projection.

Altough model-based pose estimation can be used in many applications, distinguishable features on the target object, such as markers are required. Active lighting methods do not require any features and also work on smooth surfaces without texture. Structured light methods are often used to determine the shape of an object [28], but it is also possible to extract the pose of an object if assumptions of the object shape are made.

The structured light laser sensor method proposed in this chapter can estimate the pose of a planar object and extract a 5-DOF pose of two intersecting planes. Calibrating a structured light eye-in-hand visual system is a challenging and laborious process. The calibration methods presented in literature often require three calibration steps [109, 105, 90]. By utilizing the calibration method proposed in this chapter, the procedure is fully automated and all the required parameters are obtained in a single calibration step.

CHAPTER III

Fusion of proprioceptive and exteroceptive sensors

This chapter addresses three problems: uncertain visual measurements, different sampling rates, and compensation of the sensor delay. The extended Kalman filter (EKF) framework is used for incorporating measurements with different uncertainties and sampling rates. The EKF allows reduced uncertainties of vision-based estimates and interpolation between low frequency visual measurements, making it possible to track a moving target. The expected future target pose can be predicted with the EKF to alleviate inherent latencies in the visual system and control loop. When proprioceptive and visual measurements are synchronized in the EKF, the estimate of the target pose becomes delayed. To compensate for this delay, as well as latencies in the control loop, the future target pose is predicted with the EKF.

The reduction of measurement errors and fusion of several sensory modalities with a Kalman filter (KF) framework is widely used in robotics, for example in 3-DOF robot navigation [9, 27, 56, 85] and 6-DOF pose tracking [102, 46]. These methods assume, either implicitly or explicitly, a stationary target/moving camera configuration. In [57], the localization of mobile robot with a moving target/stationary camera configuration is proposed. The trajectory of the moving target is known and the robot stops while observing the target. Kalman filters have also been applied in fusing several different visual cues for pose tracking for both moving target/stationary camera and stationary target/moving camera configurations [49, 51]. Tracking a moving target with a moving camera has not been studied in detail. However, such tracking would allow many useful applications, including the manipulation of moving objects, which is especially useful when the work space of the manipulating robot is limited. In that case, it would be possible to process large workpieces by attaching the workpieces onto a moving platform. The size of the work space is a serious limitation especially for parallel robots, which provide superior force control properties due to high stiffness.

In the visual servoing context, Kalman filters are typically used only for filtering uncertain visual measurements, and the proprioceptive information of the end-effector is not taken into account. Wilson et al. [103] propose to solve the pose estimation problem for position-based visual servoing with the KF framework, as this will balance the effect of measurement uncertainties. A similar approach has been used by Lippiello [59], utilizing several cameras and fusing the information together. However, in these approaches the KF can be understood as a single iteration of an iterative Gauss-Newton procedure for pose estimation, and as such are not likely to give optimal results for the non-linear pose estimation problem. Nomura et al. [75] propose to use KFs to reduce measurement noise in image-based visual servoing. Their approach differs from the one proposed in this thesis by modelling the uncertainty in the image space, while in this thesis the uncertainty is considered in Cartesian space. Another difference is that their state vector is the pose of the object relative to the camera, and in this thesis the state vector is the absolute pose of the object in the world frame. The world-frame-based prediction model is more linear, as the motion of the end-effector does not affect the state. It should be noted that in an eye-in-hand system, the absolute pose of a target cannot be obtained without fusing the proprioceptive information of the end-effector and the relative visual measurement.

While the above mentioned works address the problem of uncertain measurements, the sensor delay and different sampling rates of the sensors should also be considered. Data fusion of sensors with different sample rates have been proposed earlier by Langois et al. [55]. They describe a system composed of one-degree-of-freedom manipulator controlled by fusion of visual measurement and optical encoder in the KF-framework. Sim et al. [89] propose a multirate predictor control utilizing an $\alpha - \beta$ filter to predict the desired path and velocity of the end-effector. In their work, sensor delay is taken explicitly into account, but only visual measurement is considered in the filter. Their approach is also only applicable for estimating the Cartesian position, not the full pose.

Vision offers a low cost sensor modality with a high information content. Although relatively fast cameras and visual algorithms exist, the sampling rates of visual measurements are still lower than the frequency of positional encoders and joint angle sensors. By combining the visual measurements with high frequency proprioceptive information, the control loop can be run with higher frequency to allow better stability and faster convergence. In addition, the sensor delay of the visual measurements must be taken into account when fusing the measurements. Especially the eye-in-hand configuration requires precise synchronization of proprioceptive information and visual measurement. Otherwise the vision will give erroneous information when the end-effector is in motion. In addition to low sample rate and sensor delay, visual measurements are also always uncertain. The resolution of a camera is limited, image noise is present and motion blur adds error to the image. When using a single measurement at each time instant, as is typical in visual servoing, uncertainty in the visual measurements can cause undesired oscillations and weaken the accuracy. By fusing multiple measurements together, more accurate estimates of the target motion can be made compared to a single image.

The number of different tracking [35, 93, 51] algorithms is almost as great as the number of applications. Therefore, in this chapter a particular 3-D tracking algorithm is not proposed, but the theory is given in a way which allows it to be applied to any tracking algorithm.

3.1 Extended Kalman filter

In this section, the notation in the context of the Extended Kalman filter (EKF) is presented. The measurements from position encoders and visual system are integrated in an EKF. The EKF estimates the state of a system with a system model $f(\boldsymbol{x})$ describing the dynamics of the system

$$\boldsymbol{x}(t) = f(\boldsymbol{x}(t-1)) + g(\boldsymbol{u}(t-1)) + \boldsymbol{w}(t-1), \tag{3.1}$$

where $g(\boldsymbol{u})$ describes the effect of the system input $\boldsymbol{u}$ and $\boldsymbol{w}$ is the model error modeled as a zero-mean Gaussian random variable. In addition, a measurement model $h(\boldsymbol{x})$ is used to link the internal state to measurable quantities $\boldsymbol{z}$ by

$$\boldsymbol{z}(t) = h(\boldsymbol{x}(t)) + \boldsymbol{v}(t), \tag{3.2}$$

where $\boldsymbol{v}$ is the Gaussian measurement error. Uncertainty is modeled with covariance matrices so that $\boldsymbol{P} = Cov(\boldsymbol{x})$, $\boldsymbol{Q} = Cov(\boldsymbol{w})$, and $\boldsymbol{S} = Cov(\boldsymbol{v})$.

In each time-step, a measurement update is calculated as follows

$$\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + \boldsymbol{K}\left(\boldsymbol{z}(t) - \boldsymbol{H}\boldsymbol{x}(t)\right), \tag{3.3}$$

where $\boldsymbol{H}$ is the observation model matrix and the Kalman gain $\boldsymbol{K}$ is defined as

$$\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^T + \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T)^{-1}, \tag{3.4}$$

where $\boldsymbol{V}$ is observation noise with covariance $\boldsymbol{R}$.

An algorithm for extended Kalman filtering is presented in Alg. 5. The basic EKF estimation equations are only briefly mentioned in this section. For a more thorough explanation of the EKF, the reader is referred to [58].

Algorithm 5 Algorithm for extended Kalman filtering.

```
initialize the state $\boldsymbol{x}$ and the state covariance $\boldsymbol{P}$
repeat
    predict the state $\boldsymbol{x}(t) = f(\boldsymbol{x}(t-1)) + g(\boldsymbol{u}(t-1)) + \boldsymbol{w}(t-1)$
    predict the state covariance $\boldsymbol{P} = Cov(\boldsymbol{x})$
    obtain a new measurement $\boldsymbol{z}(t)$
    calculate the measurement residual $\boldsymbol{z}(t) - \boldsymbol{H}\boldsymbol{x}(t)$
    calculate the residual covariance $\boldsymbol{S}$
    calculate the Kalman gain $\boldsymbol{K} = \boldsymbol{P}\boldsymbol{H}^T(\boldsymbol{H}\boldsymbol{P}\boldsymbol{H}^T + \boldsymbol{V}\boldsymbol{R}\boldsymbol{V}^T)^{-1}$
    update the state estimate $\boldsymbol{x}(t+1) = \boldsymbol{x}(t) + \boldsymbol{K}\left(\boldsymbol{z}(t) - \boldsymbol{H}\boldsymbol{x}(t)\right)$
    update the state covariance $\boldsymbol{P}$
until filtering is stopped
```

3.2 6-DOF tracking

In 3-D tracking, the relative 3-D pose of the object with respect to the end-effector is measured by using vision. The tracked object is assumed to be moving with a constant velocity, and the end-effector motion is assumed to be known as the joint sensors of a robotic manipulator measure the end-effector pose at every time instant. The approach was first proposed in [6], and later extended by modelling the covariance matrix of the visual measurement in detail and combining the tactile measurement into the EKF model [3]. If there is uncertainty in the end-effector motion, the method described in [5] can be adopted.

The 12-dimensional state vector $\boldsymbol{x}$ contains the 6-DOF velocity and pose information of the object,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x_p}^T & \dot{\boldsymbol{x}_p}^T \end{pmatrix}^T , \tag{3.5}$$

where $\boldsymbol{x_p} = \begin{pmatrix} x & y & z & \phi & \theta & \psi \end{pmatrix}^T$. The poses are given with respect to a world coordinate system, and the rotation is represented by immediate angles. Three-parameter axis angle representation is used for the angles. The ϕ, θ, and ψ represent the rotation axis and the length of the vector $\begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^T$ gives the angle.

Because of the problem of non-unique angles, the approach from [102] is adopted, and the orientation of the objects is stored externally, outside the EKF state, so that ϕ, θ, and ψ only represent incremental changes to the orientation. After each update step of the EKF, the rotation angle of the target is integrated into rotation matrix $\boldsymbol{R}_t$, and the rotation angle is reset to zero. The homogeneous transformation describing the pose of the target in the world coordinates can now be written as

$$^{W}\boldsymbol{T}_O = \begin{pmatrix} & x \\ \boldsymbol{R}_t\boldsymbol{R}(\phi,\theta,\psi) & y \\ & z \\ \boldsymbol{0} & 1 \end{pmatrix} , \tag{3.6}$$

where $\boldsymbol{R}(\cdot)$ is a function mapping immediate angles to a rotation matrix. Now, the 3-D system can be modeled with

$$\boldsymbol{x}(t+1) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{w}(t), \tag{3.7}$$

where

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{I}_6 & \Delta t \boldsymbol{I}_6 \\ \boldsymbol{0} & \boldsymbol{I}_6 \end{pmatrix} . \tag{3.8}$$

With the assumption that an unknown constant acceleration affects the object at every time instant, the model covariance matrix $\boldsymbol{Q}$ is now

$$\boldsymbol{Q} = \begin{pmatrix} \frac{1}{4}\Delta t^4 \boldsymbol{\Sigma}_d & \frac{1}{2}\Delta t^3 \boldsymbol{\Sigma}_d \\ \frac{1}{2}\Delta t^3 \boldsymbol{\Sigma}_d & \Delta t^2 \boldsymbol{\Sigma}_d \end{pmatrix} , \tag{3.9}$$

where

$$\boldsymbol{\Sigma}_d = \begin{pmatrix} \sigma_{dx}^2 \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \sigma_{d\phi}^2 \boldsymbol{I}_3 \end{pmatrix} , \tag{3.10}$$

where σ_{dx}^2 describes the process uncertainty for the translation and $\sigma_{d\phi}^2$ the uncertainty for the rotations. The uncertainties are assumed to be unknown accelerations with Gaussian noise and zero mean. It is important to notice that by using $^{W}\boldsymbol{T}_O$ as a state variable instead of $^{C}\boldsymbol{T}_O$, the model can estimate the motion of the object more accurately. If the relative pose from the camera to the object is used as a state vector, the motion of the end-effector alters the state, and even if the object is stationary, the constant velocity assumption for the state might not hold.

The visual system provides a measurement describing the relative pose of the target object with respect to the camera, ${}^{C}\hat{\boldsymbol{T}}_{O}$. The Kalman filtering requires the difference between predicted and measured quantities to be calculated. The structure of $SO(3)$ (the space of orientations) is such that the difference between rotation angles is geometrically meaningless, which in this case can cause severe problems. Therefore, the rotation and translation parts of the measured relative pose are considered separately below.

The mapping between the current state and the measured relative translation from the camera to the target ${}^{C}\hat{\boldsymbol{t}}_{O}$ can be written as

$$ {}^{C}\hat{\boldsymbol{t}}_{O} = {}^{C}\boldsymbol{R}_{W} \left({}^{W}\boldsymbol{t}_{O} - {}^{W}\boldsymbol{t}_{C} \right). \tag{3.11} $$

Next, noting that ${}^{C}\boldsymbol{R}_{W} = {}^{W}\boldsymbol{R}_{C}^{T}$, ${}^{W}\boldsymbol{t}_{O} = \begin{pmatrix} x & y & z \end{pmatrix}^{T}$ and ${}^{W}\boldsymbol{t}_{C}$ is known, the measurement model for translation can be written as

$$ \boldsymbol{y}_{T}(t) = \boldsymbol{H}_{T}\boldsymbol{x}(t) - {}^{C}\boldsymbol{R}_{W}{}^{W}\boldsymbol{t}_{C}, \tag{3.12} $$

where

$$ \boldsymbol{H}_{T} = \begin{pmatrix} {}^{C}\boldsymbol{R}_{W} & \boldsymbol{0}_{3} & \boldsymbol{0}_{3} & \boldsymbol{0}_{3} \end{pmatrix}. \tag{3.13} $$

For the rotation measurement, it is beneficial to use a three-parameter representation, because then the additional constraints for over-parameterized representations are not necessary. However, contrary to translation, the arithmetic difference between the measured and predicted angles does not have a geometric explanation, and therefore the difference must be calculated with rotation matrices.

As the rotational parameters in the state vector only describe the incremental changes to the orientation, the measurement model for the rotation must also use incremental rotations. The measurement can be written as the difference between the stored target rotation $\boldsymbol{R}_{t}$ and the current visual measurement of the target rotation ${}^{W}\boldsymbol{R}_{O_m}$. Then the difference of these two rotations is ${}^{O_t}\boldsymbol{R}_{O_m}$, where O_t is the stored target rotation frame and O_m the measured target rotation frame. The prediction model gives a second estimate of the incremental change

$$ {}^{O_t}\boldsymbol{R}_{O_p} = \boldsymbol{R}(\phi, \theta, \psi), \tag{3.14} $$

where O_p is the predicted target rotation frame. The residual between the prediction and measurement then becomes ${}^{O_p}\boldsymbol{R}_{O_m}$.

By introducing a function $\boldsymbol{\phi}(\cdot)$ converting a rotation matrix to immediate angles, the predicted incremental angles are obtained by

$$ \boldsymbol{\phi}({}^{O_t}\boldsymbol{R}_{O_p}). \tag{3.15} $$

Now, in order to use the incremental rotation measurement in the EKF, only the gradient $\boldsymbol{H}_{R}$ of the $\boldsymbol{\phi}({}^{O_t}\boldsymbol{R}_{O_p})$ is needed with respect to the system orientation parameters, $\boldsymbol{H}_{R} = \frac{\partial \boldsymbol{\phi}({}^{O_t}\boldsymbol{R}_{O_p})}{\partial \phi, \theta, \psi}$; The gradient can be calculated analytically by taking the partial derivatives of the rotation matrix with respect to each element and then utilizing the chain rule to get the partial derivatives with respect to the orientation parameters. The measurement model for the rotation is then

$$ \boldsymbol{H}_{R} = \begin{pmatrix} \boldsymbol{0}_{3} & -\frac{\partial \boldsymbol{\phi}({}^{O_t}\boldsymbol{R}_{O_p})}{\partial \phi, \theta, \psi} & \boldsymbol{0}_{3} & \boldsymbol{0}_{3} \end{pmatrix}. \tag{3.16} $$

The covariance of the visual measurement is often modeled with a constant diagonal matrix as

$$\boldsymbol{S_c} = \begin{pmatrix} \sigma_{vx}^2 \boldsymbol{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \sigma_{v\phi}^2 \boldsymbol{I}_3 \end{pmatrix}, \tag{3.17}$$

where σ_{vx}^2 is the uncertainty for the position of the visual measurement and $\sigma_{v\phi}^2$ the uncertainty for the rotation. However, by using a constant diagonal covariance matrix, the information of the relative uncertainties between each degree of freedom are lost and the covariance is restricted to be fixed over time. Also the cross covariance terms of the visual measurement are lost. In the next subsection the covariance matrix is modelled by using a Hessian approximation.

3.2.1 Uncertainty modelling of visual measurement

In order to fuse the measurements in the EKF, the uncertainty of each individual measurement must be estimated. The uncertainty of the visual measurement depends on the resolution and optical properties of the camera. Image noise due to poor lighting or short exposure time adds uncertainty to the measured feature point locations in pixel coordinates. By assuming this uncertainty to be zero mean Gaussian and projecting it into the uncertainty of the object pose in Cartesian space, the measurement can be incorporated in the EKF. A linear approximation is used where the Hessian of the visual measurement is calculated and the probability density of the transformed measurement is considered Gaussian. As the transformation is non-linear in reality, this will only give an approximation of the true distribution. However, the approximation is reasonably good as long as the measurement is not close to a singularity.

Calculating the true Hessian would require solving the second derivatives of the pose estimation problem. This can be done analytically, but in a real-time application the computational burden is considered too large. Instead, the Hessian is approximated by the Jacobian estimate $2\boldsymbol{J}\boldsymbol{J}^T$, which ignores second derivative terms [17]. The obtained Hessian is also validated by numerical second order differentiation of the projection equations in order to verify the analytical results. The Hessian estimate and numerical approximation agree when the image Jacobian is not close to a singularity.

The Hessian estimates how much the pose changes when there is a small error in the image plane coordinates. The Jacobian $\boldsymbol{J}$ is formed by derivating the difference between the projected and measured feature coordinates in the image plane with respect to each pose parameter. The projection equations solve the transformation from the image plane coordinates (u, v) to the pose coordinates $(x, y, z, \phi, \theta, \psi)$

$$\hat{X}_i = \boldsymbol{R}_{00} X_i + \boldsymbol{R}_{01} Y_i + \boldsymbol{R}_{02} Z_i + x \tag{3.18}$$

$$\hat{Y}_i = \boldsymbol{R}_{10} X_i + \boldsymbol{R}_{11} Y_i + \boldsymbol{R}_{12} Z_i + y \tag{3.19}$$

$$\hat{Z}_i = \boldsymbol{R}_{20} X_i + \boldsymbol{R}_{21} Y_i + \boldsymbol{R}_{22} Z_i + z, \tag{3.20}$$

where $\boldsymbol{R} = \boldsymbol{R}(\phi, \theta, \psi)$, (X_i, Y_i, Z_i) are the coordinates of the feature i in the 3-D model, and $(\hat{X}_i, \hat{Y}_i, \hat{Z}_i)$ can be solved from

$$\triangle u_i = u_i - \frac{\hat{X}_i}{\hat{Z}_i} \tag{3.21}$$

$$\triangle v_i = v_i - \frac{\hat{Y}_i}{\hat{Z}_i}, \tag{3.22}$$

where (u_i, v_i) are the coordinates of the feature i in the image plane and $(\triangle u_i, \triangle v_i)$ the difference between projected and measured feature coordinates in the image plane. The Jacobian $\boldsymbol{J}$ is now

$$\boldsymbol{J} = \begin{pmatrix} \frac{\partial \triangle u_1}{\partial x} & \frac{\partial \triangle v_1}{\partial x} & \cdots & \cdots & \frac{\partial \triangle u_n}{\partial x} & \frac{\partial \triangle v_n}{\partial x} \\ \frac{\partial \triangle u_1}{\partial y} & \frac{\partial \triangle v_1}{\partial y} & \cdots & \cdots & \frac{\partial \triangle u_n}{\partial y} & \frac{\partial \triangle v_n}{\partial y} \\ \frac{\partial \triangle u_1}{\partial z} & \frac{\partial \triangle v_1}{\partial z} & \cdots & \cdots & \frac{\partial \triangle u_n}{\partial z} & \frac{\partial \triangle v_n}{\partial z} \\ \frac{\partial \triangle u_1}{\partial \phi} & \frac{\partial \triangle v_1}{\partial \phi} & \cdots & \cdots & \frac{\partial \triangle u_n}{\partial \phi} & \frac{\partial \triangle v_n}{\partial \phi} \\ \frac{\partial \triangle u_1}{\partial \theta} & \frac{\partial \triangle v_1}{\partial \theta} & \cdots & \cdots & \frac{\partial \triangle u_n}{\partial \theta} & \frac{\partial \triangle v_n}{\partial \theta} \\ \frac{\partial \triangle u_1}{\partial \psi} & \frac{\partial \triangle v_1}{\partial \psi} & \cdots & \cdots & \frac{\partial \triangle u_n}{\partial \psi} & \frac{\partial \triangle v_n}{\partial \psi} \end{pmatrix}, \tag{3.23}$$

and the Hessian $\boldsymbol{H_e} \approx 2\boldsymbol{J}\boldsymbol{C}_I^{-1}\boldsymbol{J}^T$, where $\boldsymbol{C}_I$ is a diagonal matrix describing the uncertainties for each feature in the image plane. The covariance matrix for the relative visual measurement from the camera to the object in Cartesian space can now be written as $\boldsymbol{C}_C = \boldsymbol{H_e}^{-1}$.

In the EKF, however, this is only applicable to the translational measurement. For the rotational measurement, the Hessian must be projected to the incremental rotation measurement function. For the rotational measurement model, the gradient of the measurement function $\boldsymbol{H}_R$ is already calculated. The covariance for the visual measurement is then

$$\boldsymbol{S}_H = \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_R \end{pmatrix} \boldsymbol{C}_C \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{H}_R \end{pmatrix}^T. \tag{3.24}$$

Modeling the visual uncertainty is described in Alg. 6.

Algorithm 6 Algorithm for determining visual uncertainty.

```
calculate the difference between the projected and measured feature coordinates in image plane analytically
derivate the difference with respect to each pose parameter to obtain Jacobian $\boldsymbol{J}$
approximate the Hessian by $\boldsymbol{H_e} \approx 2\boldsymbol{J}\boldsymbol{C}_I^{-1}\boldsymbol{J}^T$
project the Hessian to the space of the EKF measurement
```

3.2.2 Handling the latency and different sampling rates of the sensors

The sampling rate for a position encoder is typically much higher than for visual measurement. In addition, the visual measurement has latency due to the exposure, transfer, and processing of the image. These typically limit the response rate of visual control, but this can be alleviated by modeling the effects. Due to different sampling rates, the prediction is performed more often than the measurement update. This can be done by setting the visual measurement terms in the Kalman gain matrix or the gradient matrix to zero when there is no visual measurement.

To compensate for the sensor delay of vision, the system shown in Fig. 3.1 is used. The proprioceptive measurements are delayed by the amount of visual sensor delay so that the two are synchronized when fed to the EKF. The delay of the filter can be compensated by a modified prediction model (bottom of Fig. 3.1) to estimate the current, non-delayed state. In the modified prediction, the time interval Δt in the state transition matrix $\boldsymbol{F}$ is substituted by sensor delay τ. Compensation of the sensor delay is explained in Alg. 7.

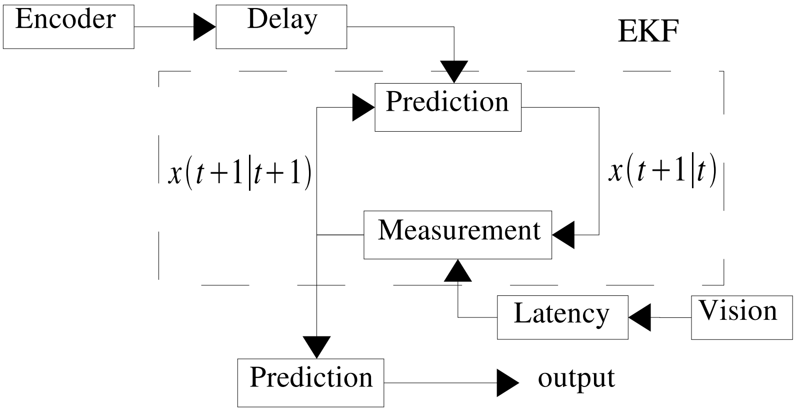


Figure 3.1: Compensation of the sensor delay.

Algorithm 7 Algorithm for compensation of the sensor delay.

```
repeat
    estimate the pose of the target using the visual measurement
    delay the encoder measurements for the amount of sensor delay in visual measurement
    run the EKF loop
    predict the current non-delayed state
    repeat
        delay the encoder measurements for the amount of sensor delay in visual measurement
        set the visual measurement terms in Kalman gain matrix to zero
        run the EKF loop
        predict the current non-delayed state
    until new visual measurement is obtained
until tracking is stopped
```

3.2.3 Experiments

Experiments were conducted on a 5-DOF parallel hydraulic manipulator [61] shown in Fig. 3.2. A linear track was mounted on the robot frame to increase the work space of the robot. The object to be tracked was affixed to the linear track. The absolute position of the object with respect to the robot frame was estimated by using visual measurement alone and by the EKF. The experiments were conducted to compare the performance of the EKF-based control and position-based visual servoing.

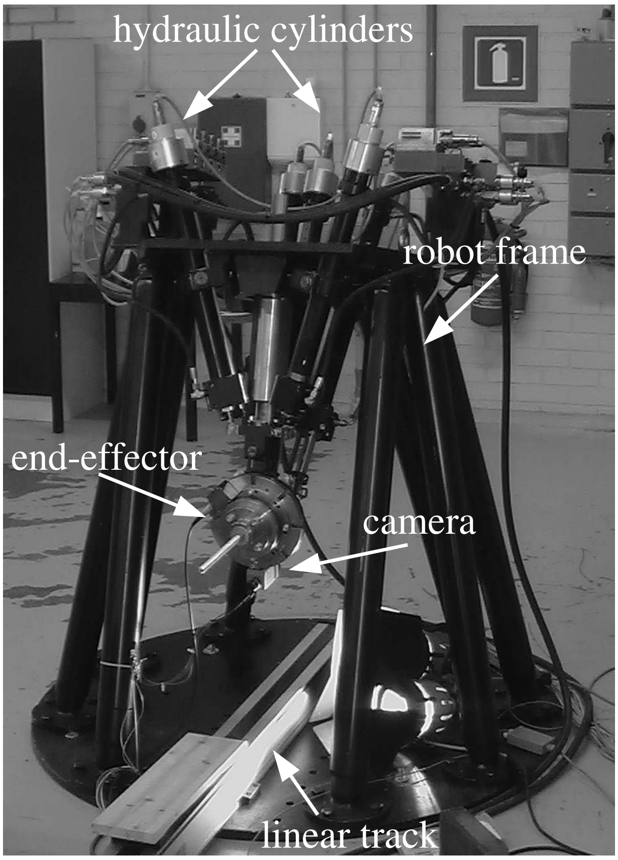


Figure 3.2: 5-DOF parallel hydraulic robot.

The robot coordinate system is based on Euler angles, whereas the EKF uses an axis angle representation. Therefore, the axis angles had to be converted to Euler angles in order to control the robot. The object rotations were selected so that the Euler angle representation did not suffer from singularities in the robot work space. In parallel manipulators, the direct kinematics of the system is difficult to solve. Mapping from joint coordinates to the pose of the end-effector is usually only obtainable through iterative methods. The inverse kinematics, on the other hand, is available. When following a known trajectory, the pose of the end-effector follows the desired trajectory approximately with some delay. As the lengths of the cylinders are controlled with PI-controllers, a true pose is only available when the PI-controller has converged. The error between the true pose and the one approximated from the trajectory is negligible, however, as long as the sensor delay is taken into account.

To validate the theory presented earlier experimentally, as well as the analysis of measurement errors in visual servoing [52], a test setup was constructed where the uncertainties for different degrees of freedom of the visual measurement could be measured.

The calibration of the vision is essential in order to get reliable results. Intrinsic camera parameters were obtained by using the Matlab camera calibration toolbox [20]. The toolbox implements Zhang's method [107], using a simple planar calibration pattern. Several hand-eye calibration techniques exist in the literature, but only an initial estimate of the hand-eye calibration was obtained using techniques from the literature, and the hand-eye calibration was further optimized by minimizing the apparent motion in the object pose estimate.

In the first experiment the robot was controlled under position-based visual servoing. The absolute pose of the object respect to the world frame ${}^{W}\boldsymbol{T}_{O}$ was determined from:

$$ {}^{W}\boldsymbol{T}_{O} = {}^{W}\boldsymbol{T}_{EE}{}^{EE}\boldsymbol{T}_{C}{}^{C}\boldsymbol{T}_{O}. \tag{3.25} $$

First the uncertainties of the visual system were determined by keeping the end-effector stationary and measuring the 3-D pose of the object. Fig. 3.3(a) shows the uncertainties of the visual measurement while the robot was stationary and the hydraulics was off. The x- coordinate corresponds to the camera optical axis, whereas measurements along y- and z- coordinates are parallel to the image plane. The uncertainty in the x- coordinate is substantially larger than for the y- and z- coordinates. Different uncertainty is observable in rotations about the camera optical axis and off-axes also in angular measurements.

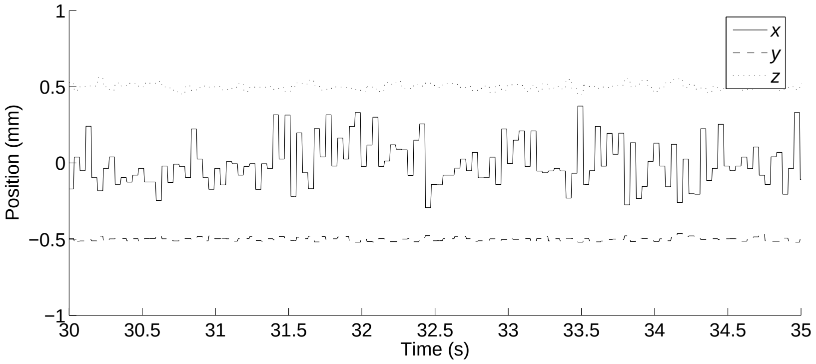
The robot was set to initial position and the visual control loop was activated. Fig. 3.3(b) shows that the system oscillated heavily under visual control. The uncertainties of the visual measurements for each axis are shown in Table 3.1. σ_o is the uncertainty of the off-line visual measurement and σ_c the uncertainty when the robot was controlled by position-based visual servoing. It is important to note that the oscillations seen in Fig. 3.3(b) are caused not only by measurement noise but also a low sample rate, and the following results show that the EKF-based estimation can alleviate this problem.

Table 3.1: Uncertainties of the vision.

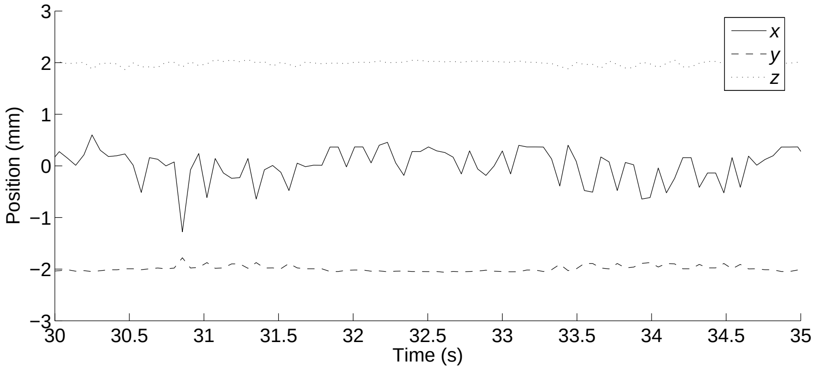
	σ_o	σ_c		σ_o	σ_c
x	$0.1493mm$	$0.4074mm$	α	0.0404°	0.0859°
y	$0.0121mm$	$0.2105mm$	β	0.1178°	0.3174°
z	$0.0211mm$	$0.1496mm$	γ	1.1032°	1.3326°

As the system oscillated heavily under position-based visual servoing, a low-pass filter was set up to smooth the visual measurement. An experiment was made where an object on the linear track was moved slowly towards the end-effector. A proportional controller was used to track the object with constant distance using the low-pass filtered visual measurement, thus moving the end-effector backwards. After some time, the motion of the linear track was reversed. The results shown in Fig. 3.4 illustrate the motion of the linear track and the end-effector controlled with low-pass filtered visual measurement. The filter parameters were tuned to give a smooth response, and thus substantial phase shift between the linear track and end-effector position can be seen.

In the following experiments, the performance of the EKF was evaluated. The linear track was again allowed to move towards the end-effector. This time the distance between the end-effector and the object was set initially larger than the set distance. The robot had to move towards the object first, before the desired tracking distance was achieved. Then a



(a)



(b)

Figure 3.3: Effect of measurement noise.

PI-position-based controller was used to track the object with a given relative pose using the EKF estimate. Similar to the previous case, the linear track motion was reversed after some time.

The experiment was conducted to compare different filtering techniques to smoothen the uncertain visual measurement. The object on the linear track was moved slowly towards the end-effector, and the position of the moving object was estimated with four different methods: visual measurement without filtering, the proposed EKF estimate, the EKF without latency compensation, and a low pass filtered visual measurement. The parameters of the low pass filter were tuned to give approximately the same amount of smoothing as the EKF. After some time, the motion of the linear track was first stopped for a while and then reversed, and the object was moved back to the original position. The motion of the linear track and the EKF estimates of the position and the velocity are shown in Fig. 3.5. The relative distance between the object and the end-effector was

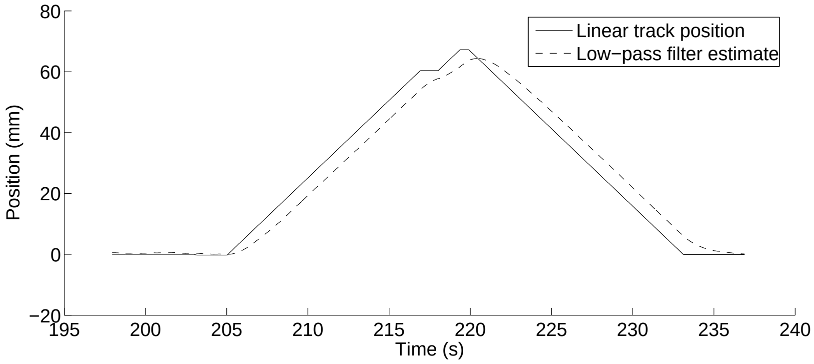


Figure 3.4: Low-pass filter in control loop.

not constant in this test-run. As the position of the object in the world coordinates was unknown, the incremental encoders of the linear track could only give relative ground truth data. The position of the linear track, as well as the visual measurement were reset in the middle of the linear track motion range in order to minimize calibration errors. Inexact hand-eye and intrinsic camera calibration generates some systematic error in distance measure when the position of the object moves relative to the camera. This is clearly visible in the converged EKF positions when the linear track is stationary. However, these inaccuracies hinder the performance of all the methods with the same amount. The effect is much less pronounced when the end-effector is allowed to track the object, and the relative distance is kept the same.

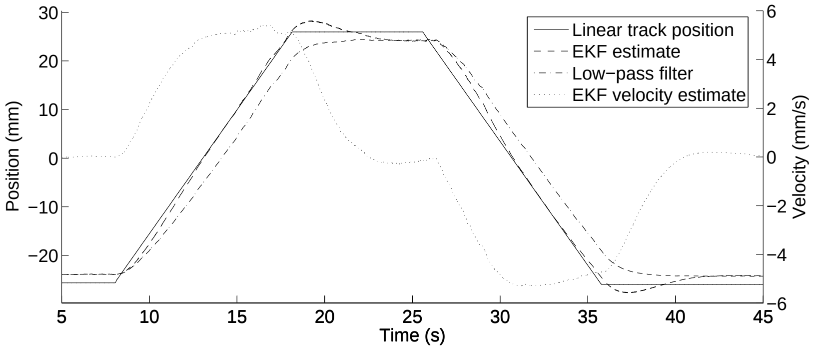


Figure 3.5: Comparison of different filtering techniques while tracking.

The comparison of the methods in Fig. 3.6 verifies that the phase shift for the EKF is significantly smaller than for a low-pass filter. When the velocity estimate of the EKF has converged, the EKF outperforms even the theoretical linear fit of the visual measurement, which was calculated off-line. The linear fit is comparable to the EKF without sensor delay compensation where the external prediction model is omitted.

Finally, a 4-DOF approach test was conducted with the EKF to study its performance

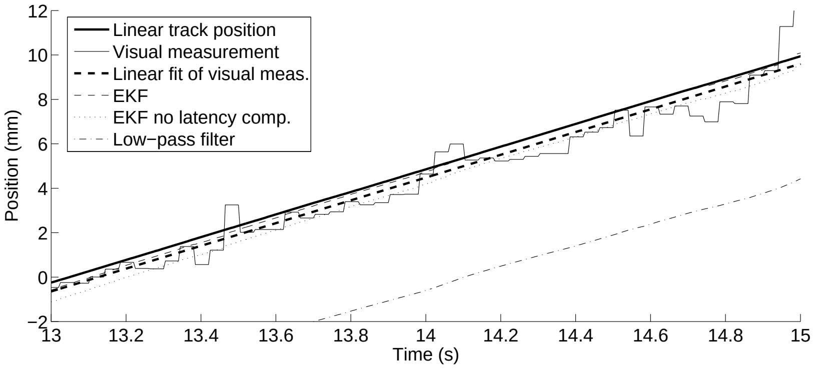


Figure 3.6: Comparison of different filtering techniques and pure visual measurement.

in a more general case. The desired pose between the end-effector and the object was set by the teaching-by-showing method. After teaching, the pose of the object was altered and the linear track was initiated with constant velocity. The new pose of the object and its velocity was estimated with the EKF. The end-effector was then controlled to the desired relative pose. In Fig. 3.7, Cartesian robot positions and the controller set point obtained using the EKF estimate are shown. Fig. 3.8 presents the EKF velocity estimates for the translations and Fig. 3.9 the robot rotation, rotation estimate and angular velocity estimate. It can be seen that the control in 4-DOF converges without oscillations even for the moving target. During the approach phase, the estimate of the object pose and velocity are less accurate, as the relative pose between the end-effector and the object changes. When the desired relative pose is achieved, the EKF estimate converges with minor noise also for velocities. It should be noted that the velocity as a differential quantity is very sensitive for measurement errors.

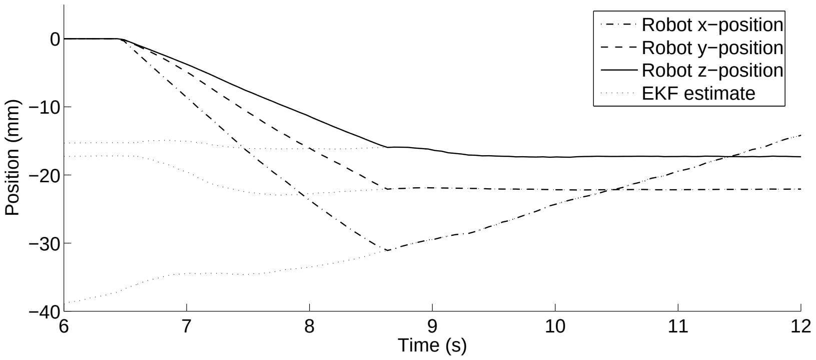


Figure 3.7: EKF approaching the object.

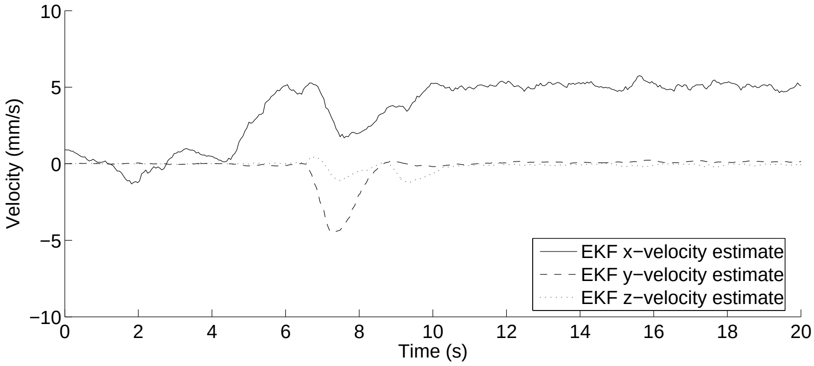


Figure 3.8: EKF velocity estimates.

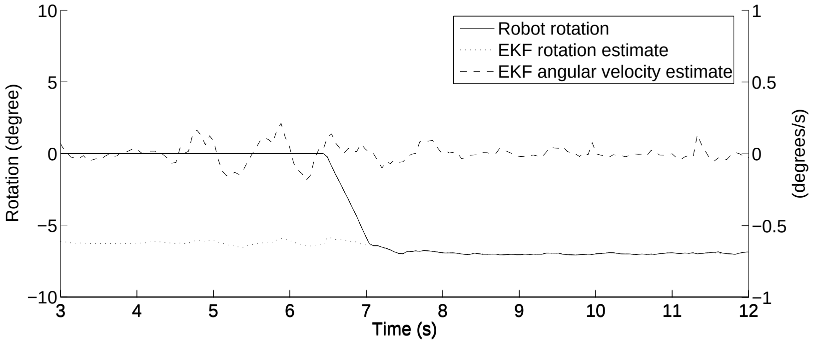


Figure 3.9: Rotation and angular velocity.

3.3 Modelling the uncertainty of the end-effector pose

In this section, a tracking system for a moving camera moving object system is proposed with uncertainties in both the camera and the object pose [5]. The system applies an EKF estimation for the state of the system. Typically the moving camera-moving target configuration arises from the eye-in-hand configuration, with the camera attached to the robot end-effector. Based on this, it is assumed that the end-effector motion can be measured by using incremental encoders, whereas the visual measurements give the relative pose between the end-effector and the target object. The model assumes that the target has reasonably constant velocity, with small unknown acceleration.

Some of the most important limitations for the use of visual tracking in robotics context are the noisy measurements, the low sample rate, and the latency of the vision system. These factors are often neglected in the development of the systems. The approach presented in this section takes these explicitly into account by integrating the encoder and visual measurements to provide a high sample rate and high accuracy estimates of the object motion. In addition, a theoretical approach of virtual measurements is proposed, which solves the problems related to the non-unique representations of 3-D rotation.

To describe the basic idea of tracking without any additional difficulties due to 3-D geometry, the integration method is first applied to a 1-D case, where the measured quantity is the distance between the end-effector and the target. Finally, a six degree-of-freedom case is proposed.

3.3.1 1-D tracking

In 1-D tracking, the state vector $\boldsymbol{x}$ contains 1-D positional data of the end-effector and the position and velocity of the object,

$$\boldsymbol{x} = \begin{pmatrix} d_1 & d_2 & \dot{d}_2 \end{pmatrix}^T , \tag{3.26}$$

where d_1 is the location of the end-effector, d_2 the location of the object, and $\dot{d}_2$ the velocity of the object. The velocity of the end-effector $\dot{d}_1$ can be read from the positional encoders. Interpreting the velocity information as the input to the system $\boldsymbol{u} = \dot{d}_1$, the system can be modeled by

$$\boldsymbol{x}(t+1) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{u}(t) + \boldsymbol{w}(t). \tag{3.27}$$

In this case, the prediction is linear with respect to both the previous state and the control input. The state transition matrix $\boldsymbol{F}$ is

$$\boldsymbol{F} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} , \tag{3.28}$$

where Δt is the time interval of predictions, describing the constant velocity model for the target object. The control input matrix $\boldsymbol{G} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}$ describes the relationship between position encoder values and change in the end-effector position. The model covariance matrix $\boldsymbol{Q}$ can be written as

$$\boldsymbol{Q} = \begin{pmatrix} \sigma_u^2 & 0 & 0 \\ 0 & \frac{1}{4}\Delta t^4 \sigma_d^2 & \frac{1}{2}\Delta t^3 \sigma_d^2 \\ 0 & \frac{1}{2}\Delta t^3 \sigma_d^2 & \Delta t^2 \sigma_d^2 \end{pmatrix} , \tag{3.29}$$

where σ_u^2 describes the uncertainty of the encoder measurement, and σ_d^2 an unknown uncertain acceleration affecting the target object, modeled as a discrete time Gaussian random process. This allows the modeling of small changes to the velocity of the target. At each time step, the error covariance $\boldsymbol{P}$ is predicted for the next time step:

$$\boldsymbol{P}(t+1) = \boldsymbol{F}\boldsymbol{P}(t)\boldsymbol{F}^T + \boldsymbol{Q}^T. \tag{3.30}$$

The distance between the end-effector and the object is measured with a vision system using a linear measurement model

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{v}(t), \tag{3.31}$$

where the measurement matrix $\boldsymbol{H} = \begin{pmatrix} -1 & 1 & 0 \end{pmatrix}$ and the measurement covariance $\boldsymbol{S} = \sigma_v^2$, where σ_v^2 is the uncertainty of the visual measurement. These models can now be directly used in a Kalman filter to derive optimal estimates for the position.

3.3.2 6-DOF tracking

In 3-D tracking, the relative 3-D pose of the object with respect to the end-effector is measured with vision. The tracked object is assumed to be moving with a constant velocity. The end-effector motion is measured by the encoder.

The 18-dimensional state vector $\boldsymbol{x}$ contains 3-D pose information of both the end-effector and the object, and the velocity of the object,

$$\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1^T & \boldsymbol{x}_2^T & \dot{\boldsymbol{x}}_2^T \end{pmatrix}^T, \tag{3.32}$$

where $\boldsymbol{x}_i = \begin{pmatrix} x_i & y_i & z_i & \phi_i & \theta_i & \psi_i \end{pmatrix}^T$. Because of the problem of non-unique angles, the approach from [102] is adopted again, and the orientation of the objects is stored externally, outside the EKF state, so that ϕ, θ, and ψ only represent incremental changes to the orientation. After each update step of the EKF, the rotation angles are integrated into rotation matrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ (for the end-effector and the target, respectively), and the rotation angles are reset to zero. The homogeneous transformation describing the pose of the end-effector in the world coordinates can now be written as

$$^{W}\boldsymbol{T}_{EE} = \begin{pmatrix} & x_1 \\ \boldsymbol{R}_1 \boldsymbol{R}(\phi_1, \theta_1, \psi_1) & y_1 \\ & z_1 \\ \boldsymbol{0} & 1 \end{pmatrix}, \tag{3.33}$$

where $\boldsymbol{R}(\cdot)$ is the function converting immediate angles to a rotation matrix. A similar form applies to the target object pose.

Now, the 3-D system can be modeled with

$$\boldsymbol{x}(t+1) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{u}(t) + \boldsymbol{w}(t), \tag{3.34}$$

where $\boldsymbol{u}$ is the output of the encoders in joint coordinates,

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{I}_6 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_6 & \Delta t \boldsymbol{I}_6 \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}_6 \end{pmatrix}, \tag{3.35}$$

and

$$\boldsymbol{G} = \begin{pmatrix} \boldsymbol{J}(\boldsymbol{\alpha}) & \boldsymbol{0} \end{pmatrix}, \tag{3.36}$$

where $\boldsymbol{J}(\boldsymbol{\alpha})$ is the robot Jacobian, dependent on the current configuration $\boldsymbol{\alpha}$. This allows the modeling of a general robot, as long as the robot Jacobian is available. The model covariance matrix $\boldsymbol{Q}$ is now

$$\boldsymbol{Q} = \begin{pmatrix} \boldsymbol{J}\boldsymbol{J}^T \Sigma_u & \boldsymbol{0}_6 & \boldsymbol{0}_6 \\ \boldsymbol{0}_6 & \frac{1}{4}\Delta t^4 \Sigma_d & \frac{1}{2}\Delta t^3 \Sigma_d \\ \boldsymbol{0}_6 & \frac{1}{2}\Delta t^3 \Sigma_d & \Delta t^2 \Sigma_d \end{pmatrix}, \tag{3.37}$$

where

$$\Sigma_u = \begin{pmatrix} \sigma_{ux}^2 \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \sigma_{u\phi}^2 \boldsymbol{I}_3 \end{pmatrix} \text{ and } \Sigma_d = \begin{pmatrix} \sigma_{dx}^2 \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \sigma_{d\phi}^2 \boldsymbol{I}_3 \end{pmatrix}, \tag{3.38}$$

where σ^2_{ux} and σ^2_{dx} describe the encoder and process uncertainties for the translation, and $\sigma^2_{u\phi}$ and $\sigma^2_{d\phi}$ the uncertainties for the rotations.

The visual system provides a measurement describing the relative pose of the target object with respect to the end-effector, ${}^{EE}\hat{\boldsymbol{T}}_O$. The rotation and translation parts of the measured relative pose are considered separately below.

The mapping between the current state and the measured relative translation from the end-effector to the target ${}^{EE}\hat{\boldsymbol{t}}_O$ can be written as

$$ {}^{EE}\hat{\boldsymbol{t}}_O = {}^{EE}\boldsymbol{R}_W \left({}^{W}\boldsymbol{t}_O - {}^{W}\boldsymbol{t}_{EE} \right). \tag{3.39} $$

Next, noting that ${}^{EE}\boldsymbol{R}_W = (\boldsymbol{R}_1 \boldsymbol{R}(\phi_1, \theta_1, \psi_1))^T$, ${}^{W}\boldsymbol{t}_O = \begin{pmatrix} x_2 & y_2 & z_2 \end{pmatrix}^T$, and ${}^{W}\boldsymbol{t}_{EE} = \begin{pmatrix} x_1 & y_1 & z_1 \end{pmatrix}^T$, the measurement model for translation can be written as

$$ \boldsymbol{y}_T(t) = \boldsymbol{H}_T \boldsymbol{x}(t), \tag{3.40} $$

where

$$ \boldsymbol{H}_T = \begin{pmatrix} -{}^{EE}\boldsymbol{R}_W & \mathbf{0}_3 & {}^{EE}\boldsymbol{R}_W & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{pmatrix}. \tag{3.41} $$

Contrary to translation, the arithmetic difference between the measured and predicted angles does not have a geometric explanation, unless either one of these is equal to zero. By introducing a technique called “virtual measurement”, this problem can be avoided by making the predicted angles zero.

Similar to translation, the mapping between the current state and the measured relative rotation from the end-effector to the target ${}^{EE}\hat{\boldsymbol{R}}_O$ is defined as

$$ {}^{EE}\hat{\boldsymbol{R}}_O = {}^{EE}\boldsymbol{R}_W {}^{W}\boldsymbol{R}_O, \tag{3.42} $$

where ${}^{EE}\boldsymbol{R}_W = (\boldsymbol{R}_1 \boldsymbol{R}(\phi_1, \theta_1, \psi_1))^T$, and ${}^{W}\boldsymbol{R}_O = \boldsymbol{R}_2 \boldsymbol{R}(\phi_2, \theta_2, \psi_2)$. Now, (3.42) can multiplied by ${}^{EE}\hat{\boldsymbol{R}}_O^T$ to get

$$ (\boldsymbol{R}_1 \boldsymbol{R}(\phi_1, \theta_1, \psi_1))^T \boldsymbol{R}_2 \boldsymbol{R}(\phi_2, \theta_2, \psi_2) {}^{EE}\hat{\boldsymbol{R}}_O^T = \boldsymbol{I}. \tag{3.43} $$

By introducting the function $\boldsymbol{\phi}(\cdot)$ converting a rotation matrix to immediate angles, and applying the function on both sides of (3.43), the virtual measurement function is obtained

$$ \boldsymbol{\phi} \left((\boldsymbol{R}_1 \boldsymbol{R}(\phi_1, \theta_1, \psi_1))^T \boldsymbol{R}_2 \boldsymbol{R}(\phi_2, \theta_2, \psi_2) {}^{EE}\hat{\boldsymbol{R}}_O^T \right) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}; \tag{3.44} $$

linking the virtual measurement, left-hand side, to the predicted angles, on the right-hand side. The frames related to the virtual measurement are shown in Fig. 3.10. From the predicted end-effector frame $\boldsymbol{EE}$ it is possible to get to the predicted object frame $\boldsymbol{O}$ through the world frame $\boldsymbol{W}$. The transform ${}^{\boldsymbol{EE}}\boldsymbol{T}_{\hat{\boldsymbol{EE}}}$ is the residual between the measured pose and the prediction.

The left-hand side of (3.44) can be denoted by $h_R(\phi_1, \theta_1, \psi_1, \phi_2, \theta_2, \psi_2, {}^{EE}\hat{\boldsymbol{R}}_O^T)$. Now, in order to use the virtual measurement in the EKF, only the gradient $\boldsymbol{H}_R$ of the virtual measurement function is needed with respect to the system orientation parameters, $\boldsymbol{H}_R =$

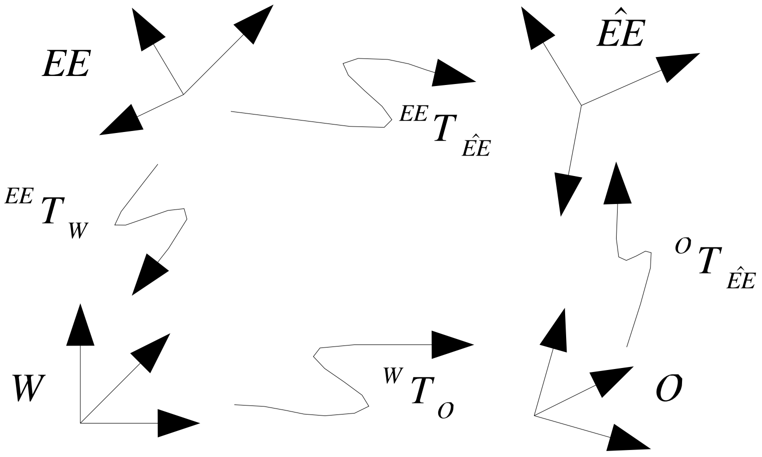


Figure 3.10: Coordinate frames linking the virtual measurement to prediction.

$\frac{\partial h_R(\cdot)}{\partial \boldsymbol{\phi}}$, where $\boldsymbol{\phi}$ denotes the set of orientation parameters. The gradient can be calculated analytically from (3.44), or estimated numerically. Thus, the visual measurements can be collected into one model:

$$\boldsymbol{y}(t) = \boldsymbol{H} = \begin{pmatrix} \boldsymbol{H_T} \\ \boldsymbol{H_R} \end{pmatrix} \boldsymbol{x}(t), \tag{3.45}$$

where

$$\boldsymbol{H}_R = \begin{pmatrix} \boldsymbol{0}_3 & \frac{\partial h_R(\cdot)}{\partial \boldsymbol{\phi_1}, \boldsymbol{\theta_1}, \boldsymbol{\psi_1}} & \boldsymbol{0}_3 & -\frac{\partial h_R(\cdot)}{\partial \boldsymbol{\phi_2}, \boldsymbol{\theta_2}, \boldsymbol{\psi_2}} & \boldsymbol{0}_3 & \boldsymbol{0}_3 \end{pmatrix}. \tag{3.46}$$

The covariance of the visual measurement is

$$\boldsymbol{S} = \begin{pmatrix} \sigma_{vx}^2 \boldsymbol{I}_3 & \boldsymbol{0}_3 \\ \boldsymbol{0}_3 & \sigma_{v\phi}^2 \boldsymbol{I}_3 \end{pmatrix}, \tag{3.47}$$

where σ_{vx}^2 is the uncertainty for the position of the visual measurement and $\sigma_{v\phi}^2$ the uncertainty for the rotation.

3.3.3 Handling the latency and different sampling rates of the sensors

To compensate for the latency of vision, the system shown in Fig. 3.11 is used. The encoder measurements are delayed for the amount of the latency so that the two are synchronized when fed to the Kalman filter. To take the delay of the filter into account, a modified prediction model (bottom of Fig. 3.11) is used to estimate the current, non-delayed state. In modified prediction, the time interval Δt in the state transition matrix $\boldsymbol{F}$ is substituted by the latency τ, and instead of the encoder measurement $\boldsymbol{u}$, an integral of the encoder velocities over the period of the latency is used:

$$\int_{t-\tau}^{t} \boldsymbol{v}_{enc}, \tag{3.48}$$

where $\boldsymbol{v}_{enc}$ are the velocities given by the encoders. The algorithm for the latency compensation is the same as presented in Alg. 7, but in the external prediction model the integral of the encoder velocities is also taken into account.

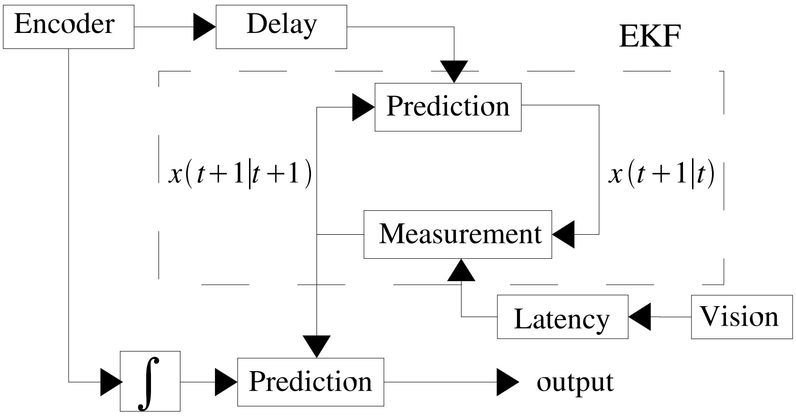


Figure 3.11: Latency compensation.

3.3.4 Experiments with a 1-D servo bench

Experiments with a 1-D servo bench were performed to study the proposed estimation approach in a control problem with real equipment. Two hydraulic cylinders were set up, one for the object and one for the end-effector. The motion of the object and the end-effector were controlled by velocity controllers.

Figure 3.12 presents a proposed controller, which controls the distance between the end-effector and a target to a desired value. An incremental position encoder and a vision system are used as sensors. The relative distance d measured by the vision system and end-effector velocity $\dot{r}$ measured by the encoder are fed to the EKF. The control loop consists of two nested controllers, an outer controller for setting the desired velocity $\dot{r}_d$ for an inner velocity controller. In the outer controller, the estimated distance $x_o - x_{ee}$ is controlled towards the desired distance d^* by a proportional controller with gain K_v. In addition, an estimate of the object velocity $\dot{x}_o$ is fed forward and added to the desired velocity $\dot{r}_d$. The inner velocity control is application-specific and depends on the robot configuration and the actuator dynamics. Due to non-linear system dynamics, there is a small steady-state error in set velocity in the experiment.

A vision system was used to measure the relative distance from the object to the end-effector. Red markers were affixed on both the end-effector and the target, and tracked in the image. The image locations of the markers were then used to determine the distance. The tracking was based on thresholding and connected component labeling [96]. The camera was attached perpendicular to the servo bench, but eye-in-the-hand configuration, where the camera is attached to the end-effector, would also be possible. The end-effector can only move in one dimension, and the object is always on the same line in Cartesian space. The lines in Cartesian space are mapped as lines in the image plane due to the perspective projection [38]. As lens distortion is considered negligible, the markers are always on the same line in pixel coordinates. A recursive flood fill algorithm is initiated at each pixel of the line. The algorithm goes through connected neighboring pixels and gives them a specified color. Pixel connectivity is defined by thresholding, which is performed in red-green-blue (RGB) space by using specified ranges of intensities in each RGB channel. The centroids of the two largest connected regions are then calculated in pixel coordinates.

The vision system can be calibrated by moving the end-effector to known positions and saving the corresponding pixel values. In a full 3D camera calibration matrix there are 11 degrees of freedom. In 1D there are only 5 degrees of freedom, and thus three calibration points are sufficient. Pixel coordinates of the image features are saved in each end-effector position, and camera calibration matrix $\boldsymbol{M} \in \Re^{3\times 2}$ is formed. If X_i are set of homogeneous 1D points and x_i are corresponding 2D points, the projection matrix $\boldsymbol{M}$ can be solved from $x_i = \boldsymbol{M}X_i$. In 1D case the projection equations reduce to

$$\begin{array}{c} m_{11}X_i + m_{12} - x_i m_{31}X_i - x_i m_{32} = 0 \\ m_{21}X_i + m_{22} - y_i m_{31}X_i - y_i m_{32} = 0 \end{array}. \tag{3.49}$$

Stacking (3.49) for each calibration point, the projection matrix $\boldsymbol{M}$ can be solved from the resulting system of linear equations using singular value decomposition. The homogeneous position of the end-effector $\boldsymbol{H}_{ee} = (u_{ee}, w_{ee})^T$, and the location of the object $\boldsymbol{H}_o = (u_o, w_o)^T$ are then

$$\boldsymbol{H}_{ee} = \boldsymbol{M}^{+} \begin{pmatrix} x^i_{ee} \\ y^i_{ee} \\ 1 \end{pmatrix}, \qquad \boldsymbol{H}_o = \boldsymbol{M}^{+} \begin{pmatrix} x^i_o \\ y^i_o \\ 1 \end{pmatrix}. \tag{3.50}$$

where $\boldsymbol{M}^{+}$ is the pseudoinverse of the projection matrix, (x^i_{ee}, y^i_{ee}) are the pixel coordinates of the end-effector, and (x^i_o, y^i_o) the pixel coordinates of the object. The Cartesian positions of the object and end-effector can then be solved from $x_{ee} = u_{ee}/w_{ee}, x_o = u_o/w_o$.

The vision system operates on a 1 GHz Linux PC equipped with a frame grabber and a color video camera. The frame-rate of the system is 25 Hz with a resolution of 720×568. The vision system calculates the distance to the object and sends it to the controller by using an RS-232 serial connection. The total latency of the vision system and serial data transmission is measured to be approximately 55 ms. Most of the latency is caused by the exposure and grabbing of the image, causing 40 ms of latency. The rest is caused by image processing and data transmission.

The performance of a control response based on EKF estimate was compared to that of a direct measurement from the visual system. The uncertainty of the visual measurement can be estimated by calculating the variance of the distance measurements when the object and the end-effector are stationary. The process uncertainty can be set very small if the motion of the object is constant. The uncertainty of the end-effector was within the encoder resolution, which was $1\mu m$. The initial guess for the object velocity was set to zero, but the object velocity estimated by the Kalman filter was allowed to converge before the control algorithm was initiated.

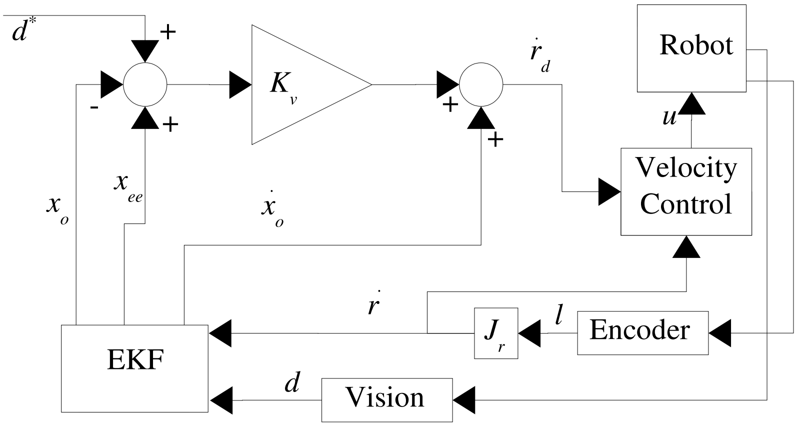


Figure 3.12: Control schematic.

In the first experiment the object was moving away from the end-effector with a constant set velocity of 10mm/s. Because of the steady state error in the velocity controller, the actual object velocity was approximately 9mm/s. Figure 3.13 shows the response of the controllers based on a direct visual measurement and the EKF estimate. The desired distance between the end-effector and the object d^* was set to 30mm. The control algorithm based on the direct visual measurement never reached the desired distance. There is a considerable steady state error that increases if the object velocity is increased. With the EKF the steady state error is smaller, as the predicted object measurement is added to the desired velocity. The steady state error still present in the EKF experiment is due to the steady state error in the velocity controller.

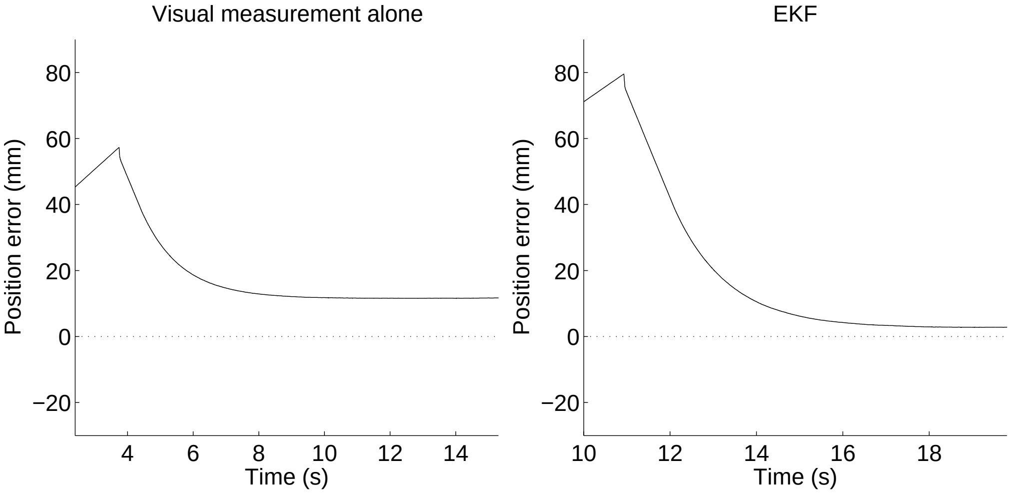


Figure 3.13: Object moving away from the end-effector.

In the second experiment shown in Fig. 3.14, the object was moving towards the end-effector with a constant set velocity of 20mm/s. Visual measurement alone cannot keep the desired distance of 30mm, and there is a significant steady state error allowing the end-effector to move dangerously close to the object. With the EKF the steady state error is considerably smaller, and can be eliminated by improving the velocity controller.

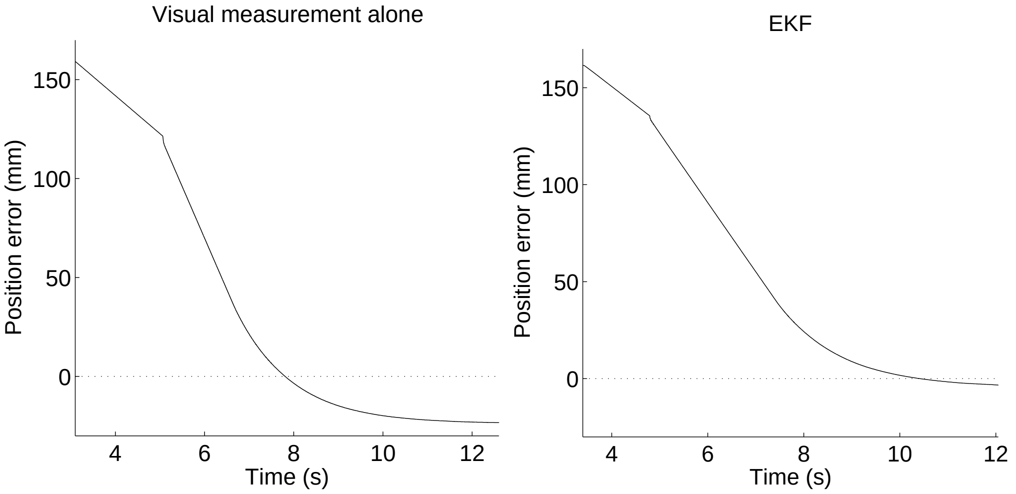


Figure 3.14: Object moving towards the end-effector.

In the third experiment shown in Fig. 3.15, the object was moving along a sinusoid pattern. The figure demonstrates that it is possible to track an arbitrary moving object, even when the constant velocity assumption of the EKF is false, if the process uncertainty is set high enough. It can be seen in Fig. 3.16 that the phase shift between the end-effector and the object location is significantly higher if the direct visual measurement is used as a control input instead of the EKF estimate. Also the amplitude of the end-effector motion is smaller than the amplitude of the object motion if the visual measurement is used directly.

3.3.5 Experiments with simulations

The proposed 6-DOF EKF-model was studied with simulations. The visual system was not modeled in detail, but Gaussian noise was added to the simulated measurements. The visual measurement consisted of the relative pose between the end-effector and the object. A Cartesian robot was used for simplicity, and therefore the robot Jacobian $\boldsymbol{J}$ was an identity matrix. The gradient of the virtual measurement function (3.44) was estimated numerically.

In the first experiment, both the object and the end-effector were moving with constant Cartesian and angular velocities. The initial guess for the velocities was set to zero and the uncertainties were initialized with high values. Figure 3.17 presents the estimated x, y and z Cartesian velocities of the target. The true object velocities were 20mm/s, -30mm/s and -20mm/s, respectively. It can be seen that the settling time for the object

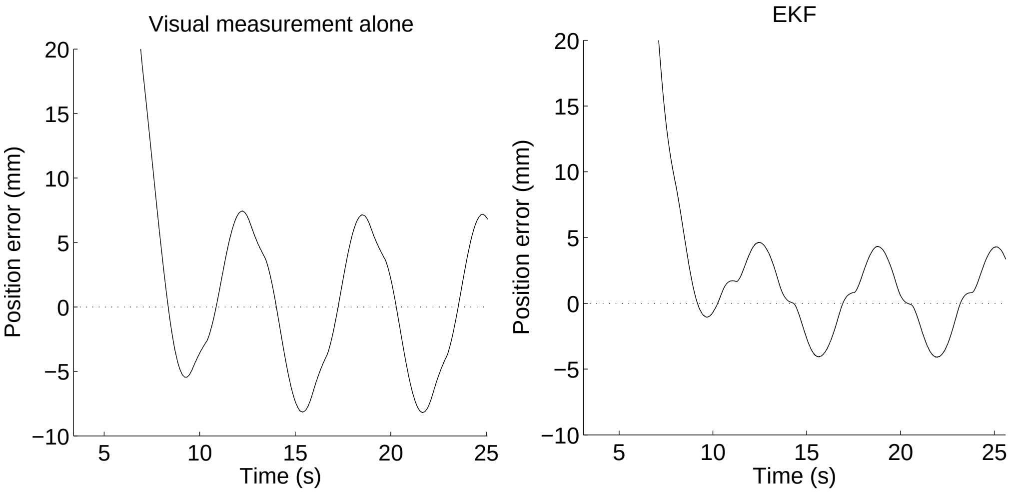


Figure 3.15: Sinusoid object movement.

velocities is less than two seconds. Figure 3.18 shows the estimated z-coordinate positions of the end-effector and the object. As the end-effector measurement is very precise, the error of the estimated end-effector location is negligible. In the estimated object position there is a small error before the estimate of the object velocity converges. The velocity of the object is first assumed to be zero, and therefore the estimate is jagged. The sampling rate of the visual measurement was again 40ms, but the latency was assumed to be zero for simplicity. This experiment demonstrates that the method of virtual measurements is applicable for 6-DOF estimation.

In the second experiment, the constant velocity assumption for the object movements was neglected. Instead, the object was allowed to follow a sinusoid track. In this experiment, the process uncertainties were set to correspond to the changing object velocities. It can be seen in Fig. 3.19 that the estimated position and velocity of the object follow the real position with a phase shift of approximately one second. This experiment demonstrates that the proposed approach improves the estimates significantly only if the initial assumptions of the object motion are not seriously violated.

It should be noted that a simplification has been made in that the covariance matrix for the visual measurement in the simulated 6-DOF case is assumed constant. In reality, the uncertainty depends on the poses of the end-effector and the object.

3.4 Summary and discussion

In this chapter, a moving camera moving target tracking method integrating visual and encoder information using an extended Kalman filter have been proposed. Noisy, low-sample-rate visual measurements with a considerable latency have been fused with encoder information to give high sample-rate high-accuracy estimates of the target motion. The applicability of the approach has been demonstrated with experiments and simula-

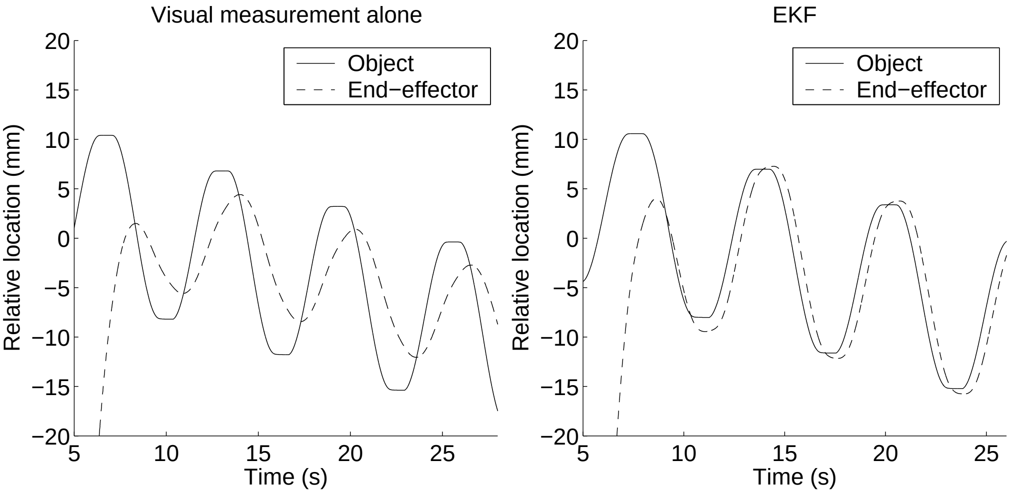


Figure 3.16: Relative locations of the end-effector and the object.

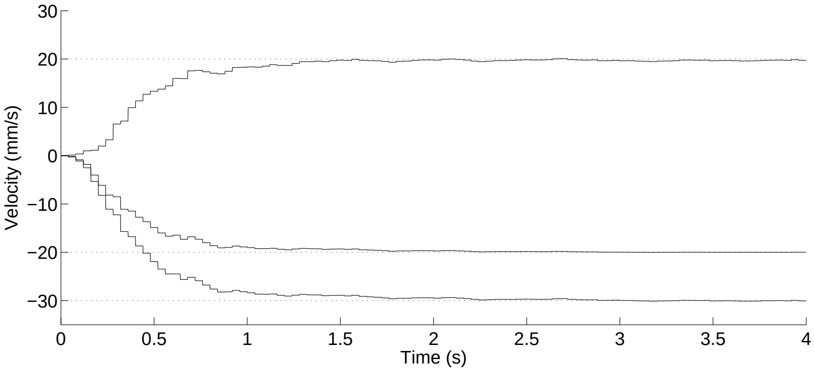


Figure 3.17: Cartesian velocities of the object.

tions, which show that the proposed approach is valid and can significantly improve the performance of a vision-based controller.

Visual control allows manipulation in dynamic environments with uncertainties. Although vision offers a sensor modality with high information content, it also suffers from limitations, such as low sample rate, high sensor delay and uncertain measurements. The proposed 6-DOF EKF tracking method addresses these three problems. The relative pose from the camera to a moving object is measured with vision. High frequency proprioceptive information of the end-effector motion and the visual measurement are fused in the EKF to estimate the absolute pose of the object in the world coordinates as well as the translational and angular velocities. The sensor delay of the visual measurement is taken into account explicitly. The proprioceptive measurements are delayed in the EKF so that they are synchronized with the visual measurement. An external prediction model then estimates the pose of the object for the present time. The uncertainties in

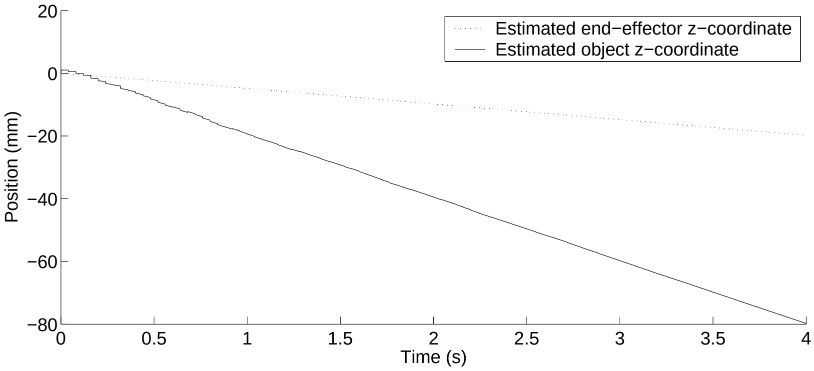


Figure 3.18: End-effector and object z-coordinate positions.

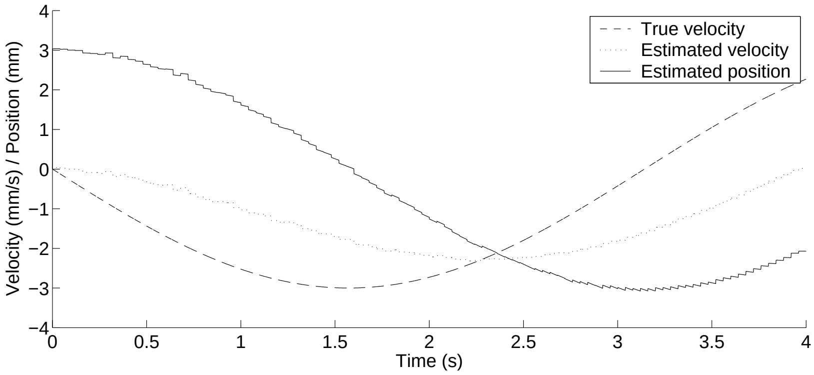


Figure 3.19: Sinusoid movement.

the end-effector pose are assumed negligible compared to the uncertainty of the visual measurement. If the end-effector pose needs to be estimated as well, the EKF model with 18-dimensional state vector proposed in [5] can be used.

Visual servoing is often either image-based or position-based. In image-based visual servoing, image features are used to control a robot directly. In position-based visual servoing, the relative pose between the robot and the target object is estimated and the robot is controlled accordingly. Visual servoing is often accompanied with an EKF to filter either the image feature locations or the relative pose. The proposed method differs from the traditional EKF frameworks used in visual servoing by taking into account the information of the end-effector motion. In the approach, the state vector is the absolute pose of the object in the world accompanied with corresponding linear and angular velocities. This allows estimation of the absolute pose of the object in the world frame compared to the relative pose between the end-effector and the object usually considered in KF-frameworks [103][75]. Simulated results on using absolute pose as the state vector have been presented earlier for eye-in-hand configuration [59]. However, using such a system on a real robot requires the sensor delay of the visual measurement to be compensated for. There are several benefits in using the absolute pose as a state vector.

First, the prediction model is more linear, as the motion of the end-effector does not affect the state. If the relative pose is used as a state vector, the motion of the end-effector alters the state, and even if the object is stationary, the constant velocity assumption for the state might not hold. Second, determining the covariance matrix for the process uncertainty is easier in the Cartesian world coordinates, as only the motion of the object must be considered and not the relative motion between the end-effector and the object in image space. Finally, compensation for the latency of the visual measurement can be accomplished.

The single camera setup induces different errors in measurements along different Cartesian coordinate axes. The distance to the object is difficult to measure accurately, as small changes in pixel coordinates cause a large change in distance. Translations perpendicular to the optical axis of the camera, on the other hand, are less noisy, as translation on the image plane causes a large change in pixel coordinates. The same applies for rotations. Rotation measurements about the camera optical axis are accurate, but when the object is rotated about an axis perpendicular to the optical axis, only small changes in pixel coordinates can be seen. For this reason it is essential to model the uncertainties of the visual measurement in detail. By modelling the covariance of the visual measurement so that it takes into account the cross covariance terms of the measurement, it is possible to fuse different sensors together. This chapter has validated the previous analysis of measurement errors in visual servoing experimentally [52].

The approach was validated by experiments with a 5-DOF parallel hydraulic manipulator. The experiments verified that the method can be used for controlling a robotic manipulator, when visual measurement alone is too noisy for the task. The method combines sensors with different sampling rates, providing high frequency efficient and robust control reducing the phase shift, compared to a pure visual feedback due to sensor delay compensation. On the other hand, the experiments indicated that some degrees of freedom are difficult to control very accurately with visual feedback only, and thus the use of other sensory modalities together with vision appears an appealing alternative.

Chapter IV

Fusion of sensors without common representation

When using a single measurement at each time instant, as is typical in visual servoing, the uncertainty in visual measurements can cause undesired oscillations and hinder the accuracy. However, even when adequately filtered, visual measurement is often not accurate in all degrees of freedom. Only the object translations perpendicular to the camera optical axis can be determined accurately. The object translation along the camera optical axis is difficult to measure, as even a big change in the object distance induces only small changes in the image, due to limited perspective effects. The same applies for rotations, as rotation around the camera optical axis can be determined accurately, whereas rotations around the off axes yield only diminutive changes in the image. To alleviate these problems, vision can be complemented by other sensor modalities. One attractive complementary option is to use a tactile or force sensor to probe the local shape of an object. When the tooltip is in contact with the object and the pose of the tooltip can be determined with direct kinematics, information about the object can be extracted. However, a single tooltip measurement can only give one point on the object surface. Without other information this measurement will be difficult to use, as it is not known which location of the object the measurement has been taken at. Also if the object is moving, the point of the contact can move even if the position of the tooltip is stationary.

Combining a force sensor with vision would seem appealing, as these two sensors can complement each other. Since force and vision measure fundamentally different sensor modalities, the information from these sensors cannot be fused directly. Vision can extract the full pose of a known object with respect to the camera, but a force sensor can measure forces only locally. When the force sensor is used only to detect if the tooltip is in contact with the object, no other information can be gained. Combining this binary information with visual measurement requires that both measurements can be related to a common estimated state. This can be achieved, as the incremental encoders or joint angle sensors of the robot can determine the pose of the robot end-effector in the world coordinates. If also the hand-eye calibration of the camera and the tool and the object geometries are known, both visual and tactile measurements can be related to the object pose in the world coordinate frame. However, a single tooltip measurement can only give constraints to the pose of the object but not the full pose. Therefore, a single measurement is meaningless unless it can be fused with other sensor modalities or over time. Combining several sensor modalities or multiple measurements over time can reduce the uncertainty of the measurements, but in order to fuse the measurements, the uncertainty of each individual measurement must be estimated. Also the sensor delay of the visual measurements must be taken into account when fusing the measurements. Especially the eye-in-hand configuration requires accurate synchronization of the proprioceptive information and visual measurement. Otherwise vision will give erroneous information when the end-effector is in motion.

This chapter addresses the problem of combining sensors without an inherent common representation. It is proposed how visual and tactile measurements can be fused together, taking into account the uncertainty of each individual measurement. A model-based pose estimation algorithm is used to extract the unknown pose of a moving target. The uncertainty of the pose depends on the uncertainty of the measured feature points in the image plane, and this uncertainty is projected into Cartesian space. A tooltip measurement is used to probe the local shape of the object by moving on the object surface and keeping a constant contact force. An EKF is then used for incorporating the multimodal measurements with different uncertainties and sampling rates. The EKF allows reduced uncertainties of vision-based estimates and interpolation between low frequency visual measurements, making it possible to track a moving target. The expected future target pose can also be predicted with the EKF to alleviate inherent latencies in the vision system. When proprioceptive and visual measurements are synchronized in the EKF, the estimate of the target pose becomes delayed. To compensate for this delay, the future target pose is predicted with the EKF. Most importantly, to the author's knowledge the use of contact information to compensate for the uncertainty of vision in estimating the object pose while the tooltip is sliding on the object surface has not been proposed before.

Previous work on combining haptic information with vision uses primarily the two sensors separately. Vision is used to generate a 3D model of an object and a force sensor to extract physical properties, such as the stiffness of the object [10]. Also stereo vision has been used in the modelling of the object [54], and active touch to determine how the object deforms. Pomares et al. [80] combined a force sensor and an eye-in-hand camera using structured light to detect changes in the contact surface. Vision is first used to detect zones likely to have discontinuities on the surface and the force sensor is used for verifying the discontinuity. Force sensors have also been used for determining linking structures and functions of objects, such as scissors or pliers [99]. In a work by Tanaka et al. [92] vision is used to extract the 3D shape and pose of an object. The robot hand is then allowed to move the object and the contact force is measured by a force sensor. The mass of the object is then extracted using known friction coefficients. Ueda et al. [98] propose a "touch and see" system consisting of a range sensor and a force-feedback sensor. One robot hand is equipped with a range sensor and a CCD camera and another robot with a force sensor. A deformable object is pushed by the haptic robot, and a range finder observes how the deformed object returns to its original shape when the force is released. Also arrays of tactile sensors have been used to model the 3D shape of an object with a laser range finder first observing the target object [78].

In this chapter, contact information is combined with vision to extract more accurate information about the object pose. A constant stiffness for the object is assumed, making it possible to use the tooltip measurements for determining the object's position and orientation. The method is independent of friction and can be used even when the tooltip is sliding on the object surface while the object is in motion.

4.1 Tooltip probing

Force or tactile sensors allow accurate measurements of the object surface when in contact. By probing the surface of the object with a tooltip, it is possible to extract the location of the contact point. In order to fuse this information with the model of the object in the world coordinates, the position of the tooltip must be known in the world coordinates. If this information is known and the surface of the object is assumed to be planar, the position of the tooltip can be used as a measurement. Even though this measurement does not define the pose of the object completely, it can complement the information given by vision. The rotation around camera off-axes and the distance to the object are difficult to measure with vision. However, probing by the tooltip can give accurate information on these very same degrees of freedom. A single tooltip measurement does not give other information than the distance to the object, but several measurements in different locations of the object can be combined to produce an accurate estimate of the object rotations. It should be noted that the rotation around the tool axis cannot be measured by tooltip probing, but if the camera is on the same axis as the tool, accurate measurements from the vision are obtained.

Next, a measurement model for the tactile measurement is defined. The surface of the object is assumed planar. Thus, the distance d from the tooltip P_t to the plane is

$$d = \frac{|{}^W\boldsymbol{n}\cdot \overrightarrow{P_t P_0}\,|}{|{}^W\boldsymbol{n}|}, \tag{4.1}$$

where ${}^W\boldsymbol{n} = {}^W\boldsymbol{R}_O\boldsymbol{n}$ is the normal vector of the surface and P_0 a point on the object plane. P_0 can be defined as the origin of the object in the world coordinates, which is stored in the state vector of the EKF. Also the rotation matrix ${}^W\boldsymbol{R}_O$ can be extracted from the state. When the tooltip is in contact with the object, the distance from the object to the tooltip is assumed zero. For a rigid non-deformable object this assumption is reasonable. The position of the tooltip P_t can be obtained in the world coordinates, as the dimensions of the tool are measurable and can be assumed constant if the stiffness of the tool is high enough. Fig. 4.1 explains the tooltip measurement where the normal vector of the surface $\boldsymbol{n}$ is presented in the object coordinates.

To use tooltip measurement (4.1) in the EKF, the gradient of the measurement model, i.e. the distance function, must be calculated. The translational parameters of the distance function are independent of the rotation matrix ${}^W\boldsymbol{R}_O$, reducing the translational gradient to

$$\frac{\partial d}{\partial x, y, z} = {}^W\boldsymbol{n} \cdot \frac{\partial\, \overrightarrow{P_t P_0}}{\partial x, y, z}. \tag{4.2}$$

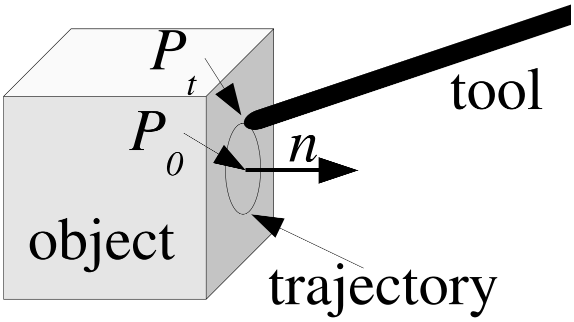


Figure 4.1: Tooltip probing.

By utilizing the chain rule, the partial derivatives of the rotation matrix ${}^{W}\boldsymbol{R}_O$ are obtained for each orientation parameter. The rotational gradients can be written as

$$\frac{\partial d}{\partial \phi, \theta, \psi} = \left(\frac{\partial^{W} \boldsymbol{R}_O}{\partial \phi, \theta, \psi} \boldsymbol{n} \right) \overrightarrow{P_t P_0} \ . \tag{4.3}$$

The 12 element gradient vector is then

$$\boldsymbol{H}_F = \begin{pmatrix} \frac{\partial d}{\partial x} & \frac{\partial d}{\partial y} & \frac{\partial d}{\partial z} & \frac{\partial d}{\partial \phi} & \frac{\partial d}{\partial \theta} & \frac{\partial d}{\partial \psi} & 0 & \dots & 0 \end{pmatrix}. \tag{4.4}$$

The visual measurements and the tooltip measurement can now be combined into one model

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) = \begin{pmatrix} \boldsymbol{H_T} \\ \boldsymbol{H_R} \\ \boldsymbol{H_F} \end{pmatrix} \boldsymbol{x}(t), \tag{4.5}$$

and the covariance then becomes

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_H & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_t^2 \end{pmatrix}, \tag{4.6}$$

where σ_t^2 is the variance of tooltip distance measurement modelling the deformability of the object and stiffness of the tool.

4.1.1 Tool center point calibration

Methods for determining the tool center point are extensively used in industry, and many patents have been filed for such methods over years. In this section, the method presented in [95] is extended by solving the parameters of an unknown calibration plane, as well as the tool center point. The method also takes explicitly into account the uncertainty of the measurements by providing a linear least squares estimate of the parameters. The tooltip is allowed to contact the calibration plane in several different places to solve the surface normal of the calibration plane. By tooltip probing, several measurements on the surface of the calibration plane are obtained. The coordinates of the end-effector position are stored for each measurement in data matrix $\boldsymbol{D}$. The data matrix is transformed to have a zero mean by subtracting the mean of the distribution from the data. A covariance matrix $\boldsymbol{C}$ is formed from the data matrix $\boldsymbol{D}$ and the best fitting plane can be solved by finding the eigen vectors of the covariance matrix $\boldsymbol{C} = \boldsymbol{D}\boldsymbol{D}^T$. The first two eigen vectors form the basis of the plane and the third eigen vector is the surface normal $\boldsymbol{n_c}$. Next the tooltip center point with respect to the end-effector, as well as the distance parameter d_c of the calibration plane are solved. The end-effector is allowed to contact the calibration plane by rotating the end-effector to different orientations and moving the tool towards the calibration plane slowly until contact occurs. Different poses during the tooltip centerpoint calibration are shown in Fig. 4.1.1. The pose of the end-effector is stored for each pose and the unknown parameters are estimated by a linear least squares fit. This can be done by solving a system of over-determined linear equations $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{c}$, where $\boldsymbol{x} = \begin{pmatrix} d_x & d_y & d_z & d_c \end{pmatrix}^T$ describe the unknown parameters, $\boldsymbol{M}$ is a $i \times 4$ matrix and $\boldsymbol{c}$ is a column vector. Each row of the matrix $\boldsymbol{M}$ is defined as

$$\begin{pmatrix} \boldsymbol{R_i}\boldsymbol{n_c} & 1 \end{pmatrix}, \tag{4.7}$$

and elements of the column vector $\boldsymbol{c}$ as

$$-\boldsymbol{n_c}^T\boldsymbol{t_i}, \tag{4.8}$$

where $\boldsymbol{R_i}$ is the end-effector rotation for each pose, $\boldsymbol{t_i}$ is the end-effector translation, and $\boldsymbol{n_c}$ is the surface normal of the calibration plane. The system of linear equations can be solved by singular value decomposition to obtain d_c and the tooltip center point location $\begin{pmatrix} d_x & d_y & d_z \end{pmatrix}^T$. The algorithm for the tool center point calibration is given in Alg. 8.

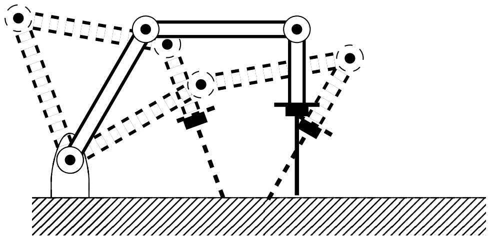


Figure 4.2: Different poses during the tooltip centerpoint calibration.

4.1.2 Handling the sensor delay and different sampling rates of the sensors

Due to different sampling rates, the tooltip measurement is performed more often than the visual measurement update. This can be done by setting the visual measurement terms in the Kalman gain matrix or the gradient matrix to zero, when there is no visual measurement.

To compensate for the sensor delay of vision, the system shown in Fig. 3.1 in the previous chapter is used. It should be noted that the tactile measurements are not taken into account in the external prediction model. If the visual sensor delay is great and the constant velocity assumption for the object movement does not hold, this may cause error in the prediction.

Algorithm 8 Algorithm for tool center point calibration.

```
for i=1 to number of translations do
    move the end-effector downwards along $z$-axis until contact occurs
    store the end-effector translation in a data matrix $\boldsymbol{D}$
    move the end-effector to the next place in $x$-$y$-plane
end for
substract the mean of the positions from the data matrix $\boldsymbol{D}$
compute the covariance of the data matrix $\boldsymbol{C} = \boldsymbol{D}\boldsymbol{D}^T$
find the eigen vectors of the covariance matrix $\boldsymbol{C}$
store the third eigen vector into surface normal vector $\boldsymbol{n_c}$
for i=1 to number of poses do
    move the end-effector downwards along $z$-axis until contact occurs
    get the end-effector rotation $\boldsymbol{R_i}$ and translation $\boldsymbol{t_i}$
    transform the surface normal $\boldsymbol{n_c}$ into homogeneous coordinates $\begin{pmatrix}\boldsymbol{R_i}\boldsymbol{n_c} & 1\end{pmatrix}$ relative
    to the end-effector
    store the homogeneous surface normal vector as $i$:th row of matrix $M$
    store $-\boldsymbol{n_c}^T\boldsymbol{t_i}$ as the $i$:th element of a column vector $c$
    move the end-effector to the next pose
end for
solve $\boldsymbol{x}$ from the system of linear equations $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{c}$ by singular value decomposition
the unknown parameters are now stored in vector $\boldsymbol{x} = \begin{pmatrix}d_x & d_y & d_z & d_c\end{pmatrix}^T$
tool center point is $\begin{pmatrix}d_x & d_y & d_z\end{pmatrix}$
```

4.1.3 Experiments

Experiments were conducted on a 5-DOF parallel hydraulic manipulator [61] shown in Fig. 3.2. A linear track was mounted on the robot frame to increase the work space of the robot, and the object to be tracked was affixed on the linear track. The absolute pose of the object with respect to the robot frame was estimated by using visual measurement alone in the EKF and by combining the visual and tactile measurements in the EKF. Both the robot tool and the target object were made of steel and very rigid.

The direct kinematics of the system is difficult to solve in parallel manipulators. Mapping from joint coordinates to pose of the end-effector is usually only obtainable through iterative methods. The direct kinematics problem was solved off-line by the Gauss-Newton method. However, the problem is solvable in real time [69].

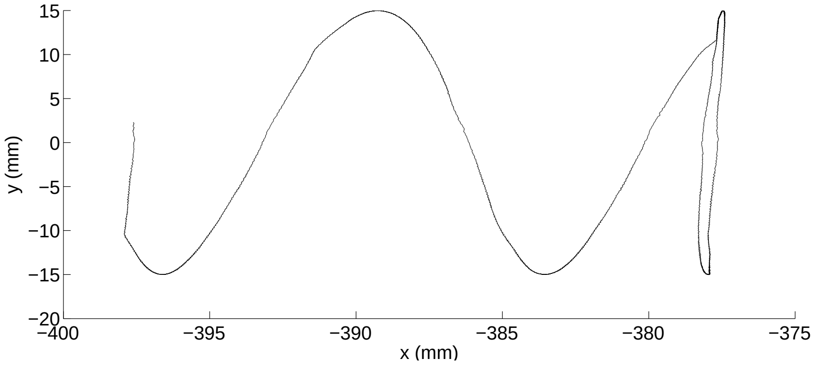
In the experiments, the end-effector was allowed to take contact to the rigid object in order to compare the EKF with and without tactile measurements. The end-effector followed a circular trajectory in the $y - z$-plane, as shown in Fig. 4.3. Force control was set up to control the third, x-axis, and keep a constant contact force. The linear track was moved with a constant velocity on the force controlled axis. In Fig. 4.4, it can be seen that the force controller can keep a constant contact force when the object is stationary, but when the object is in motion the controller will induce a steady state error. However, the contact is stable for the whole motion. The oscillation in Fig. 4.4 is partially due to measurement noise, partially caused by the force controller characteristics.

Figs. 4.5 and 4.6 illustrate the translations and rotations of the estimated object poses, respectively. True motion was only present in the translational x-coordinate. The linear track was allowed to move along the x-coordinate with constant velocities, as shown in Fig. 4.7. The apparent motion seen in the other figures is caused by imperfections in hand-eye and intrinsic camera calibration, as well as uncertain measurements. As there was only translational movement present, the process uncertainties for the rotations were set to zero. The tooltip measurement requires an initial estimate of the object pose, which can be estimated with vision. The tooltip measurement was switched on at $t = 20s$ once the vision had a coarse estimate of the object pose.

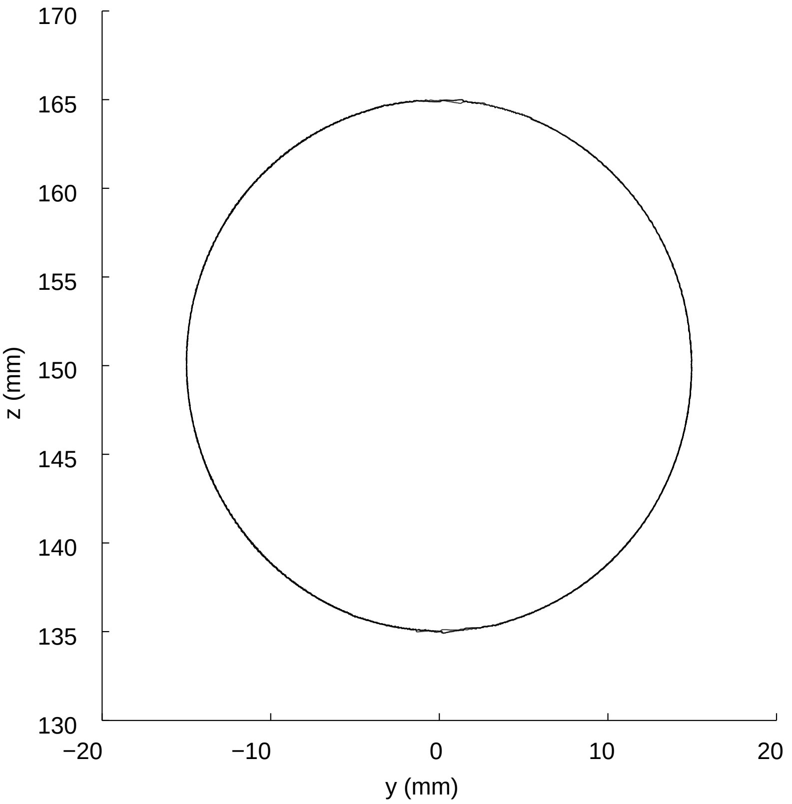
In Fig. 4.5 it can be seen that the tooltip measurement does not contribute to the estimate of the translational y- and z-coordinates. These axes are perpendicular to the tool axis, and therefore the tooltip measurement cannot give any additional information on the object pose. However, the visual measurement has only a small uncertainty on these axes according to the Hessian approximation, and they can be considered accurate even without the tooltip measurement. The x-axis, on the other hand, is on the tool axis and tooltip probing can compensate for the uncertain visual measurements. The EKF with tooltip measurement follows the linear table motion more accurately than the vision-only EKF estimate. When the velocity of the linear table changes, the EKF with tooltip measurement can adapt to the new velocity almost instantaneously, whereas the velocity estimate of the vision-only EKF oscillates and does not converge to the true velocity. The mean error for the EKF with tooltip measurement is $0.5255mm$ and for the vision-only EKF $1.0822mm$ for a time interval $t = 20s - 68s$.

In the estimated object rotations, the effect of the tooltip measurement is even more drastic. Once the tooltip measurement is switched on, the estimate of the object orientation quickly converges. The uncertainty of the visual measurement is very high for β- and γ-rotations, and the vision-only EKF estimate converges slowly. The visual measurements of the rotation around the camera optical axis (α) are much more precise, and even the vision-only EKF converges quickly. An interesting phenomenon can be observed in the α-axis rotation. Even though the tooltip measurement should not give any information on rotations around this axis, the estimate of the rotation still converges very quickly. This can be explained by the covariance matrix of the visual measurement. As the y- and z-axis are measured very accurately by vision, and the tooltip measurement constrains the β- and γ-rotations as well as the x-axis translation, the EKF converges to an accurate pose also for the α-axis.

The estimated uncertainties shown in Figs. 4.8 and 4.9 describe how the uncertainty propagates as more and more measurements are included in the estimate. As the covariance is represented as a 7×7 matrix, only the diagonal terms are presented in the figures and the rotations are converted into Euler angles. The figures clearly show much faster convergence rates when the tooltip measurement is used. The last experiment was conducted to find how the proposed method would perform in a control scenario. The estimate of the object pose was converted into an equation of a planar surface, and the distance between the plane and the tooltip was calculated. If the estimate of the object pose is directly fed to the robot controller, this experiment shows how much the controller set point differs from the surface of the object. In Fig. 4.10, three different distance estimates are compared. The pure visual estimate (dashed line) is very noisy and shown only for the first few seconds of the experiment. The vision-only EKF estimate (dotted line) filters the visual measurement considerably, but does not converge to zero distance. The EKF with tooltip measurement (solid line) can keep the distance very close to zero.



(a) Motion in $x-y$-plane.



(b) Motion in $y-z$-plane.

Figure 4.3: End-effector motion.

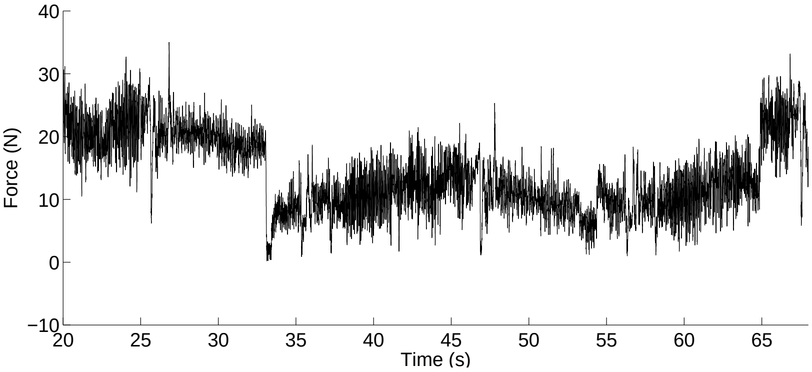


Figure 4.4: Contact force.

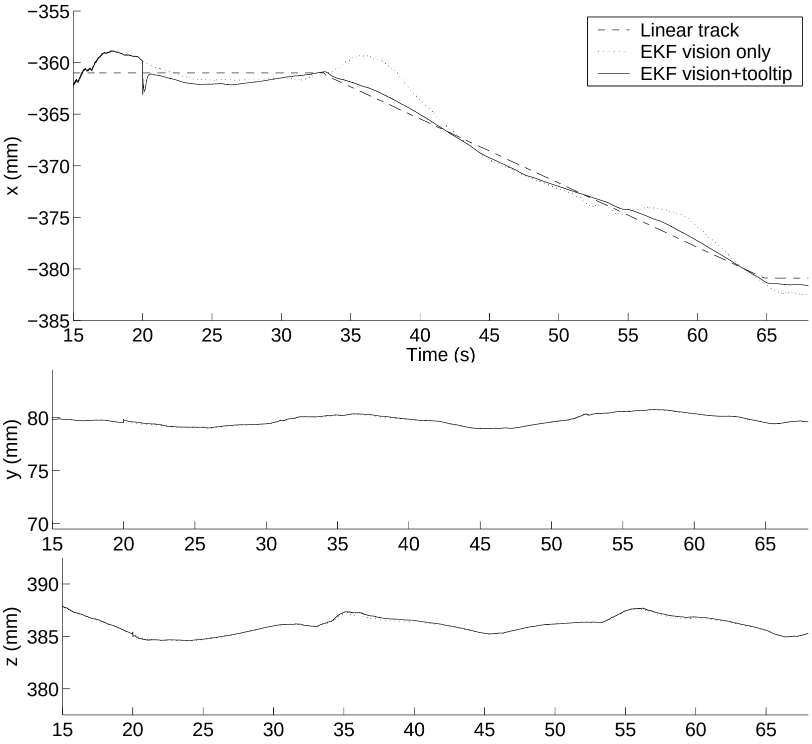


Figure 4.5: Estimated object translations.

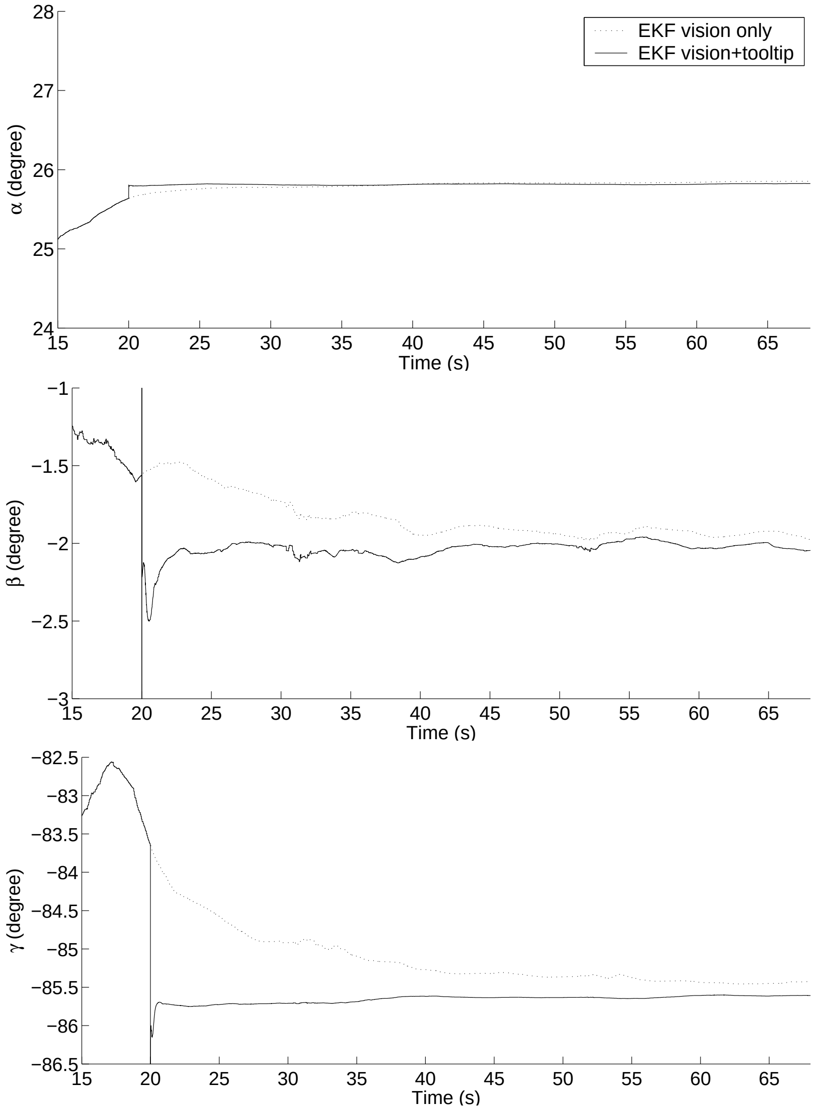


Figure 4.6: Estimated object rotations.

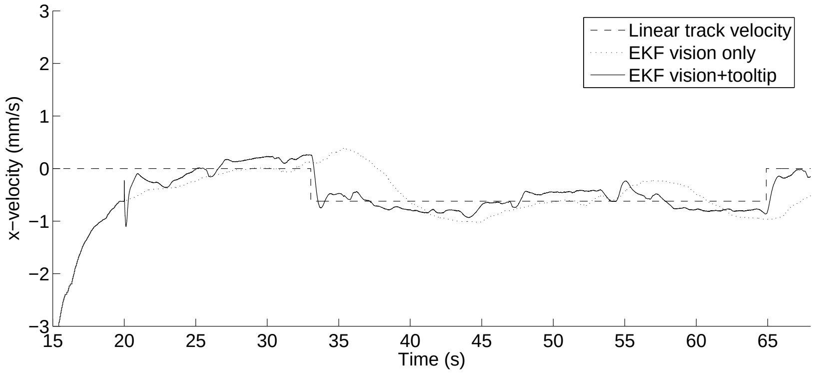


Figure 4.7: Estimated object x-coordinate velocity.

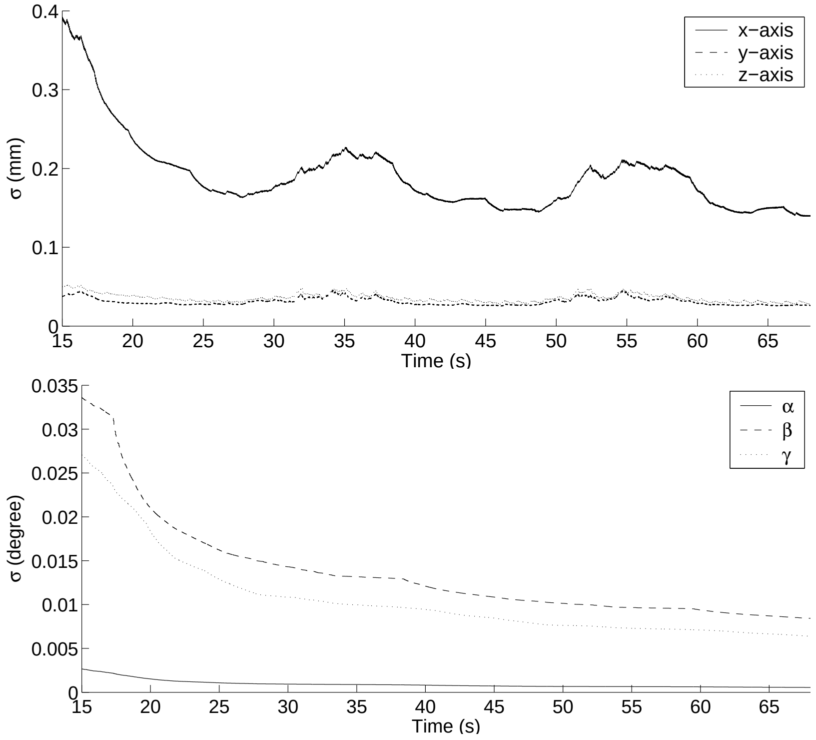


Figure 4.8: Estimated uncertainties without tooltip measurement.

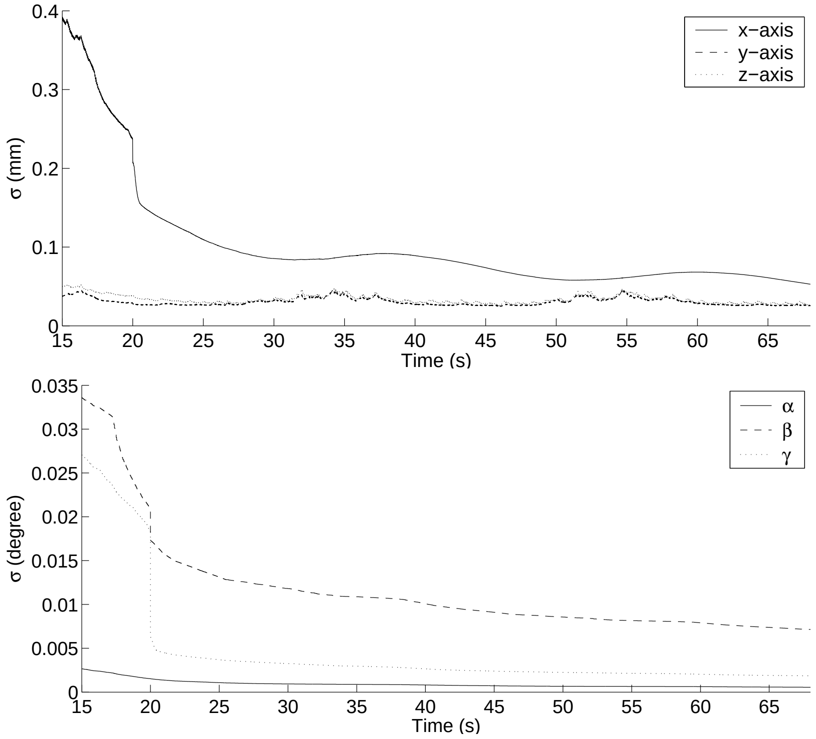


Figure 4.9: Estimated uncertainties with tooltip measurement.

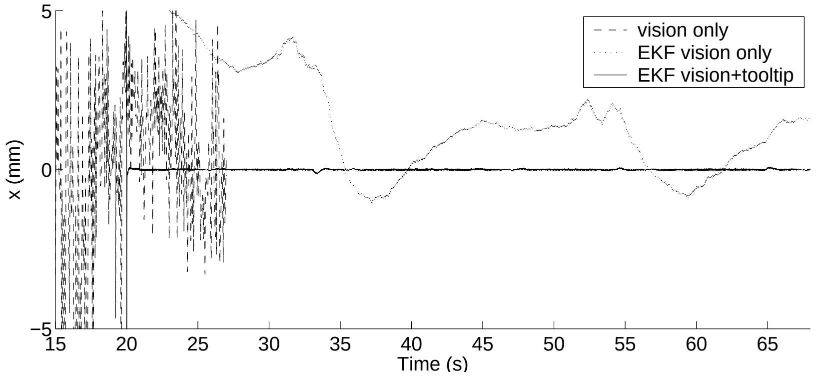


Figure 4.10: Estimated distance to the object plane.

4.2 Plane equation as a state vector for the EKF

In the case of structured light sensor, the full pose of the object cannot be determined from the visual measurement. Instead, the surface of the object can be assumed piecewise planar and the plane equation can be estimated. In [4] the visual measurements of the plane parameters are fused over time by utilizing the EKF. Tooltip probing adds information to the estimate, increasing the accuracy further. Even though the measurement and prediction models are linear, there is a nonlinear normalization step modifying the state and covariance matrix. The EKF estimates the state of a system using a system model $\boldsymbol{F}$ describing the dynamics of the system:

$$\boldsymbol{x}(t+1) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{w}(t), \tag{4.9}$$

where $\boldsymbol{w}$ is the model error modeled as a zero-mean Gaussian random variable. In addition, a measurement model $\boldsymbol{H}\boldsymbol{x}(t)$ is used to link the internal state to measurable quantities $\boldsymbol{y}$ by

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) + \boldsymbol{v}(t), \tag{4.10}$$

where $\boldsymbol{v}$ is the Gaussian measurement error. Uncertainty is modeled with covariance matrices so that $\boldsymbol{P} = Cov(\boldsymbol{x})$, $\boldsymbol{Q} = Cov(\boldsymbol{w})$, and $\boldsymbol{S} = Cov(\boldsymbol{v})$.

The target object can be modeled as two intersecting surfaces, both having a constant but unknown orientation and position in space. The parameters of the surfaces are affected by Gaussian noise with zero mean. The two planes are independent from each other, and thus two separate extended Kalman filters can be run instead of one, to reduce the complexity. The 4-dimensional state vector $\boldsymbol{x}$ contains the parameters of a plane

$$\boldsymbol{x} = \begin{pmatrix} a & b & c & d \end{pmatrix}^T . \tag{4.11}$$

The plane parameters are given with respect to the world coordinate system. The parameters are redundant, as a plane can be defined using only three parameters. The surface normal of the plane could be presented in polar coordinates instead of a normal vector. Another possibility is to fix one parameter, let us say the distance of the plane to the origin d, to be constant. However, such mappings are nonlinear and can have singularities. Instead, the plane is presented in general equation form $ax+by+cz+d=0$, and the parameters are normalized so that the normal of the plane $(a,b,c)^T$ is normalized to unit length after each update step of the EKF. It should be noted that when the plane parameters are normalized, also the covariance $\boldsymbol{P}$ of the plane parameters needs to be updated. The covariance can be approximated by $\boldsymbol{N}\boldsymbol{P}\boldsymbol{N}^T$, where $\boldsymbol{N}$ is the Jacobian of the normalization function. The visual measurement is described by a 4 × 4 identity matrix $\boldsymbol{H}_V = \boldsymbol{I}$.

In order to fuse the measurements in the EKF, the uncertainty of each individual measurement must be estimated. By assuming the uncertainty of the visual measurement in pixel coordinates to be zero mean Gaussian and projecting it into the uncertainty of plane parameters in world coordinates, the measurement can be incorporated in the EKF. The linear transformation $\boldsymbol{A}$ described in (2.4), estimates how much the plane parameters change when there is a small error in the image plane coordinates. If the statistics for the image plane parameters are known, also the statistics for the plane parameters can be determined. The covariance matrix can be defined as

$$\boldsymbol{S}_H = \boldsymbol{A}\sigma_i^2\boldsymbol{A}^T, \tag{4.12}$$

where $\boldsymbol{A}$ is a linear mapping from the image plane parameters to the plane parameters, and σ_i^2 defines the uncertainties for the image parameters. The visual measurement gives the parameters of a plane accompanied with corresponding covariance matrix for the uncertainties of the parameters.

Next, the measurement model for the tooltip measurement is defined. The surface of the object is assumed planar. Thus, the distance d_t from the tooltip point $P_t = (x_t, y_t, z_t)$ to the plane is

$$d_t = \frac{|ax_t + by_t + cz_t + d|}{\sqrt{a^2+b^2+c^2}}, \tag{4.13}$$

where a,b,c and d are the plane parameters. Positive distances are defined above the surface and negative distances inside the object. As the normal vector $\boldsymbol{n}$ is normalized to unit length in the EKF, the distance equation reduces to

$$d_t = ax_t + by_t + cz_t + d. \tag{4.14}$$

When the tooltip is in contact with the object, the distance from the object to the tooltip is assumed zero. For a rigid non-deformable object this assumption is reasonable. The position of the tooltip P_t can be obtained in the world coordinates, as the dimensions of the tool are measurable and can be assumed constant if the stiffness of the tool is high enough.

As the tooltip measurement (4.14) is linear, the measurement model is described by

$$\boldsymbol{H}_F = \begin{pmatrix} x_t & y_t & z_t & 1 \end{pmatrix}. \tag{4.15}$$

The visual measurements and the tooltip measurement can be now combined into one model

$$\boldsymbol{y}(t) = \boldsymbol{H}\boldsymbol{x}(t) = \begin{pmatrix} \boldsymbol{H_V} \\ \boldsymbol{H_F} \end{pmatrix} \boldsymbol{x}(t) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ x_t & y_t & z_t & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}, \tag{4.16}$$

and the covariance then becomes

$$\boldsymbol{S} = \begin{pmatrix} \boldsymbol{S}_H & \boldsymbol{0} \\ \boldsymbol{0} & \sigma_t^2 \end{pmatrix}, \tag{4.17}$$

where σ_t^2 is the variance of tooltip distance measurement modelling the deformability of the object and the stiffness of the tool.

In case the tool or the object is highly flexible, the deformability can be taken into account by modeling the contact as a spring system. Instead of assuming the distance from the object to the tooltip to be zero, the distance is estimated using a normal contact force $f_n = |\boldsymbol{f}|$ and a spring constant k by Hooke's law $d_t = -kf_n$.

The normal contact forces $\boldsymbol{f}_1$ and $\boldsymbol{f}_2$ for an object consisting of two intersecting planes is shown in Fig. 4.11. $\boldsymbol{n}_1$ is the surface normal of the first plane and $\boldsymbol{n}_2$ the surface normal of the second plane. The intersection line of the planes is the cross product of the surface normals $\boldsymbol{n}_3 = \boldsymbol{n}_1 \times \boldsymbol{n}_2$, as it is perpendicular to both surface normals. $\hat{\boldsymbol{n}}_1 = \boldsymbol{n}_1 \times \boldsymbol{n}_3$ and $\hat{\boldsymbol{n}}_2 = \boldsymbol{n}_2 \times \boldsymbol{n}_3$ can be defined as vectors along the planes that are perpendicular to the intersection line. By projecting the contact force to vectors $\hat{\boldsymbol{n}}_1$ and $\hat{\boldsymbol{n}}_2$, information on the contact type can be obtained. The projected forces are $\hat{\boldsymbol{f}}_1 = P(\hat{\boldsymbol{n}}_1)\boldsymbol{f}$ and $\hat{\boldsymbol{f}}_2 = P(\hat{\boldsymbol{n}}_2)\boldsymbol{f}$, where

$$P(\boldsymbol{n}) = \frac{1}{a^2 + b^2 + c^2} \begin{pmatrix} a^2 & ab & ac \\ ab & b^2 & bc \\ ac & bc & c^2 \end{pmatrix} \tag{4.18}$$

and $\boldsymbol{n} = \begin{pmatrix} a & b & c \end{pmatrix}$. As the vector $\hat{\boldsymbol{n}}_1$ is parallel to the first plane, forces along the vector cannot be caused by surface normal forces of the first plane. Therefore, they must be caused by contact to the second plane. The same applies for the second plane: forces along the vector $\hat{\boldsymbol{n}}_2$ indicate contact to the first plane. It should be noted that when the tool is sliding on the object, friction forces may produce forces parallel to the projected forces. However, if the friction forces are smaller than the desired contact forces, the force threshold can be such that the contact information is only used when the force is larger than the maximum friction forces. An algorithm for estimating the position deformation is presented in Alg. 10.

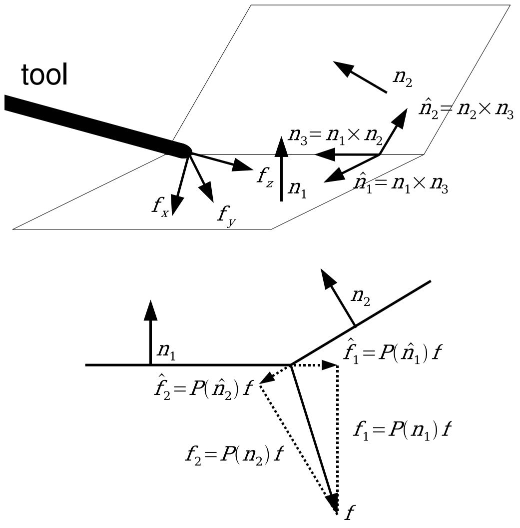


Figure 4.11: Forces during contact to two surfaces.

4.3 Summary and discussion

Sensor-based control allows manipulation in dynamic environments with uncertainties. Although vision is a sensor modality with high information content, it also suffers from

Algorithm 9 Algorithm for using forces to estimate position.

```
if $\hat{\boldsymbol{f}}_1 > threshold$ then
    project the contact force $\boldsymbol{f}$ along the surface normal $\boldsymbol{n}_2$
    store the projected contact force into vector $\boldsymbol{f}_2$
    estimate the deformation of the second surface by Hooke's law $d_t = -k|\boldsymbol{f}_2|$
end if
if $\hat{\boldsymbol{f}}_2 > threshold$ then
    project the contact force $\boldsymbol{f}$ along the surface normal $\boldsymbol{n}_1$
    store the projected contact force into vector $\boldsymbol{f}_1$
    estimate the deformation of the first surface by Hooke's law $d_t = -k|\boldsymbol{f}_1|$
end if
```

limitations, such as low sample rate, high sensor delay and uncertain measurements. Some Cartesian degrees of freedom are also less accurate for visual measurements due to limited perspective effects. A method for combining force and vision to estimate a pose of a target in an EKF has been proposed. A relative pose from an end-effector mounted camera to a moving object is measured using vision. High frequency positional information of the tooltip motion on the surface of the object and visual measurement are fused in the EKF to give accurate high rate pose estimates, as well as translational and angular velocities. The sensor delay of the visual measurement is taken into account explicitly. The positional measurements are delayed in the EKF so that they are synchronized with the visual measurement. An external prediction model then estimates the pose of the object for the present time. Inaccuracies in the visual measurement are compensated with accurate measurements from tooltip probing. When the tooltip is in contact with the object surface, the position of the tooltip is calculated using the cylinder lengths of a parallel manipulator or the joint sensors of a serial manipulator. The method can be used also when the tool is sliding on the object surface as it is independent of the friction forces.

In [101] a serial robot was used to probe the pose of an object as well as contact parameters. However, the proposed approach used vision only to estimate the pose of the tool and not the pose of the object. The approach proposed in this thesis uses vision to estimate the pose of the object, and the position of the tooltip, as well as the pose of the camera are obtained from the joint sensors a robotic manipulator. An algorithm proposed in [60] combines vision with force and joint angle sensors. A camera fixed to the world frame, as well as a wrist force sensor and joint sensors of an 6-DOF industrial robot are fused in an EKF. While the approach in [60] takes advantage of the force sensor measurements directly in the pose estimate as well as the proprioceptive information from the joint sensors, they assume frictionless point contact, making it impossible to use sensor fusion when the tooltip is moving on a physical surface.

The proposed approach was validated by experiments with a 5-DOF parallel hydraulic manipulator. The experiments verified that the method improves the estimation accuracy of the pose of a moving target significantly by combining sensors with different sensor modalities. The method combines data from very different sensors into one model, providing a novel and accurate method for 6-DOF pose tracking.

CHAPTER V

Force and vision control

In this chapter, control methods combining force and vision sensors are presented. Force and vision sensors measure fundamentally different physical phenomena. Thus, the use of standard sensor fusion techniques for combining the measurements has significant problems, as common data representation for force and vision sensors can not be found easily. For this reason, traditional control approaches based on a single type of sensor input can not be used.

There are three basic strategies for combining force and vision: switched control, hybrid control, and shared control [74, 73]. In switched control, force and vision are used separately, at different time instants. Switched control, which switches between image-based visual servoing and force control has been presented in [72, 110]. In hybrid control, each Cartesian degree of freedom is controlled by either force or vision sensors [16, 76, 79]. Hybrid control can be used to follow unknown surfaces [104] and track arbitrary contours [23]. Hybrid control methods taking into account the uncertainties in kinematics, dynamics and the camera model also exist [108]. Vision can also be used in a shared control scheme where the information from force and vision sensors is used simultaneously for controlling the same axes [12].

One of the problematic stages of force controlled machining is when the tool comes into contact with the workpiece. Even if force control is used, the inertia of the robot causes a momentary force peak on the instant of impact. The impact can cause damage to the tool or the workpiece, especially when their stiffness is high. The force overshoot can be reduced by decreasing the velocity of the end-effector before the contact occurs. The use of visual information has been proposed as a solution to reduce the impact forces [72, 110]. Vision can be used to reduce the force overshoot by reducing the velocity as the point of contact approaches. If the pose of the end-effector is known relative to the target, an efficient velocity profile for approaching the target can be determined.

The diverseness of both the applications and the machinery for the integration of force and vision is extensive. Fusion of force and vision has been utilized in both assembly tasks with parallel robots [8] and disassembly tasks with serial robots [81]. Vision can be sometimes used to predict forces before they can be detected with force sensors. For example, automatic deburring can greatly benefit from visual information if vision is used to detect the size of the burr, and machining speed is adjusted accordingly [106]. A grinding tool can be moved at higher speed when the size of the burr is small and the speed can decreased when the vision system detects a larger burr size. A wide variety of automatic robot deburring tasks, such as verifying deburring performance [19] and determining the burr size and location [34] benefit from sensor integration. Also locating the workpiece and ensuring proper tool contact [88] and grinding tasks [50] can use sensor fusion. One interesting area of applications is microrobotics, where force and vision sensors have been utilized in both mobile microrobots [22] and microassembly where a microscope is coupled with a haptic interface [45] or an optical beam deflection force sensor [110] to provide methods for example for part assembly and alignment [36]. Also high speed robotics can benefit from vision and force control, for example to catch falling objects [71].

In this chapter, three different control strategies are proposed: switched control, hybrid control and shared control. The switched control experiments were done with a hydraulic 1-DOF servo bench and 5-DOF parallel hydraulic manipulator. For hybrid and shared control experiments, a 6-DOF serial manipulator was used to track a contour of two planar surfaces. Planar contour tracking fusing force and vision sensors has been proposed before for a 1-DOF [70] as well as for a 6-DOF [76] case. Pomares and Torres [82] combined image-based visual servoing with force sensing to detect discontinuities in contact surface, where only force sensing is used for recognizing the orientation of the planar surface. Force control can also be used to probe the shape of a workpiece. Mason and Salisbury [68] have solved the unknown geometry of the object by force probing. The surface normal of the object can be estimated if the contact point is known and the contact forces are measurable. However, the method is only applicable to frictionless point contact. When the tooltip slides on the object, unknown friction forces affect the force measurements along the axis of the motion, but the other axes perpendicular to the motion can still give information about the shape of the object. Another approach estimates the geometric parameters of an object by utilizing EKF [87]. The framework also assumes frictionless contact, but in discussion it is proposed that the friction coefficient could be incorporated in the EKF estimate. To the author's knowledge, using contact information to compensate for the uncertainty of vision in estimation has not been proposed before.

5.1 Switched controller

In this section, a switched controller is proposed which applies a smooth transition method from vision-based velocity control to force control, allowing minimal force overshoot at the time of impact while also providing fast approach.

In switched control, there is a transition stage from visual to force control. Nelson et al. [72, 110] have considered the use of vision as a means to decrease the impact forces on contact. In [110], a switching control scheme which uses image-based visual servoing for visual control is proposed. In contrast to directly controlling the robot by using image point features, visual features are used in this section to estimate the relative pose of the end-effector with respect to the target.

Figure 5.1 presents an overview of the proposed switched controller, where switching is used to select either vision- or force-based control. Three sensors are used: a force sensor, an incremental position encoder, and a vision system. The force sensor can be fitted directly to the end-effector to measure the contact force directly. The vision-based control loop consists of two nested controllers, the outer controller setting the desired velocity for the inner velocity controller. The use of an incremental position encoder for velocity measurements makes it possible to have high frequency in the inner controller even while the sampling frequency of vision is low. Next, each of the subparts of the system are explained in more detail.

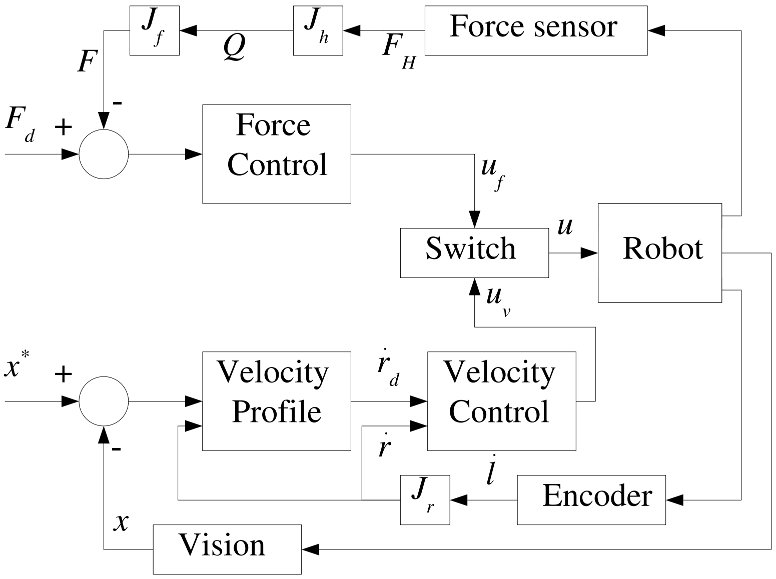


Figure 5.1: Controller.

5.1.1 Desired velocity profile

Approaching the object should be as fast as possible, and at the same time the impact forces should be minimized. To achieve a smooth contact, the velocity of the end-effector must be low when the contact occurs. Approaching the object can be done by using high speed, and when the end-effector is near the object, the speed can be reduced for a smooth contact. Usually there is always a maximum acceleration the system will endure. There can also be constraints in the maximum velocity of the end-effector.

In an optimal approach, the acceleration stays constant at the upper limit of the acceleration the system can endure. Vision based velocity control calculates an optimal velocity based on the distance to the object. The desired velocity is as high as possible so that it is still possible to decelerate the end effector to a safe contact velocity.

The velocity profile of the vision-based velocity control is linear in the time-velocity coordinate system, as can be seen in Fig. 5.2. Vision is used to determine the distance to the target, so the desired velocity must be known in position-velocity coordinates.

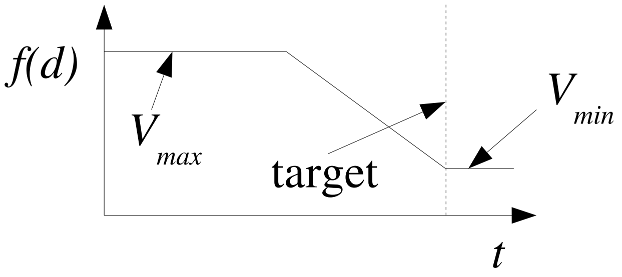


Figure 5.2: Velocity profile.

To attain constant deceleration, the desired velocity $f(d)$ during the deceleration is

$$f(d) = \sqrt{V_{min}^2 + 2ad}, \tag{5.1}$$

where V_{min} is the desired speed of contact for a low gain force control, a is the allowed deceleration, and d is the distance to the point of contact. To limit the speed outside the zone of constant deceleration, the desired velocity can be written as

$$f(d) = \min\left(V_{max}, \max\left(\sqrt{V_{min}^2 + 2ad}, V_{min}\right)\right). \tag{5.2}$$

The direction vector of the velocity $\boldsymbol{v}$ can be written as

$$\boldsymbol{v} = \frac{\boldsymbol{x} - \boldsymbol{x}^*}{\|\boldsymbol{x} - \boldsymbol{x}^*\|}, \tag{5.3}$$

where $\boldsymbol{x}$ is the measured end-effector position and $\boldsymbol{x}^*$ the desired position in Cartesian space. Noting that $d = \|\boldsymbol{x} - \boldsymbol{x}^*\|$, the desired velocity of the end-effector $\dot{\boldsymbol{r}}_d$ is then

$$\dot{\boldsymbol{r}}_d = f(d)\boldsymbol{v} = f(\|\boldsymbol{x} - \boldsymbol{x}^*\|)\frac{\boldsymbol{x} - \boldsymbol{x}^*}{\|\boldsymbol{x} - \boldsymbol{x}^*\|}. \tag{5.4}$$

Thus, the desired motion is a straight line path with the speed controlled by the distance to the target.

The latency and low frame-rate of the visual system cause error to the estimated position. The incremental position encoder gives high frequency proprioceptive information of the end-effector motion. By fusing visual and encoder information, it is possible to compensate for the latency of the vision system and interpolate the position of the end-effector. Encoders can only be used for calculating relative positions as the location of the objects is not known. The encoder position at the time instance when the position was measured by the visual sensor is stored. The difference between the current encoder position and the stored position is calculated and added to the position measured by the visual system. The compensated position of the end-effector is as follows:

$$\boldsymbol{x} = \boldsymbol{x}_{vis}(t_0) + \int_{t_0}^{t} \boldsymbol{v}_{enc}, \tag{5.5}$$

where $\boldsymbol{x}_{vis}(t_0)$ is the end-effector position at the time instance when the image was taken and $\boldsymbol{v}_{enc}$ the velocity in Cartesian coordinates given by the encoders.

The focus in the experiments was on hydraulic manipulators, but the proposed controller is also applicable to electrically driven robots equipped with a force sensor.

5.1.2 Velocity control

A position is achieved by velocity control. When operating in the velocity control mode, the controller drives an actuator at a velocity rate specified by a command. Once the motion starts, the actuator moves indefinitely at the required speed until it is commanded to change speed or stop.

The actuator space velocity $\dot{\boldsymbol{l}}$ is measured by encoders, and the end-effector velocity in Cartesian space is

$$\dot{\boldsymbol{r}} = \boldsymbol{J}_r \dot{\boldsymbol{l}}, \tag{5.6}$$

where $\boldsymbol{J}_r$ is the robot Jacobian. The velocity during the approach phase is controlled using a proportional velocity controller

$$\boldsymbol{u_v} = \boldsymbol{K}_v \left(\dot{\boldsymbol{r}}_d - \dot{\boldsymbol{r}} \right), \tag{5.7}$$

where $\boldsymbol{K}_v$ is a suitable positive definite matrix gain and $\dot{\boldsymbol{r}}_{\boldsymbol{d}}$ the desired velocity.

The controller needs at least two poles in the origin in order to follow a ramp signal with a zero steady-state error [33]. A standard proportional-integral (PI)-controller has only one pole in the origin, and adding a second integrative term could make the system unstable. However, if the deceleration rate is relatively small, there will be no significant deviation from the desired velocity even with a proportional (P)-controller during the ramp phase.

5.1.3 Force control

The aim of the force controller is to attain and maintain a constant desired force between the robot end-effector and a target. Let $\boldsymbol{F}_d$ denote the desired force vector. The force controller can now be written

$$\boldsymbol{u}_f = \boldsymbol{K}_p \left(\boldsymbol{F_d} - \boldsymbol{F} \right) + \boldsymbol{K}_i \int_{0}^{t} \left(\boldsymbol{F_d} - \boldsymbol{F} \right), \tag{5.8}$$

where $\boldsymbol{K}_p$ and $\boldsymbol{K}_i$ are suitable positive definite matrix gains. The proportional-integral control scheme converges to a steady state with zero force error, provided that the control gains are properly chosen to ensure stability of the closed loop system.

5.1.4 Switched control

The switched controller consists of a vision-based velocity controller and a force controller. Velocity control is used in approaching the object, and switched to force control when the object is reached. The transition from visual to force control is based on force thresholding. Vision is used if the force is under the threshold F_t and if the threshold is exceeded, force control is applied. Thus, the switched controller can be written as

$$\boldsymbol{u} = \begin{cases} \boldsymbol{u}_v & \text{if } \|\boldsymbol{F}\| \leq F_t \\ \boldsymbol{u}_f & \text{if } \|\boldsymbol{F}\| > F_t \end{cases} . \tag{5.9}$$

Inertial coupling of the end-effector mass can cause measurable forces. If the inertial forces are greater than the force threshold, the forces should be compensated. The Cartesian acceleration of the end-effector can be calculated from the encoder's positional information. An acceleration sensor can also be used if the encoder data is too noisy. If the Cartesian acceleration and the mass of the end-effector are known, the inertial forces can be estimated.

Because the velocity and force controllers are separate, they do not hinder each other's performance, even if the sampling rate of vision is low. That is, after the contact, only the force controller is used.

5.1.5 Experiments

Making contact with a workpiece is essentially a one-dimensional motion. For that reason, a test setup with a one-axis robot has been developed to analyze the controller proposed in Sec. 5.1 experimentally.

The experimental setup consists of three parts: a hydraulic actuator with a sled, a rigid target, and a vision system. The sled moves on a linear guideway, placed on an I-bar bolted to the floor.

A rubber damper is mounted to the actuator so that there is some flexibility when the actuator impacts the target. The diameter of the piston rod is 25 mm. An LT5 A1 force sensor is installed directly to the hydraulic cylinder for the force measurements. The rated capacity of the hydraulic cylinder is 2000 kg. The displacement of the sled is measured by a linear position encoder. The setup is shown in Fig. 5.3.

The controller operates on a dSpace DS1103 controller board connected to a PC, which is used to build the controller with Simulink. The force sensor and the position encoder are directly connected to the controller. The vision system operates on a separate PC, connected to the controller with a serial link. Different velocity profiles were compared with respect to their rate of convergence and the force overshoot at impact.

In the first experiment, the magnitude of the force overshoot was examined at different constant approach velocities. In this experiment, visual information was not used to determine the location of the target, and there was no deceleration before contact. The test bench was started from the velocity control, and switched to force control after touching the object. The desired contact force was set to $F_d = 200N$, and the threshold

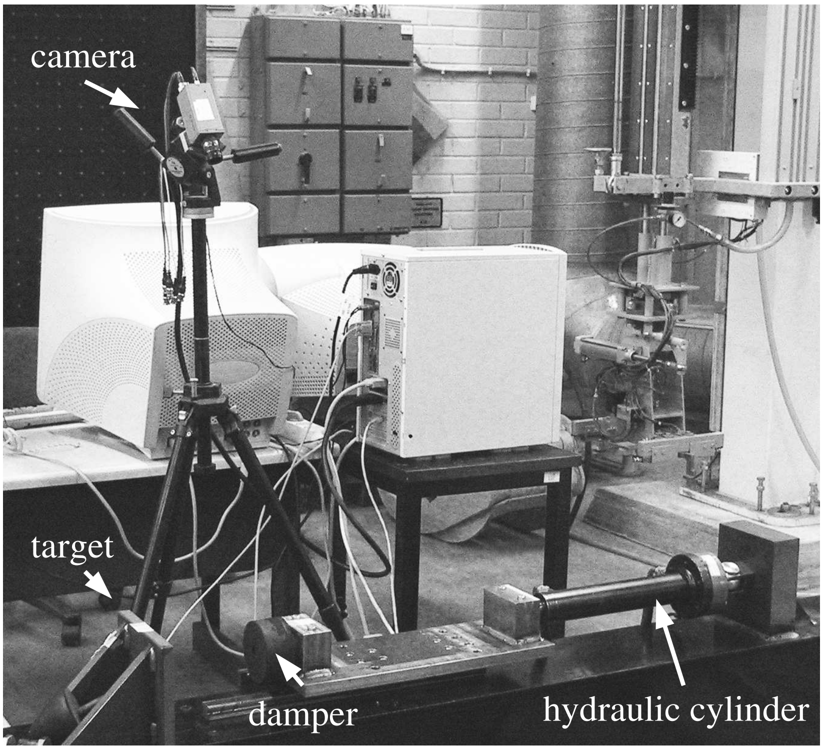


Figure 5.3: Test setup.

force for switching to $F_t = 50N$. Three different velocities and their corresponding contact forces are shown in Figs. 5.4 and 5.5, respectively.

Fig. 5.5 it can be seen that the force overshoot increases when the velocity is increased. This is due to the inertial effects of the end-effector. The force overshoot can be reduced by decreasing the contact velocity. However, the time to contact increases if the approach velocity is decreased.

In the second experiment the proposed controller was used. The distance to the object was determined by vision only, without the incremental encoder. The deceleration rate was set to $64mm/s^2$, and the contact velocity to the smallest velocity of the first experiment. The resulting velocity and force are shown in Fig. 5.6.

In Fig. 5.6 it can be seen that the contact force overshoot is negligible, but the time to contact is smaller compared to the constant velocity approach with the same overshoot. The jagged velocity profile in Fig. 5.6 is caused by the low frame-rate of the visual system and can be improved by incorporating the encoder measurements.

In the third experiment, integration of the encoder and visual measurements was used, as proposed in Sec. 5.1.1. Figure 5.7 shows the measured velocity and force. The figure shows that the deceleration is now smooth. The contact velocity and therefore the contact force are the same as in the previous experiment.

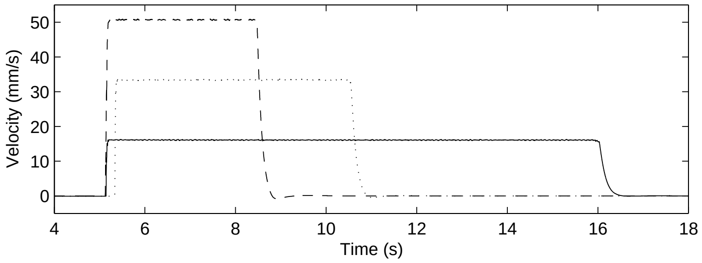


Figure 5.4: Constant velocity approach and force control.

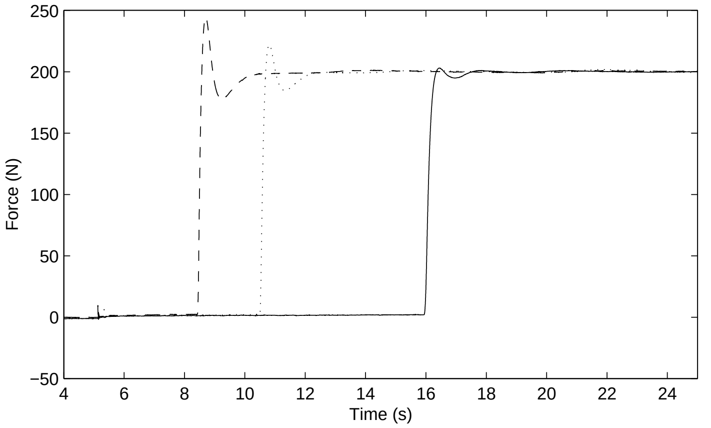


Figure 5.5: Contact forces with constant velocity approach.

In the fourth experiment, the proposed approach was compared to a P-controller for the position during the approach phase. This experiment is comparable to the approach proposed by Zhou et al.[110], but instead of visual features, the control is done in Cartesian coordinates. However, the visual measurements are approximately linearly dependent on the Cartesian measurements, and therefore the results are comparable. The gain of the controller was optimized so that the maximum and contact speeds were not exceeded. The velocity and force of the proportional position control is shown in Fig. 5.8. It can be seen that the time to contact is higher than with the proposed controller, and therefore the proposed approach gives faster convergence when the contact force needs to be limited. In addition, as the desired contact speed is a simple physical quantity, it is often easier to set, compared to the parameters of the proportional controller.

Experiments were also conducted with a moving target object [62]. In Fig. 5.9 the object moves away from the end-effector. As the velocity of the end-effector is higher than the velocity of the object, the end-effector eventually reaches the object and impact occurs. The experiment shows that the end-effector and the object have the same velocity after the impact and that the contact force is stable. When the object moves towards the end-effector, shown in Fig. 5.10, a force peak occurs during impact.

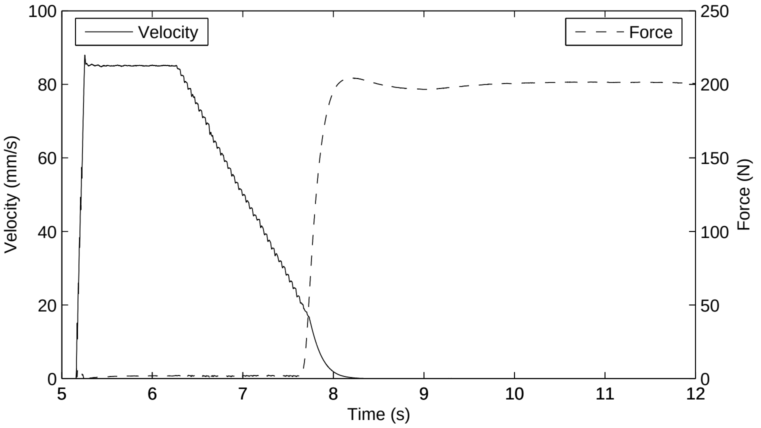


Figure 5.6: Measured velocity and force using vision and force sensors.

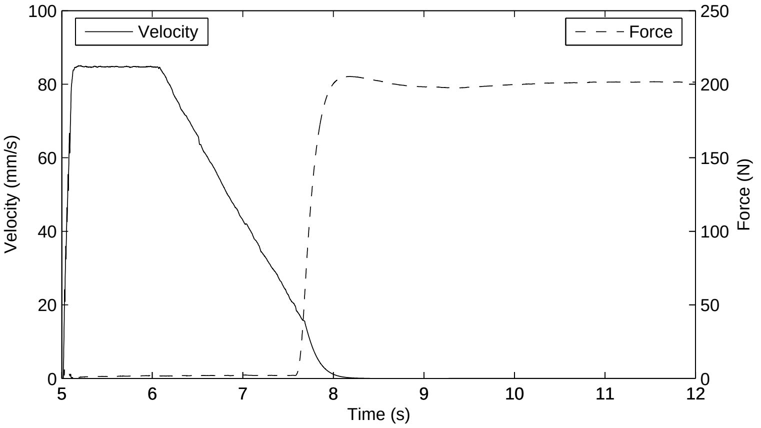


Figure 5.7: Measured velocity and force using vision, encoder and force sensors.

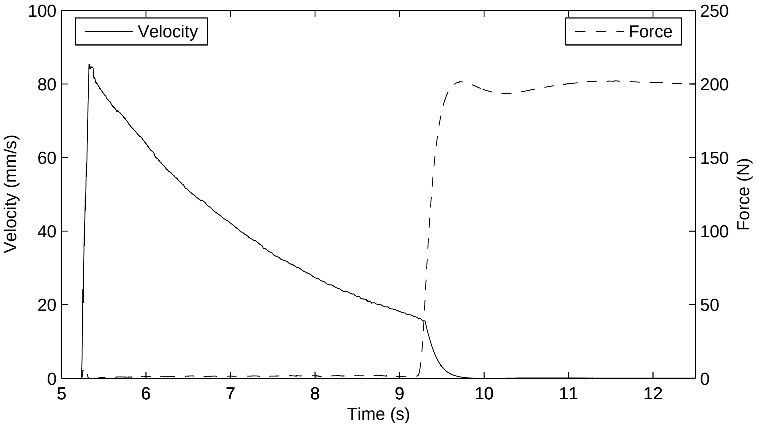
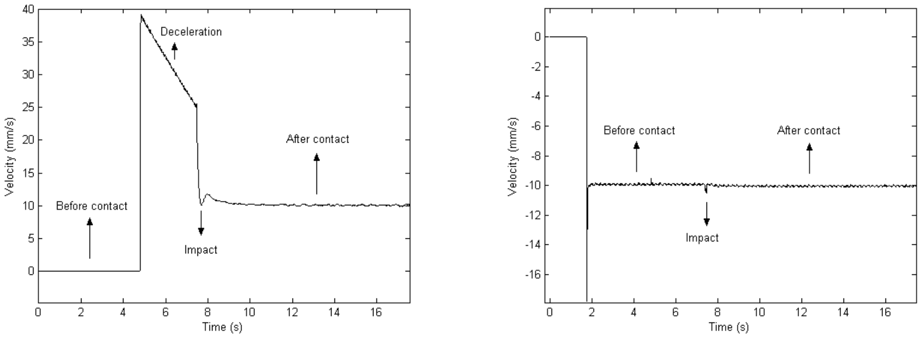


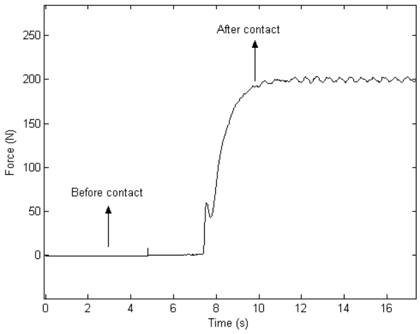
Figure 5.8: Position-based velocity profile and force control.

Impact experiments with 5-DOF parallel hydraulic manipulator shown in Fig. 3.2 verified that force peak can be reduced by decreasing the impact velocity [63]. In Fig. 5.11 it is visible that higher impact velocity causes larger force peak. Different materials also have different force control properties, and even when using the same velocity the force peak can be different, as shown in Fig. 5.12.



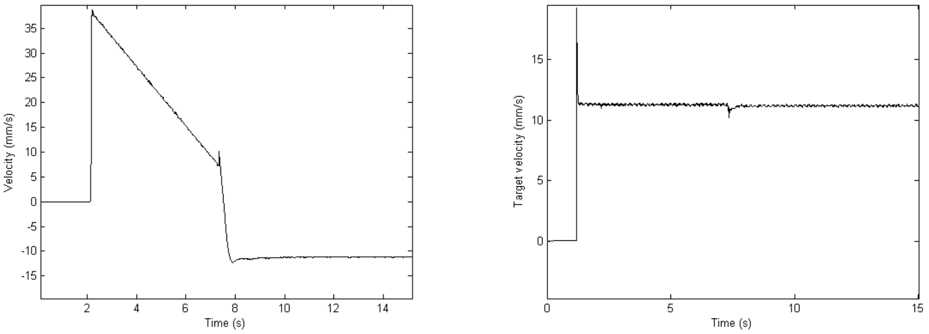
(a) End-effector velocity.

(b) Object velocity.



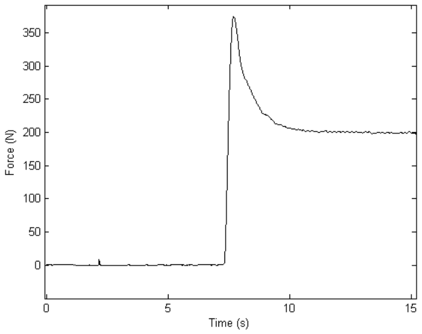
(c) Contact forces.

Figure 5.9: Integrated force and velocity control with vision with target movement away from the end-effector. [62]



(a) End-effector velocity.

(b) Object velocity.



(c) Contact forces.

Figure 5.10: Integrated force and velocity control with vision with target movement towards the end-effector. [62]

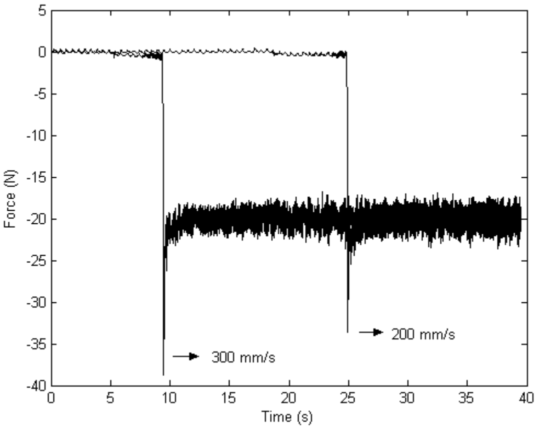


Figure 5.11: Impact forces with different velocities. [63]

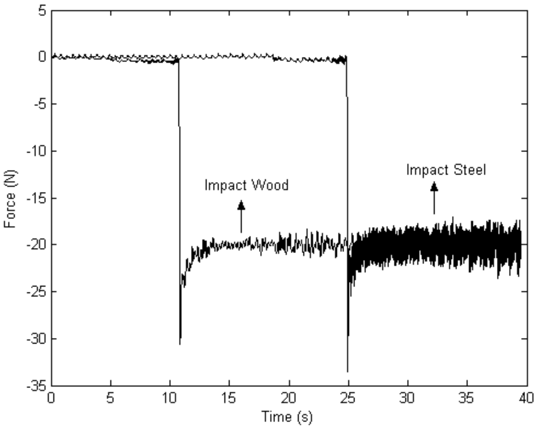


Figure 5.12: Impact forces to different materials. [63]

5.2 Hybrid control

In this section, an online trajectory following task which utilizes laser projection vision and force sensors is proposed. The trajectory to be tracked is assumed to lay on a piecewise planar surface bounded by another plane. The laser striping system projects two laser lines perpendicular to each other on both planes. The visual system estimates the location and orientation of the two planes. This information is used to control the orientation of the tool. While the tooltip is in contact with the object, and following the trajectory, the robot joint sensors provide additional information, giving one point on the trajectory. The force sensor also measures contact forces to both planes and the friction force along the trajectory. Three different force controllers are set up to control three axes: an axis along the surface normal, an axis along the trajectory, and an axis perpendicular to the previous axes. These three axes are controlled with proportional-controllers to have constant contact forces. The vision is essential also for force control, as the orientation of the tool affects the directions of the three force controlled axes.

Measurements from the vision and robot joint sensors are fused in the EKF. Two separate filters are set up to estimate each independent plane measured by the visual sensor. The state vector of each EKF contains the parameters of one plane. The proposed EKF method is compared to an EKF method which uses only vision to estimate the state. Also a method using visual measurement directly to set the controllable axes is compared to the previous methods.

5.2.1 Hybrid controller

The control system consists of an EKF-based controller, which determines the correct orientation for the tool, and a force controller, which maintains desired the contact forces. All the controllers are proportional-velocity controllers, which feed the desired linear and angular velocities to the robot controller. The control loop is run with the maximum 140Hz rate supported by the robot controller.

The set point for the tool orientation controller is estimated by the EKF-based on the visual and the tooltip measurements. The orientation controller is therefore a shared controller utilizing both the visual and proprioceptive information from the joint sensors. The force controller, on the other hand, controls three translational axes in the task-frame coordinates. The forces are measured by a wrist mounted 6-DOF force sensor. Each axis is controlled by a separate force controller and with axes perpendicular to each other. The first axis is perpendicular to the object surface, the second axis is along the trajectory, and the third axis is perpendicular to the first two axes. The controller outputs the desired velocities in the end-effector coordinate system, which are then fed to the robot controller. Each axis can be set to have a different desired force, and also the maximum velocity for each axis can be limited. Usually only the second axis, controlling the end-effector velocity along the trajectory, will be limited by the maximum velocity restriction.

5.2.2 Experiments

Experiments were conducted to show that the EKF estimate reduces the visual measurement noise considerably and allows higher gain to be used during control. The system consists of a laser striping visual system and a 6-DOF force transducer integrated into a 6-axis serial manipulator, shown in Fig. 5.13.

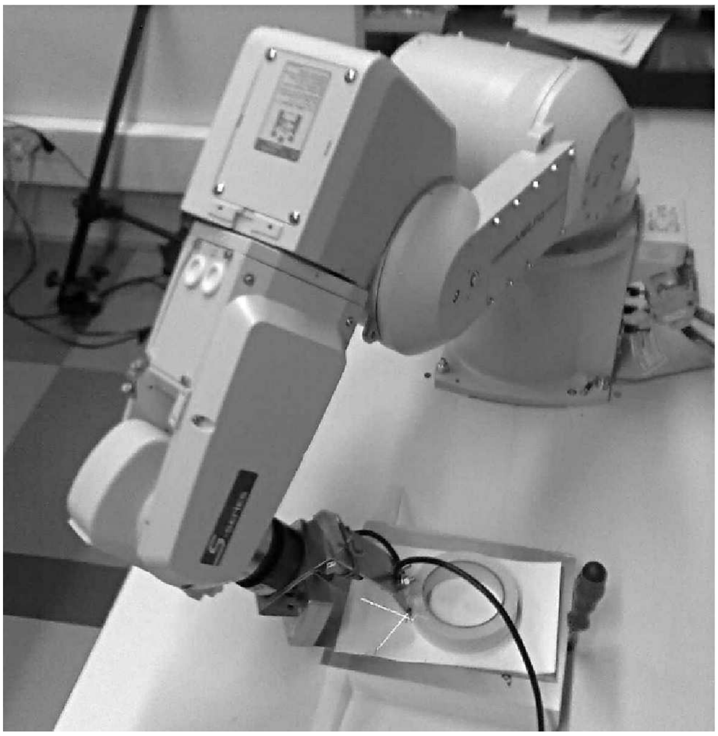


Figure 5.13: 6-DOF serial manipulator.

In the experiments, three different estimators were used to control the orientation of the end-effector: a simple unfiltered visual measurement, EKF-based estimate utilizing only visual measurement, and an EKF estimate fusing visual and tooltip measurement. Then, a force controller was set up to control three axes: z-axis perpendicular to the target object, x-axis along the trajectory, and y-axis perpendicular to the first two axes. First the estimation methods were compared in hybrid control under a relatively low control gain. Fig. 5.17 shows the trajectories under the different estimation methods. In the figure it can be seen that the estimation method fusing visual and tooltip measurement gives the smoothest trajectory, and the vision-only estimate oscillates the most. However, the scale of the plot should be noted. The oscillations using the a visual estimate are under 0.1mm in amplitude, which can be considered adequate for most tasks. In the contact forces there are no drastic differences between the different estimation methods. The contact forces under the pure visual estimate in Fig. 5.16 oscillate slightly more than when using the filtered estimates shown in Figs. 5.14 and 5.15.

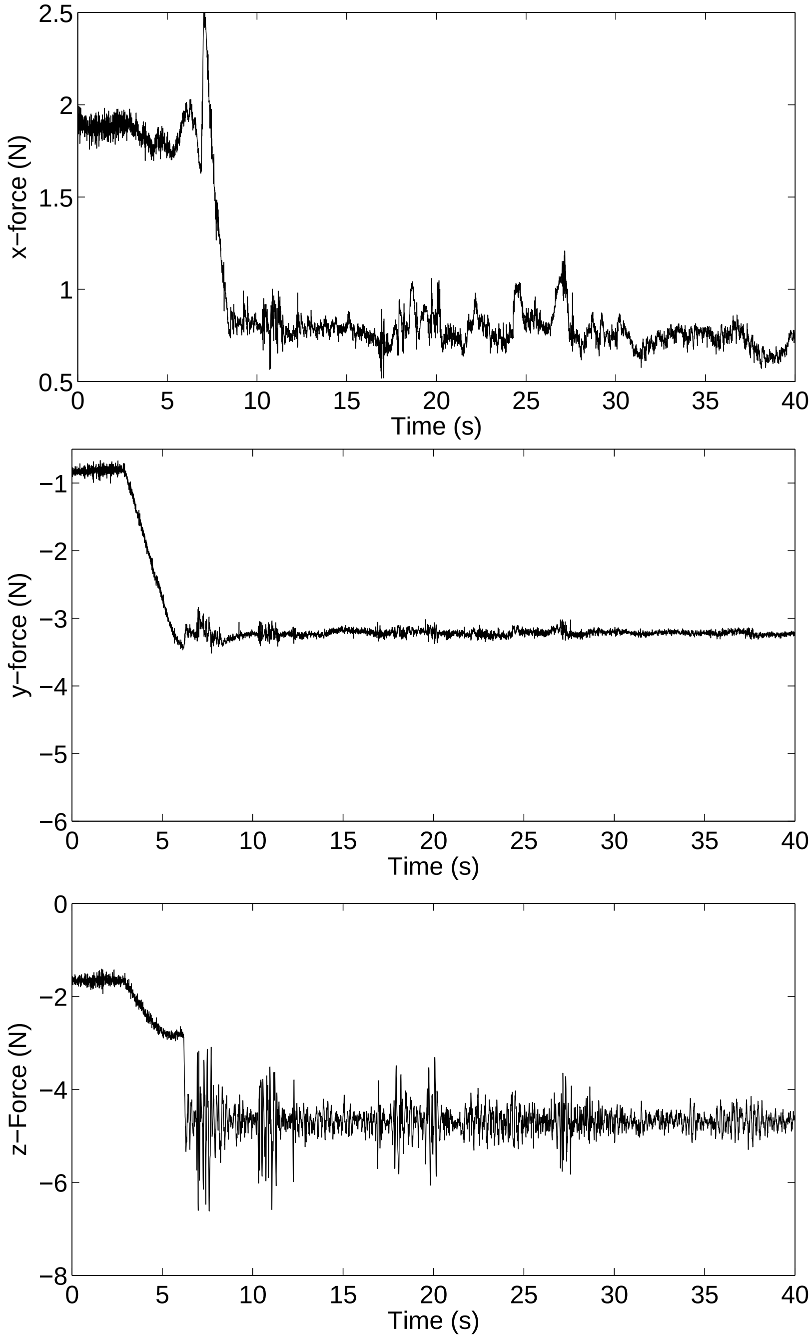


Figure 5.14: Contact forces during hybrid control with Kalman filter estimate using only vision.

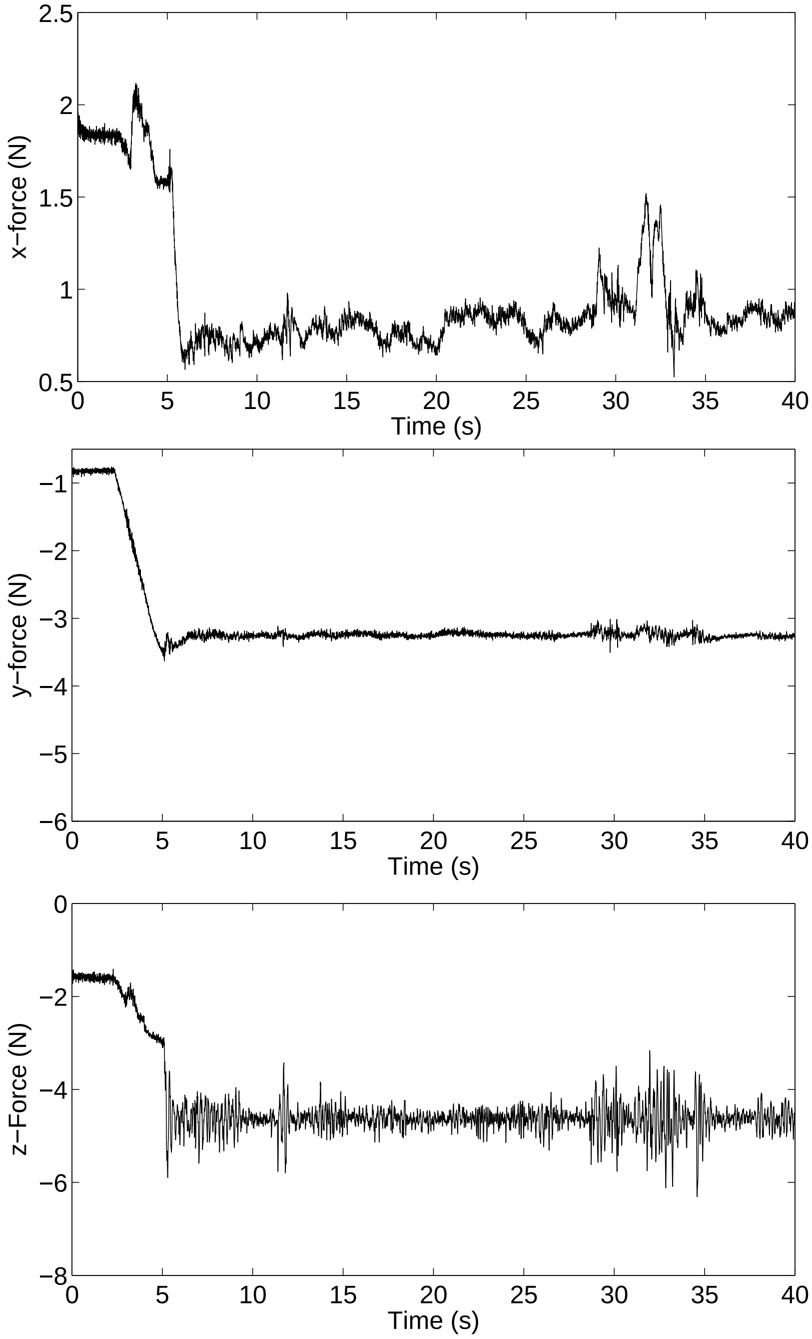


Figure 5.15: Contact forces during hybrid control with Kalman filter estimate using vision and tooltip measurement.

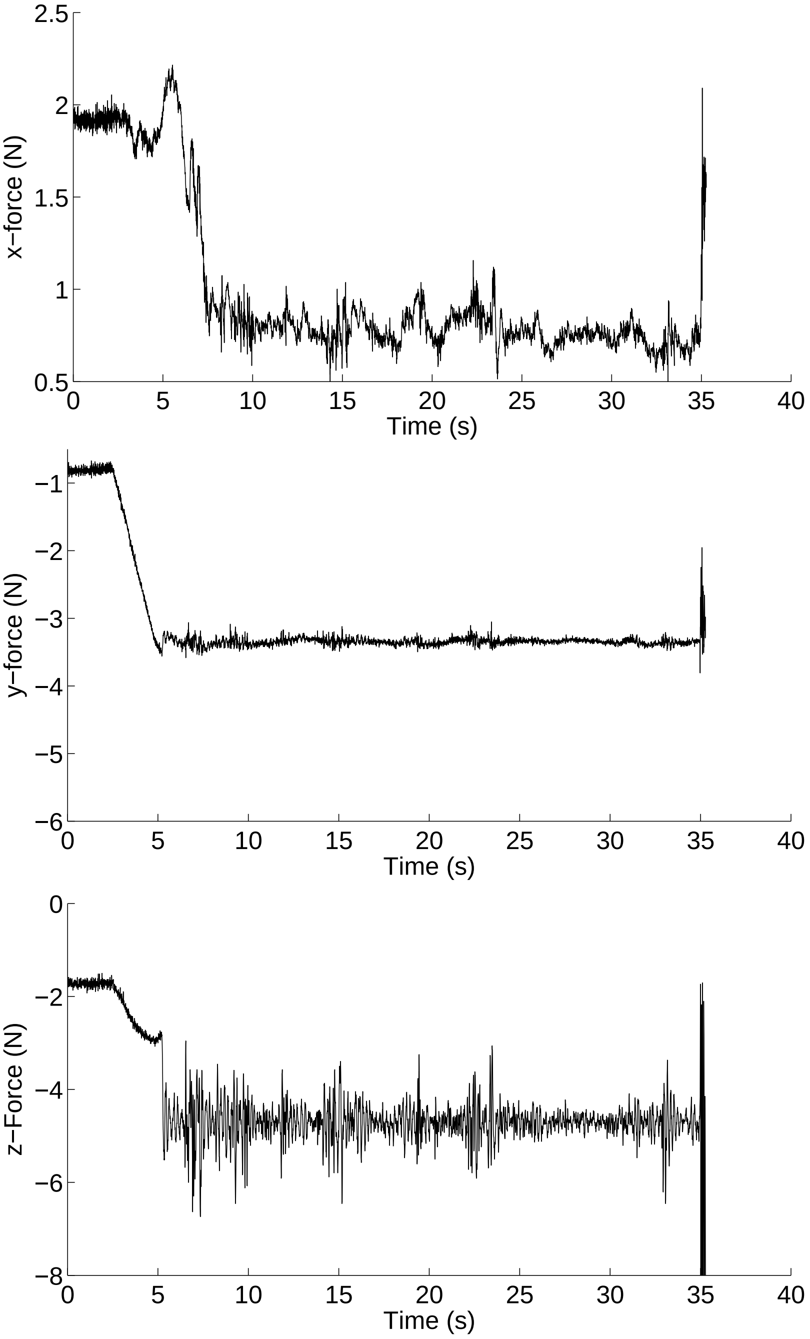


Figure 5.16: Contact forces during hybrid control with vision without filtering.

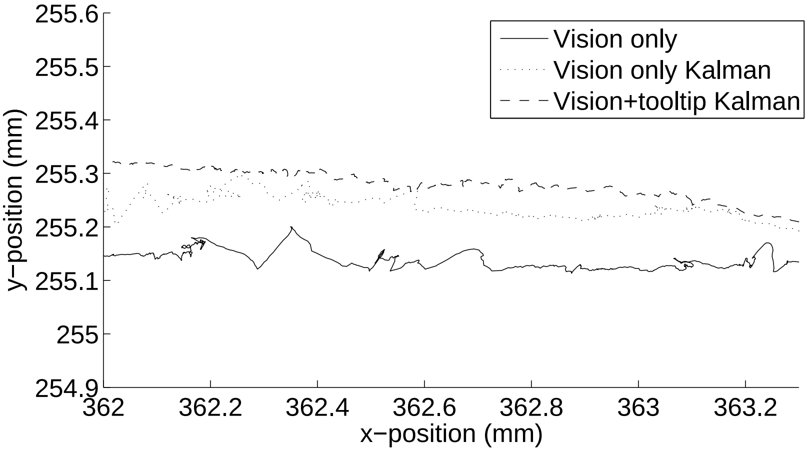


Figure 5.17: Comparison of different estimation methods with a low control gain.

Next, the control gain was increased by a decade to allow higher speed to be used during the control. Figs. 5.18 and 5.19 show the contact forces in the end-effector coordinates. The tooltip point in the Cartesian world coordinates is shown in Fig. 5.20. It can be seen that the control utilizing unfiltered visual measurement oscillates heavily and cannot follow the trajectory reliably. The EKF-based controllers, on the other hand, give a smooth response. It should be noted that each test run was unique, and therefore the absolute positions cannot be compared, but only the relative movement.

The effect of the tooltip measurement is less obvious. By looking at the EKF-estimate of the plane equations in the world coordinates, a slight difference in the two EKF methods can be seen. In Fig. 5.21 it is visible that the EKF with a tool-tip measurement deviates slightly from the mean of the visual measurement. Without ground truth data, it is difficult to determine which estimate performs better. However, both the EKF-measurements are stable in control.

The trajectories in the Cartesian world coordinates are shown in Fig. 5.22. The EKF with and without tooltip measurement are again almost identical. The unfiltered visual measurement cannot track the contour but oscillates in a fixed position.

An experiment was also made with the EKF with tooltip measurement where the tool was moved with a higher velocity along the seam. Trajectories for two different velocities are shown in Fig. 5.23. The contact to the second surface cannot be maintained when the velocity is doubled. The friction forces become higher than the desired contact force to the second surface, and the tool drifts along the first surface.

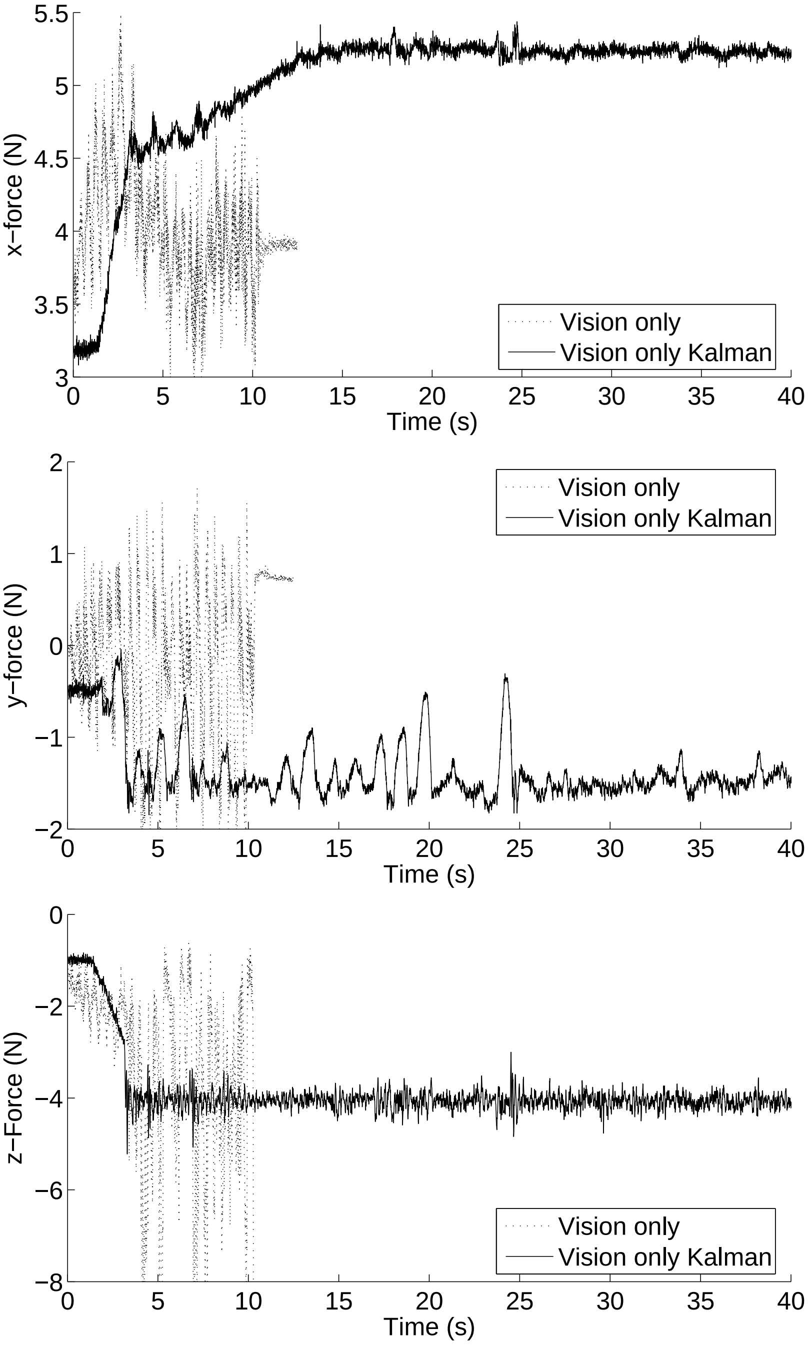


Figure 5.18: Contact forces during high gain hybrid control with vision without filtering and Kalman filter estimate using only vision.

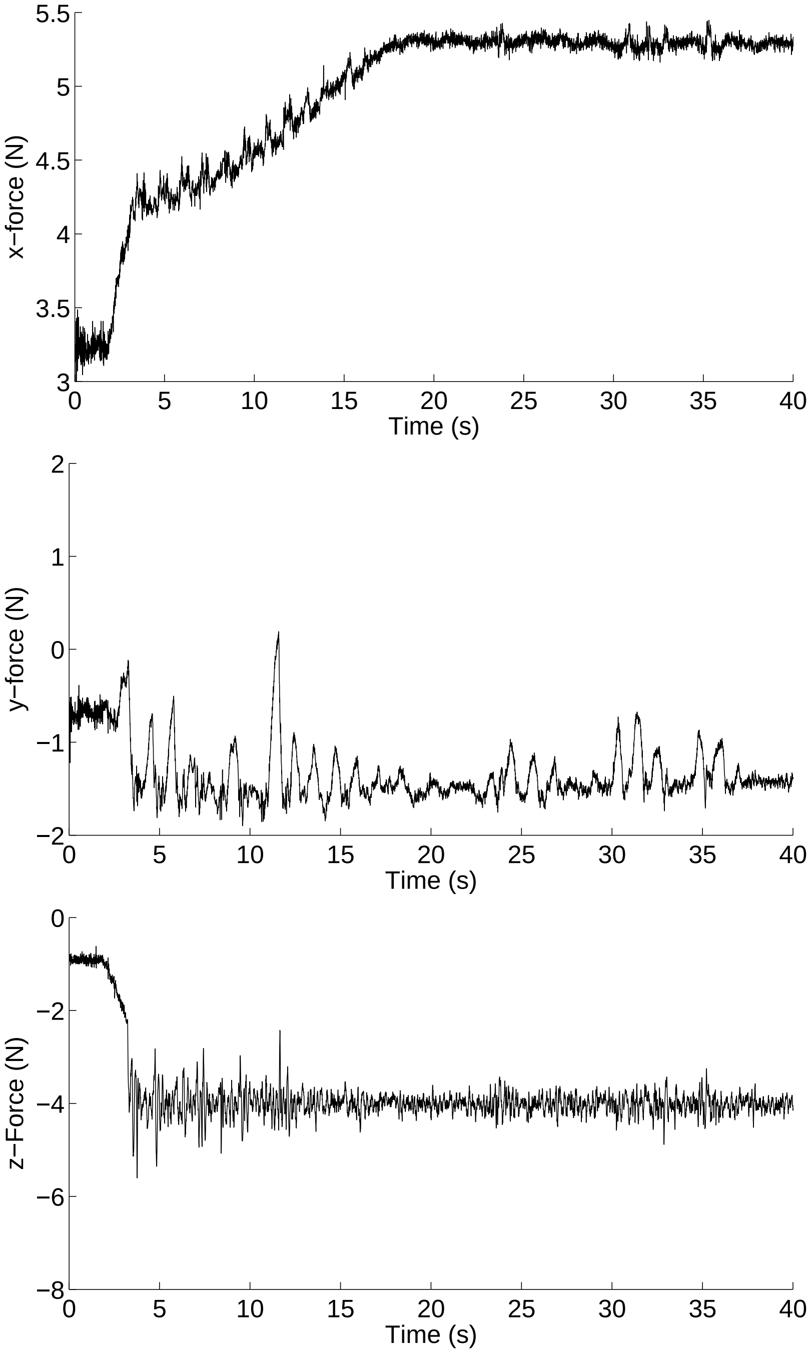


Figure 5.19: Contact forces during high gain hybrid control with Kalman filter estimate using vision and tooltip measurement.

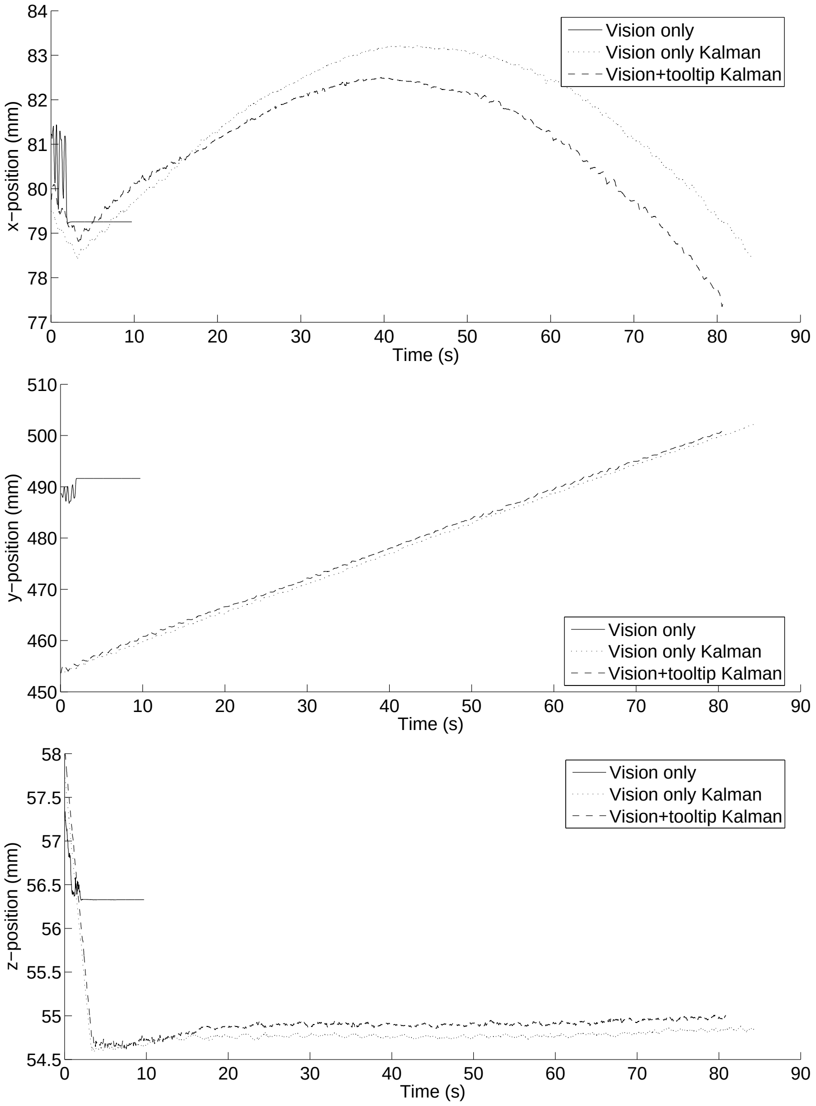


Figure 5.20: Toolpoint position in high gain hybrid control.

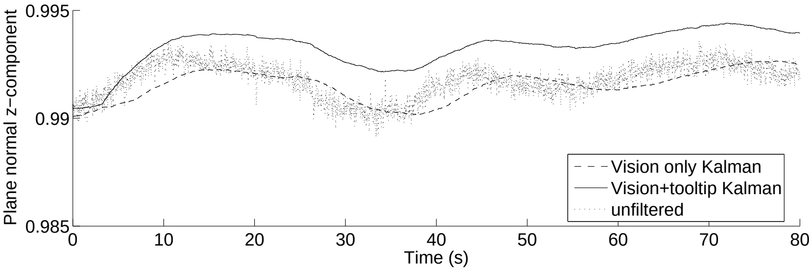


Figure 5.21: Comparison of Kalman estimates.

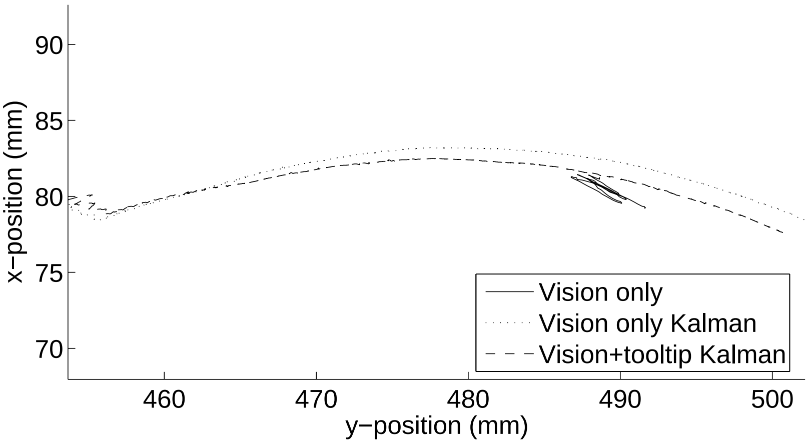


Figure 5.22: Trajectory during high gain hybrid control experiment.

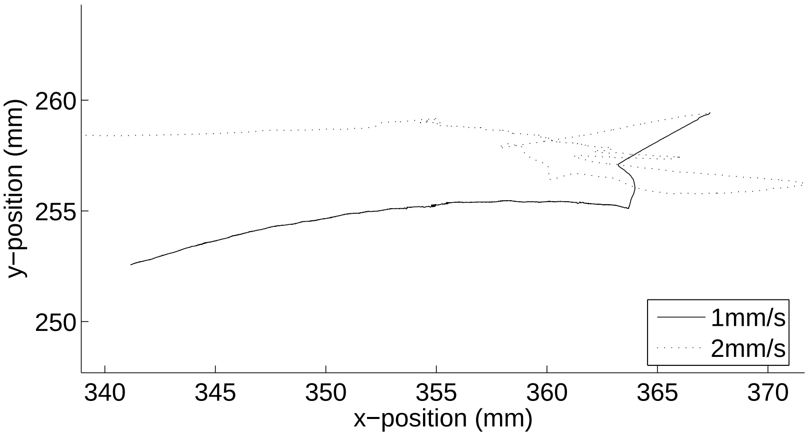


Figure 5.23: Trajectories while using different velocities during contact in hybrid control.

5.3 Shared control

In shared control, force and vision sensors are used to control the same axes simultaneously. The system for the shared control is the same as for the hybrid control introduced in the previous section. Also the same orientation controller is utilized to keep the desired tool pose during control. The position controller, on the other hand, is a shared controller using the information from the EKF estimate directly. The estimates of the two plane equations give an intersection line of the planes. The shared controller is set up to follow the intersection line by moving the tooltip towards the intersection line and at the same time along the line. Schematic of the vectors needed in shared control is presented in Fig. 5.24. Distance d_l from the toolpoint P_t to the intersection line is calculated as

$$d_l = \frac{|\boldsymbol{v_1}\ \overrightarrow{P_t P_1}\ |}{|\boldsymbol{v_1}|}, \tag{5.10}$$

where P_1 is a point arbitrary on the intersection line. P_1 can be solved by finding a solution to a system of linear equations of the intersecting planes

$$\begin{cases} a_1 x + b_1 y + c_1 z = d_1 \\ a_2 x + b_2 y + c_2 z = d_2 \end{cases}, \tag{5.11}$$

by setting for example the x- coordinate of the equations to zero and solving for y and z. The desired movement of the tooltip can be defined as a sum of vectors $d_l \frac{\boldsymbol{v_4}}{|\boldsymbol{v_4}|}$ and $v_d \frac{\boldsymbol{v_1}}{|\boldsymbol{v_1}|}$, where v_d is the desired velocity along the seam. An algorithm for the shared controller is specified in Alg. 10.

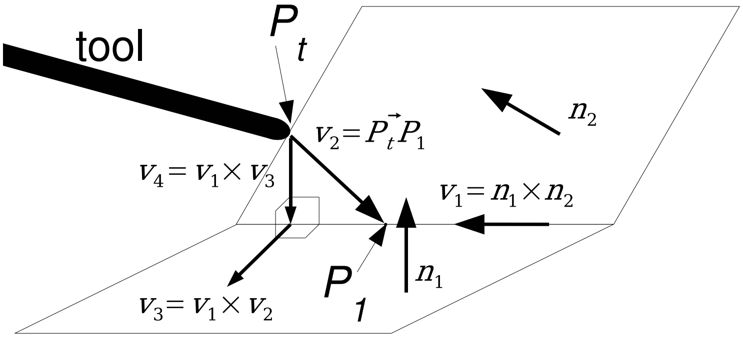


Figure 5.24: Forces during shared control.

5.3.1 Experiments

Experiments with the shared control were conducted to prove the concept of using the EKF estimate directly in control and also to compare the shared control approach to hybrid control. Different parameters for the desired velocity v_d and spring constant v_k, as well as different control frequencies were also investigated to verify the robustness

Algorithm 10 Shared control algorithm.

```
repeat
    find the direction vector $\boldsymbol{v_1} = \boldsymbol{n_1} \times \boldsymbol{n_2}$ of the intersection line of the two planes
    find a point $P_1$ on the intersection line by setting $x = 0$ and solving the plane equations for $y$ and $z$
    calculate $\boldsymbol{v_2} = \overrightarrow{P_t P_1}$, $\boldsymbol{v_3} = \boldsymbol{v_2} \times \boldsymbol{v_1}$ and $\boldsymbol{v_4} = \boldsymbol{v_3} \times \boldsymbol{v_2}$
    calculate the distance $d_l$ from the tooltip to the intersection line
    the desired movement of the toolpoint is $d_l \frac{\boldsymbol{v_4}}{|\boldsymbol{v_4}|} + v_d \frac{\boldsymbol{v_1}}{|\boldsymbol{v_1}|}$
until
```

of the control. The same experimental setup was used as with the hybrid experiments shown in Fig. 5.13.

Trajectories under different velocities during contact are shown in Fig. 5.25. The shared control oscillates slightly with the slow velocity, and when the velocity is doubled, the trajectory following becomes less reliable. When it is compared to hybrid control with the same velocities in Fig. 5.23, it can be seen that the hybrid control gives a smoother trajectory with the low velocity, but shared control can still track the trajectory even when the velocity is doubled.

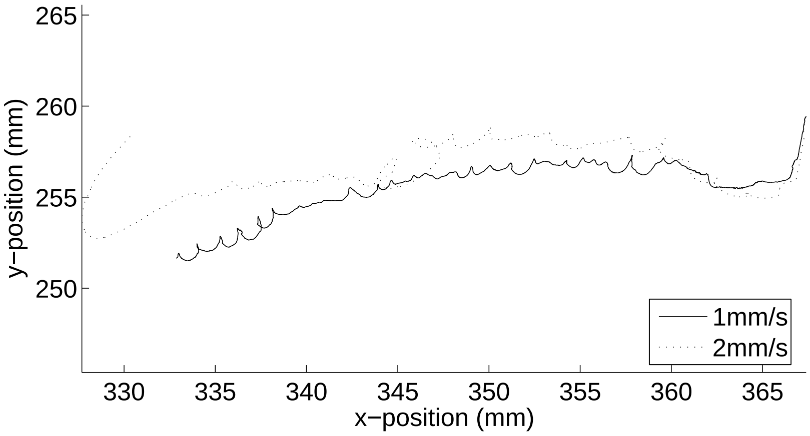


Figure 5.25: Trajectories while using different velocities during contact in shared control.

The shared controller requires an estimate of the spring constant k. In the experiments the spring constant was determined offline by measuring the contact forces f_{c1} and f_{c2} in two different end-effector positions, z_{c1} and z_{c2}, giving $k = \frac{f_{c1} - f_{c2}}{z_{c1} - z_{c2}}$. Determining the spring constant with this method is not very accurate and can give large variations to the value of the constant if several measurements are made with different tool poses or contact forces. Therefore an experiment was made where the spring constant was altered deliberately to test the performance of the controller in case the spring constant is not very accurate. Fig. 5.26 shows the resulting trajectories of three different test runs with spring constants k, $2 \times k$ and $5 \times k$. The control is stable for the first two test runs, but becomes unstable when the spring constant is further increased. The corresponding forces in Fig. 5.28 are surprisingly stable even for the $5 \times k$ spring constant, but from the x-force it is visible that contact to the second surface is not obtained.

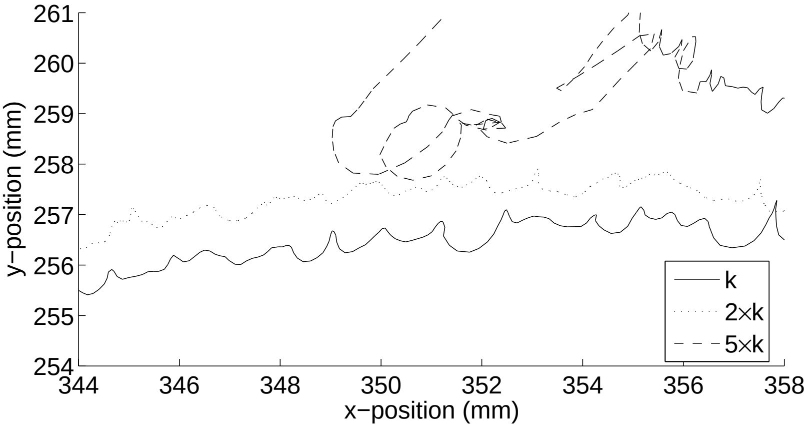


Figure 5.26: Trajectories under different spring constants.

Also experiments with different control frequencies were made in order to estimate how well the shared control method would perform if the frequency of the robot controller is low. The trajectory under low frequency shown in Fig. 5.27 is almost the same as with the higher frequency, but the force control properties shown in Fig. 5.29 are hindered considerably.

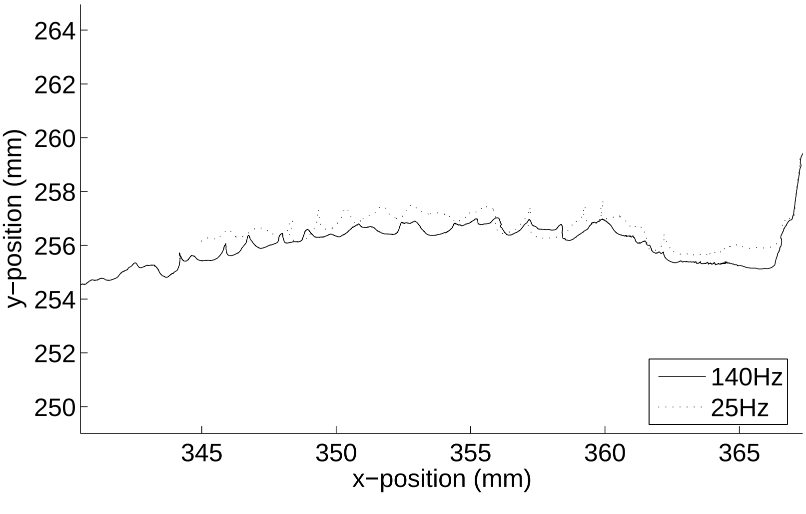


Figure 5.27: Trajectories while using different control frequencies.

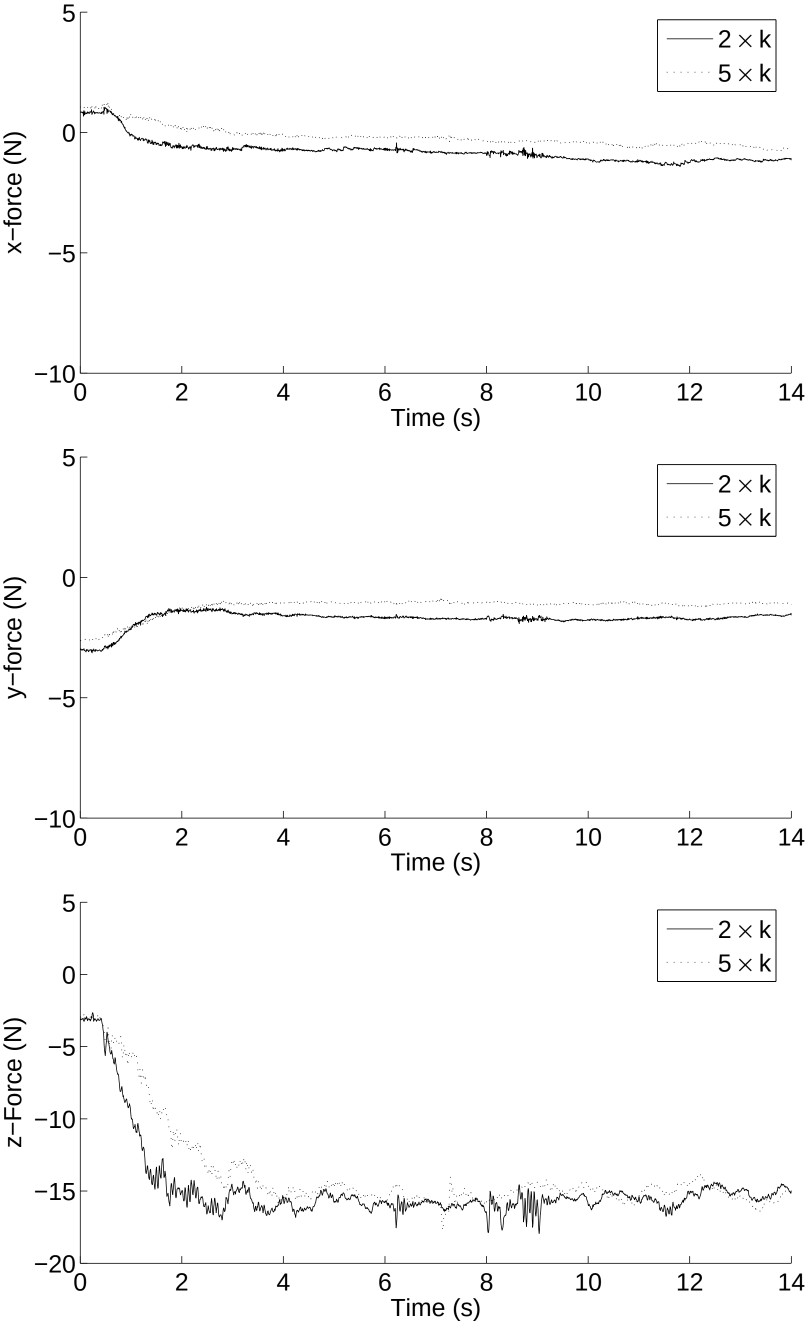


Figure 5.28: Forces under different spring constants.

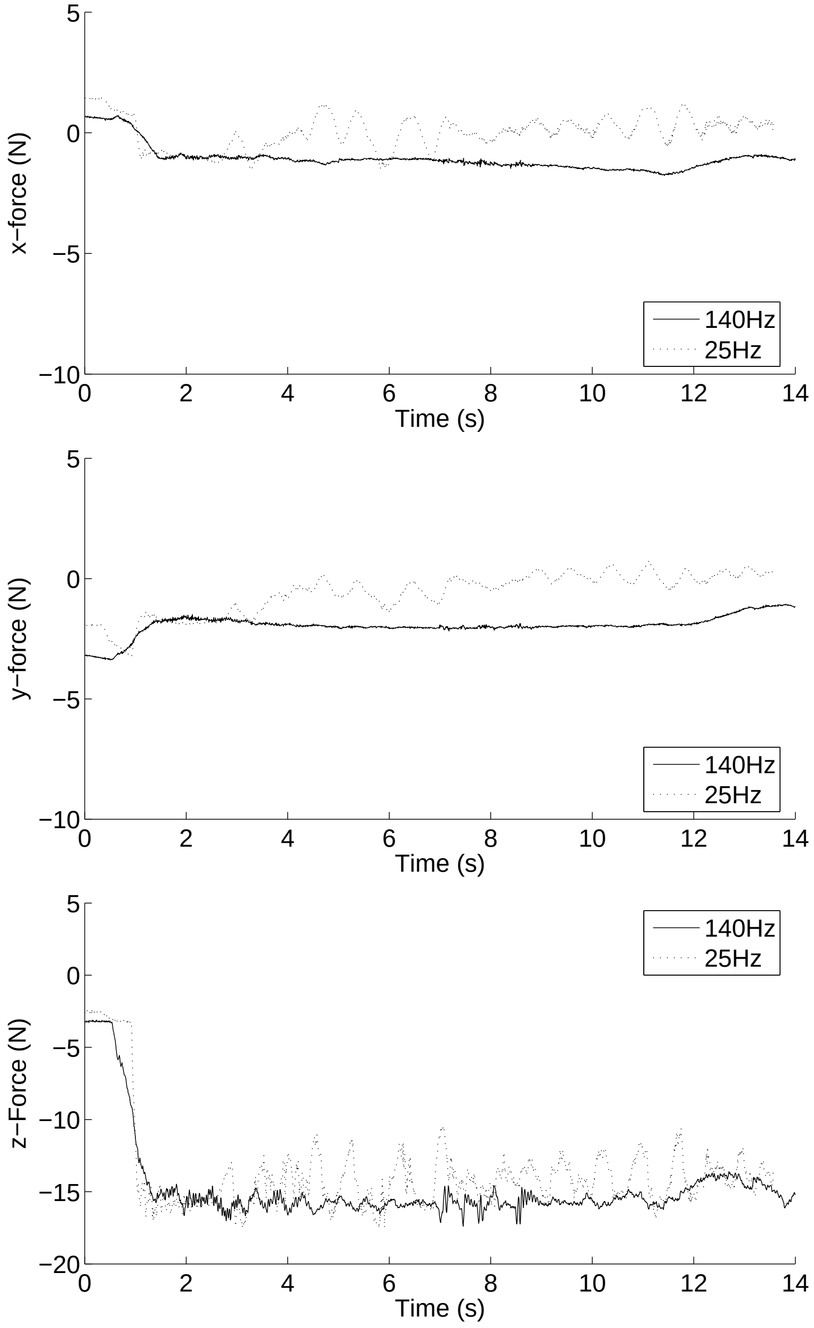


Figure 5.29: Forces while using different control frequencies.

5.4 Summary and discussion

This chapter presented three different control methods combining force and vision sensors: switched, hybrid and shared control. A novel approach for reducing the impact force overshoot of a force controller in the initial contact phase was proposed. The approach is based on measuring the distance to the target using vision and reducing the velocity before the contact. Thus, the impact force, caused mainly by the inertia of the system, is reduced. The approach gives an optimal velocity profile when the maximum velocity, maximum acceleration, and contact velocity are specified. The approach was demonstraded with experiments, which showed that combined force/vision control reduces the overshoot and approach time compared to pure force control. The optimal velocity profile with constant deceleration gives faster approach time than velocity control based on image features. Using the velocity profile method, the selection of parameters is trivial compared to image-based visual servoing.

A hybrid control was also demonstrated to follow a trajectory based on the estimate from the EKF. The estimate of the EKF was used to control the orientation of the tool when a simple proportional force control was used to track a continuous trajectory. The proposed approach was validated by experiments with a 6-DOF serial manipulator. The experiments showed that high control gains cause visual control without filtering to oscillate due to measurement uncertainty, and that the proposed method can increase the stability of control considerably. The hybrid contour tracking methods presented in literature are often restricted to planar contours [16], but the proposed approach can track arbitrary contours in 6-DOF. The proposed approach differs from traditional hybrid control methods [16, 79] by modelling the dynamic uncertain environment and fusing several sensor modalities together in one model, where the earlier methods use sensor measurements directly in control.

Experiments with shared control were also presented. The estimation method fusing force and vision in EKF was also used for shared control experiments. The system was stable under control and was able to follow the trajectory with a wide variety of operating parameters. However, according to the experiments, hybrid control outperformed shared control in all but one experiment. When a high velocity was used for contour tracking, the hybrid controller became unstable, but shared control was still able to track the contour. The shared controller requires an estimate of the spring constant k to be estimated. The experiments showed the method is not very sensitive to the value of the spring constant, and therefore the system can adapt even if the measurement of the spring constant is not very accurate or if there are variances over time. In future work the spring constant could also be added to the state of the EKF and estimated online. The tool used in the experiments was flexible and deformed while in contact. Hooke's law is not a very accurate model for the tool, as the forces also cause rotational displacement not taken into account. More realistic physical modeling of the tool or the use of a more rigid tool could improve the shared control results.

The control loop frequency of modern industrial manipulators is usually adequate for sensor based robotics. Earlier, latencies in setting the desired position, as well as low sample rate prevented efficient sensor-based control unless the sensor was integrated into the robot controller internally. There have been attempts to describe open control architecture which would not require hardware modifications of the robotic manipulator or the controller [18], but as every manufacturer has their own controllers, this is a difficult task. Experiments were also conducted where the control frequency was deliberately decreased to simulate how well the method would perform on lower-rate industrial manipulators. Even though the system was still able to follow the contour, the force control properties where hindered considerably.

CHAPTER VI

Conclusion

Sensor-based robotics allows manipulation in uncertain and dynamic environments. Force and vision offer complementary sensor modalities allowing fast and precise manipulation, but combining the sensors is problematic, as they do not share a common representation. Although framework models for sensor fusion have been proposed, multisensor control is still a challenging field of study and requires a great deal of knowledge from the robot programmer.

In the thesis, methods for fusion of proprioception, force and vision sensors in robotic manipulation were proposed. Different visual sensing approaches were proposed and also methods for determining the uncertainties of visual measurements were given. The objective of the thesis was to develop methods for estimating a dynamic uncertain environment by fusing several sensor modalities together in one model to provide better control stability and performance than using the sensor measurements directly in control. In the thesis, a method for estimating the pose of a target object that was also allowed to move during manipulation was proposed. Information from three different sensors was combined in an EKF to provide an accurate, high-sample-rate estimate of the pose of the target object and its velocities. The proposed method offered better stability and performance by reducing the oscillations and allowing higher gains to be used during control than when using the sensor measurements directly in control. Also a method for transition from vision-based control to combined vision and force control was proposed. The approach was demonstraded with experiments, which showed that the proposed transition method reduces the force overshoot and approach time compared to pure force control.

Filtering the sensor measurements often causes a phase shift in the estimate. When using EKF the expected future target pose can be predicted making it possible to track a moving target and alleviate the inherent latencies in the visual system and control loop. Three parameter axis angle representation was selected for estimating the orientations in order to minimize the amount of state variables. The use of four parameter quaternions would simplify the calculation of the gradient in the estimation model, but there is a drawback when using quaternions, as quaternions must be normalized to unit length after each update step. Although the normalization of the state vector itself is straightforward, also the covariance of the state must be modified to correspond to the normalized state. As the normalization process is nonlinear the new covariance matrix is only an approximation of the convariance of normalized state.

Three different multi-sensor control strategies were verified experimentally. Switched control uses a vision sensor for approaching the object and switches to force control once the contact is made. The experimental results verified that the velocity profile approach proposed in the thesis reduced the time to impact and allowed a smooth contact without force overshoot. In hybrid control, force and vision sensors control separate degrees of freedom. In the experiments it was shown that the proposed estimation method allowed higher control gains to be used during control than when the control was based on visual measurement alone. The results with shared control where force and vision control the same degrees of freedom confirmed that the proposed shared control method can be used in control, although the performance was not on a par with hybrid control.

Experiments were performed on several real physical platforms. The methods were validated with both hydraulic and electric motor manipulators. Also parallel and serial platforms were both covered. Switching control experiments with one dimensional hydraulic servo bench allowed to study the transition phase from visual control to force control without additional difficulties due to 3-D geometry. The tracking experiments were verified with 5-DOF parallel hydraulic manipulator, which allowed complete control of the hydraulic cylinders. Hybrid and shared control experiments were performed on an industrial 6-DOF articulated robot arm.

Visual sensor calibration is usually a laborious process, especially for position based visual servoing. Although it is often enough to calibrate the intrinsic parameters of the visual sensor once, hand-eye calibration must be performed by the robot operator every time the relative pose of the camera changes with respect to the end-effector. In this thesis, an automated procedure for structured light laser sensor calibration was proposed. Although the absolute accuracy of the laser sensor alone was insufficient for the robot control, robust and efficient control was achieved by combining force and vision sensors.

Inertial coupling of the end effector causes forces even if the end effector is not in contact with the object. The inertial coupling was not considered in this thesis as the accelerations of the end-effector during contact were very small. During the approach phase, however, inertia causes measurable forces when the end-effector is decelerating. For this reason the switching rule from visual to force control, was based on the visual information only. The switching rule could be improved by including the force measurements in the EKF model during the approach phase. This would require either compensating for the inertial effects or setting a large force threshold for the contact information during the approach phase. The shared control method could be improved by more realistic physical modeling of the tool. The tool used in the experiments was flexible and deformed while in contact causing rotational displacement not taken into account. The use of a more rigid tool could also improve the shared control results. Future work consists of validating the methods with real industrial applications such as grinding and deburring tasks.

Bibliography

[1] Agin, G. Calibration and use of a light stripe range sensor mounted to the hand of a robot. Tech. Rep. CMU-RI-TR-85-20, Robotics Institute, Pittsburgh, PA, November 1985.

[2] Alkkiomäki, O., Kyrki, V., Kälviäinen, H., Liu, Y., and Handroos, H. Smooth transition from motion to force control in robotic manipulation using vision. In *International Conference on Control, Automation, Robotics and Vision* (Dec. 2006).

[3] Alkkiomäki, O., Kyrki, V., Kälviäinen, H., Liu, Y., and Handroos, H. Multi-modal force/vision sensor fusion in 6-DOF pose tracking. In *International Conference on Advanced Robotics* (2009).

[4] Alkkiomäki, O., Kyrki, V., Kälviäinen, H., Liu, Y., and Handroos, H. Online trajectory following with position based force/vision control. In *International Conference on Advanced Robotics* (2009).

[5] Alkkiomäki, O., Kyrki, V., Liu, Y., Handroos, H., and Kälviäinen, H. Moving camera moving target tracking by integration of vision and encoder data. In *International Conference on Advanced Robotics* (2007).

[6] Alkkiomäki, O., Kyrki, V., Liu, Y., Handroos, H., and Kälviäinen, H. Challenges of vision for real-time sensor based control. In *Fifth Canadian Conference on Computer and Robot Vision* (2008).

[7] Alkkiomäki, O., Kyrki, V., Liu, Y., Handroos, H., and Kälviäinen, H. Complementing visual tracking of moving targets by fusion of tactile sensing. *Robotics and Autonomous Systems 57*, 11 (2009), 1129 – 1139.

[8] Amirat, Y., Pontnau, J., Babaci, S., and Francois, C. A flexible assembly cell integrating a paraller manipulator for accurate automatic assembly tasks. In *Conference on Industrial Automation and Control* (1995), pp. 308 – 13.

[9] Andersen, G. I., Christensen, A. C., and Ravn, O. Augmented models for improving vision control of a mobile robot. In *IEEE International Conference on Control and Applications* (1994).

[10] Andrews, S., and Lang, J. Interactive scanning of haptic textures and surface compliance. *Sixth International Conference on 3-D Digital Imaging and Modeling* (Aug. 2007), 99 – 106.

[11] Arkin, R. C., and Mackenzie, D. C. Planning to behave: A hybrid deliberative/reactive robot control architecture for mobile manipulation. In *International Symposium on Robotics and Manufacturing, Maui, HI* (1994).

[12] Baeten, J., Bruyninckx, H., and Schutter, J. D. Shared control in hybrid vision/force robotic servoing using the task frame. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Lausanne, Switzerland, Oct 2002), pp. 2128 – 2133.

[13] Baeten, J., Bruyninckx, H., and Schutter, J. D. Tool/camera configurations for eye-in-hand hybrid vision/force control. In *IEEE International Conference on Robotics and Automation* (May 2002), pp. 1704 – 1709.

[14] Baeten, J., Bruyninckx, H., and Schutter, J. D. Integrated vision/force robotic servoing in the task frame formalism. *International Journal of Robotics Research 22*, 10-11 (Oct-Nov 2003), 941 – 954.

[15] Baeten, J., and Schutter, J. D. Improving force controlled planar contour following using on-line eye-in-hand vision based feedforward. In *International Conference on Advanced Intelligent Mechatronics* (Sep 1999), pp. 902 – 907.

[16] Baeten, J., and Schutter, J. D. Hybrid vision/force control at corners in planar robotic-contour following. *IEEE/ASME Transactions on Mechatronics 7*, 2 (Jun 2002), 143 – 151.

[17] Bates, D. M., and Watts, D. G. *Nonlinear Regression Analysis and its Applications.* Wiley, New York, 1988.

[18] Bona, B., Greco, C., and Calvelli, T. Experimental validation of a simple architecture for force/torque and vision control in a 6-dof industrial robot. In *IEEE International Conference on Robotics and Automation* (Oct 2003), vol. 3, pp. 3010 – 3015.

[19] Bone, G., and Elbestawi, M. Sensing and control for automated robotic edge deburring. In *International Conference on Industrial Electronics, Control, Instrumentation and Automation* (Nov 1992), vol. 2, pp. 927 – 933.

[20] Bouguet, J. Y. Camera calibration toolbox for matlab, 2008. www.vision.caltech.edu/bouguetj.

[21] Bruyninckx, H., and De Schutter, J. Specification of force-controlled actions in the task frame formalism a synthesis. *IEEE Transactions on Robotics and Automation 12*, 4 (Aug 1996), 581 – 589.

[22] Büerkle, A., Schmoeckel, F., Wörn, H., Amavasai, B., Caparrelli, F., and Travis, J. A versatile vision system for micromanipulation tasks. In *International Conference on Multisensor Fusion and Integration for Intelligent* (Aug 2003), pp. 271–276.

[23] Chang, W.-C. Cartesian-based planar contour following with automatic hybrid force and visual feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2004).

[24] Chaumette, F. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control*, no. 237 in Lecture Notes in Control and Information Sciences. Springer-Verlag, 1998, pp. 66 – 78.

[25] Chaumette, F., Rives, P., and Espiau, B. The task function approach applied to vision-based control. In *International Conference on Advanced Robotics* (Jun 1991), vol. 2, pp. 1392 – 1397.

[26] Chen, C., and Kak, A. Modeling and calibration of a structured light scanner for 3-D robot vision. In *IEEE International Conference on Robotics and Automation* (Mar 1987), vol. 4, pp. 807 – 815.

[27] Choomuang, R., and Afzulpurkar, N. Hybrid Kalman filter/fuzzy logic based position control of autonomous mobile robot. *International Journal of Advanced Robotic Systems 2*, 3 (Sept. 2005), 197 – 208.

[28] Claes, K., Koninckx, T., and Bruyninckx, H. Automatic burr detection on surfaces of revolution based on adaptive 3D scanning. In *International Conference on 3-D Digital Imaging and Modeling* (Jun 2005), pp. 212 – 219.

[29] Comport, A., Marchand, E., and Chaumette, F. A real-time tracker for markerless augmented reality. In *IEEE International Symposium on Mixed and Augmented Reality* (2003), pp. 36 – 45.

[30] De Schutter, J., De Laet, T., Rutgeerts, J., Decre, W., Smits, R., Aertbelien, E., Claes, K., and Bruyninckx, H. Constraint-based Task Specification and Estimation for Sensor-Based Robot Systems in the Presence of Geometric Uncertainty. *International Journal of Robotics Research 26*, 5 (2007), 433 – 455.

[31] DeMenthon, D., and Davis, L. S. Model-based object pose in 25 lines of code. In *European Conference on Computer Vision* (1992), pp. 335 – 343.

[32] Deng, L., Wilson, W., and Janabi-Sharifi, F. Characteristics of robot visual servoing methods and target model estimation. In *IEEE International Symposium on Intelligent Control* (2002), pp. 684 – 689.

[33] Dorf, R. C., and Bishop, R. H. *Modern Control Systems*. Prentice-Hall, 2004.

[34] Dornfeld, D. Intelligent deburring of precision components. In *International Conference on Industrial Electronics, Control, Instrumentation and Automation* (Nov 1992), vol. 2, pp. 953 – 960.

[35] Drummond, T., and Cipolla, R. Real-time visual tracking of complex structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 7 (2002), 932 – 946.

[36] Enikov, E., Minkov, L., and Clark, S. Microassembly experiments with transparent electrostatic gripper under optical and vision-based control. In *IEEE Transactions on Industrial Electronics* (Aug 2005), vol. 52, pp. 1005 – 1012.

[37] Espiau, B. Effect of camera calibration errors on visual servoing in robotics. In *International Symposium on Experimental Robotics* (1993), pp. 187 – 193.

[38] Haralick, R. M., and Shapiro, L. G. *Computer and Robot Vision, Volume II.* Prentice-Hall, 1993.

[39] Harris, C., and Stephens, M. A combined corner and edge detection. In *Fourth Alvey Vision Conference* (1988), pp. 147 – 151.

[40] Heikkila, J. Moment and curvature preserving technique for accurate ellipse boundary detection. In *International Conference on Pattern Recognition* (1998).

[41] Horaud, R., Dornaika, F., Lamiroy, B., and Christy, S. Object pose: The link between weak perspective, paraperspective, and full perspective. *International Journal of Computer Vision 22*, 2 (March 1997), 173 – 189.

[42] Horaud, R. P., and Dornaika, F. Hand-eye calibration. Tech. Rep. RT 99IMAG, LIFIA–IMAG–INRIA Rhône-Alpes, Grenoble, France, June 1994.

[43] Hutchinson, S., Hager, G., and Corke, P. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation 12* (1996), 651 – 670.

[44] Jurie, F., and Dhome, M. Real time tracking of 3D objects: an efficient and robust approach. *Pattern Recognition 35* (2002), 317 – 328.

[45] Kim, B., Kang, H., and Deok-Ho. Flexible microassembly system based on hybrid manipulation scheme. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (2003).

[46] Klein, G., and Drummond, T. Robust visual tracking for non-instrumented augmented reality. In *2nd IEEE and ACM International Symposium on Mixed and Augmented Reality* (2003), pp. 113 – 122.

[47] Komuro, T., Ishii, I., Ishikawa, M., and Yoshida, A. A digital vision chip specialized for high-speed target tracking. *IEEE Transactions on Electron Devices 50*, 1 (Jan 2003), 191 – 199.

[48] Kragic, D., and Christensen, H. I. Robust visual servoing. *International Journal of Robotic Research 22*, 10-11 (2003), 923 – 940.

[49] Kragic, D., and Kyrki, V. Initialization and system modeling in 3-D pose tracking. In *International Conference on Pattern Recognition* (Hong Kong, China, 2006).

[50] Kwok, K., Loucks, C., and Driessen, B. Rapid 3-D digitizing and tool path generation for complex shapes. In *IEEE International Conference on Robotics and Automation* (May 1998), vol. 4, pp. 2789 – 2794.

[51] Kyrki, V., and Kragic, D. Integration of model-based and model-free cues for visual object tracking in 3D. In *IEEE International Conference on Robotics and Automation* (2005), pp. 1566 – 1572.

[52] Kyrki, V., Kragic, D., and Christensen, H. I. Measurement errors in visual servoing. *IEEE International Conference on Robotics and Automation 2* (April 26–May 1 2004), 1861 – 1867.

[53] Kyrki, V., and Schmock, K. Integation methods of model-free features for 3D tracking. In *SCIA* (2005), H. Kälviäinen, J. Parkkinen, and A. Kaarna, Eds., vol. 3540 of *Lecture Notes in Computer Science*, Springer, pp. 557 – 566.

[54] Lang, J., Pai, D., and Woodham, R. Robotic acquisition of deformable models. In *IEEE International Conference on Robotics and Automation* (May 2002), pp. 933 – 938.

[55] Langlois, D., and Croft, E. A low-level control policy for data fusion. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems* (Aug 2001).

[56] Larsen, T. D., Bak, M., Andersen, N. A., and Ravn, O. Location estimation for an autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry. In *International Conference on Multisource-Multisensor Information Fusion* (1998).

[57] Lee, J. M., Son, K., Lee, M., Choi, J., Han, S., and Lee, M. H. Localization of a mobile robot using the image of a moving object. *IEEE Transactions on Industrial Electronics 50*, 3 (2003), 612 – 619.

[58] Lewis, F. L. *Optimal Estimation.* Wiley-Interscience, 1986.

[59] Lippiello, V., Siciliano, B., and Villani, L. 3D pose estimation for robotic applications based on a multi-camera hybrid visual system. *IEEE International Conference on Robotics and Automation* (2006), 2732 – 2737.

[60] Lippiello, V., Siciliano, B., and Villani, L. A position-based visual impedance control for robot manipulators. *IEEE International Conference on Robotics and Automation* (April 2007), 2068 – 2073.

[61] Liu, Y., and Handroos, H. Modeling and control of a parallel redundant manipulator with hydraulic actuators. *Proceedings of the IMech Part I Journal of Systems and Control Engineering 220*, 3 (2006), 211 – 221.

[62] Liu, Y., Handroos, H., Alkkiomäki, O., Kyrki, V., and Kälviäinen, H. Tracking of a moving target by combination of force/velocity control based on vision for a hydraulic manipulator. In *IEEE International Conference on Mechatronics and Automation* (Aug. 2007), pp. 3226 – 3231.

[63] Liu, Y., Handroos, H., Alkkiomäki, O., Kyrki, V., and Kälviäinen, H. Development of a hybrid position/force controlled hydraulic parallel robot for impact treatment. In *International Conference on Machine Automation* (2008).

[64] Malis, E., Chaumette, F., and Boudet, S. 2 1/2-D visual servoing. *IEEE Transactions on Robotics and Automation 15*, 2 (1999), 238 – 250.

[65] Malis, E., Morel, G., and Chaumette, F. Robot control using disparate multiple sensors. *International Journal of Robotic Research 20*, 5 (2001), 364 – 377.

[66] Malis, E., and Rives, P. Robustness of image-based visual servoing with respect to depth distribution errors. In *IEEE International Conference on Robotics and Automation* (Taipei, Taiwan, 2003), pp. 1056 – 1061.

[67] Mason, M. Compliance and force control for computer-controlled manipulators. *IEEE Transactions on Systems, Man, and Cybernetics 11*, 1 (1981), 418 – 432.

[68] Mason, M., and Salisbury, J. K. *Robot Hands and the Mechanics of Manipulation.* MIT Press, Cambridge, MA, 1985.

[69] Merlet, J.-P. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *International Journal of Robotics Research 23*, 3 (2004), 221 – 236.

[70] Mihaylova, L., Bruyninckx, H., De Schutter, J., and Staffetti, E. Planar contour tracking in the presence of pose and model errors by Kalman filtering techniques. In *Multisensor Fusion and Integration for Intelligent Systems, 2001. MFI 2001. International Conference on* (2001), pp. 329 – 334.

[71] Namiki, A., Imai, Y., Ishikawa, M., and Kaneko, M. Development of a high-speed multifingered hand system and its application to catching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Nov 2003), pp. 2666 – 2671.

[72] Nelson, B., Morrow, J., and Khosla, P. Improved force control through visual servoing. In *American Control Conference* (Jun 1995).

[73] Nelson, B., Morrow, J., and Khosla, P. Robotic manipulation using high bandwidth force and vision feedback. *An International Journal on Mathematical and Computer Modelling* (1995).

[74] Nelson, B. J., and Khosla, P. K. Force and vision resolvability for assimilating disparate sensory feedback. *IEEE Transactions on Robotics and Automation 12* (Oct 1996), 714 – 731.

[75] Nomura, H., and Naito, T. Integrated visual servoing system to grasp industrial parts moving on conveyer by controlling 6DOF arm. *IEEE International Conference on Systems, Man, and Cybernetics 3* (2000), 1768 – 1775.

[76] Olsson, T., Bengtsson, J., Johansson, R., and Malm, H. Force control and visual servoing using planar surface identification. In *IEEE International Conference on Robotics and Automation* (May 2002), pp. 4211 – 4216.

[77] Park, F., and Martin, B. Robot sensor calibration: Solving Ax=Xb on the Euclidean group. *IEEE Transactions on Robotics and Automation 10*, 5 (October 1994), 717 – 721.

[78] Payeur, P., Pasca, C., Cretu, A.-M., and Petriu, E. Intelligent haptic sensor system for robotic manipulation. In *IEEE Transactions on Instrumentation and Measurement* (Aug 2005), vol. 54, pp. 1583 – 1592.

[79] Pichler, A., and Jägersand, M. Uncalibrated hybrid force-vision manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Nov 2000).

[80] Pomares, J., Gil, P., García, G. J., Sebastián, J. M., and Torres, F. Improving detection of surface discontinuities in visual-force control systems. *Image and Vision Computing 26*, 10 (2008), 1435 – 1447.

[81] Pomares, J., Puente, S., Garcia, G., and Torres, F. Multi-sensorial system for the generation of disassembly trajectories. In *World Automation Congress* (July 2006).

[82] Pomares, J., and Torres, F. Movement-flow-based visual servoing and force control fusion for manipulation tasks in unstructured environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 35*, 1 (Feb. 2005), 4 – 15.

[83] Prats, M., Martinet, P., del Pobil, A., and Lee, S. Vision force control in task-oriented grasping and manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Nov. 2007), pp. 1320 – 1325.

[84] Reid, I. Projective calibration of a laser-stripe range finder. *Image and Vision Computing 14* (1996), 659 666.

[85] Ryong, L. S. Position estimation of free-ranging AGV systems using the extended Kalman filter technique. *Transactions of the Korean Institute of Electrical Engineers 38*, 12 (Dec. 1989), 971 – 982.

[86] Samson, C., Espiau, B., and Borgne, M. L. *Robot Control: The Task Function Approach.* Oxford University Press, 1991.

[87] Schutter, J. D., Bruyninckx, H., Geeter, J. D., Katupitiya, J., and Demey, S. Estimating first order geometric parameters and monitoring contact transitions during force controlled compliant motion. *International Journal of Robotics Research 18* (1999), 1161 – 1184.

[88] Selleck, C., and Loucks, C. A system for automatic edge finishing. In *IEEE International Conference on Systems Engineering* (Aug 1990), vol. 58, pp. 423 – 429.

[89] Sim, T., Hong, G., and Lim, K. Multirate predictor control scheme for visual servo control. *Control Theory and Applications 149*, 2 (Mar 2002), 117 – 124.

[90] Strobl, K., Sepp, W., Wahl, E., Bodenmuller, T., Suppa, M., Seara, J., and Hirzinger, G. The DLR multisensory hand-guided device: the laser stripe profiler. In *IEEE International Conference on Robotics and Automation* (May 2004), vol. 2, pp. 1927 – 1932.

[91] Tacvision project. Integration of Visual and Tactile Control, 2009. http://www.it.lut.fi/project/tacvision/.

[92] Tanaka, S., Tanigawa, T., Abe, Y., Uejo, M., and Tanaka, H. T. Active mass estimation with haptic vision. In *International Conference on Pattern Recognition* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 256 – 261.

[93] Taylor, G., and Kleeman, L. Fusion of multimodal visual cues for model-based object tracking. In *Australiasian Conference on Robotics and Automation* (2003).

[94] Thomas, U., Florke, J., Detering, S., and Wahl, F. An integrative approach for multi-sensor based robot task programming. In *IEEE International Conference on Robotics and Automation* (2004), pp. 1149 – 1154.

[95] Thorne, H. F. Tool center point calibration for spot welding guns. US Patent (US5910719), 1999.

[96] Trucco, E., and Verri, A. *Introductory Techniques for 3-D Computer Vision.* Prentice-Hall, 1998.

[97] Tsai, R. Y., and Lenz, R. K. A new technique for fully autonomous and efficient 3D robotics hand-eye calibration. *IEEE Transactions on Robotics and Automation 5* (1989), 345 – 358.

[98] Ueda, N., Hirai, S., and T.Tanaka, H. Extracting rheological properties of deformable objects with haptic vision. In *IEEE International Conference on Robotics and Automation* (Apr 2004), pp. 3902 – 3907.

[99] Uejo, M., and Tanaka, H. T. Active modeling of articulated objects with haptic vision. In *International Conference on Pattern Recognition* (2004).

[100] Vacchetti, L., Lepetit, V., and Fua, P. Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence 26* (2004), 1385 – 1391.

[101] Verscheure, D., Swevers, J., Bruyninckx, H., and Schutter, J. On-line identification of contact dynamics in the presence of geometric uncertainties. *IEEE International Conference on Robotics and Automation* (May 2008), 851 – 856.

[102] Welch, G., and Bishop, G. SCAAT: Incremental tracking with incomplete information. In *Computer graphics and interactive techniques* (Los Angeles, CA, USA, August 3–8 1997), pp. 333 – 344.

[103] Wilson, W., Williams Hulls, C., and Bell, G. Relative end-effector control using cartesian position based visual servoing. *IEEE Transactions on Robotics and Automation 12*, 5 (Oct 1996), 684 – 696.

[104] Xiao, D., Ghosh, B., Xi, N., and Tarn, T. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Transactions on Control Systems Technology 8*, 4 (Jul 2000), 635 – 645.

[105] Yamauchi, K., Saito, H., and Sato, Y. Calibration of a structured light system by observing planar object from unknown viewpoints. In *International Conference on Pattern Recognition* (Dec. 2008), pp. 1–4.

[106] Young, D. L., Byung, H. K., and Jong, O. P. Robotic deburring strategy using burr shape recognition,. In *IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct 1999), vol. 3, pp. 1513 – 1518.

[107] Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 11 (2000), 1330 – 1334.

[108] Zhao, Y. Cheah, C. Hybrid vision-force control for robot with uncertainties. In *IEEE International Conference on Robotics and Automation* (2004).

[109] Zhou, F., and Zhang, G. Complete calibration of a structured light stripe vision sensor through planar target of unknown orientations. *Image and Vision Computing 23*, 1 (2005), 59 – 67.

[110] Zhou, Y., Nelson, B., and Vikramaditya, B. Fusing force and vision feedback for micromanipulation. In *IEEE International Conference on Robotics and Automation* (Leuven, Belgium, 1998), pp. 1220 – 1225.

ACTA UNIVERSITATIS LAPPEENRANTAENSIS

316. DABAGHMESHIN, MAHSA. Modeling the transport phenomena within the arterial wall: porous media approach. 2008. Diss.

317. HAIMALA, JUHA. Supplier´s position in project marketing networks. 2008. Diss.

318. UOTILA, TUOMO. The use of future-oriented knowledge in regional innovation processes: research on knowledge generation, transfer and conversion. 2008. Diss.

319. LAPPALAINEN, TOMMI. Validation of plant dynamic model by online and laboratory measurements – a tool to predict online COD loads out of production of mechanical printing papers. 2008. Diss.

320. KOSONEN, ANTTI. Power line communication in motor cables of variable-speed electric drives – analysis and implementation. 2008. Diss.

321. HANNUKAINEN, PETRI. Non-linear journal bearing model for analysis of superharmonic vibrations of rotor systems. 2008. Diss.

322. SAASTAMOINEN, KALLE. Many valued algebraic structures as measures of comparison. 2008. Diss.

323. PEUHU, LEENA. Liiketoimintastrategisten vaatimusten syntyminen ja niiden toteutumisen arviointi keskisuurten yritysten toiminnanohjausjärjestelmähankkeissa: Tapaustutkimus kolmesta teollisuusyrityksestä ja aineistolähtöinen teoria. 2008. Diss.

324. BANZUZI, KUKKA. Trigger and data link system for CMS resistive plate chambers at the LHC accelerator. 2008. Diss.

325. HIETANEN, HERKKO. The pursuit of efficient copyright licensing – How some rights reserved attempts to solve the problems of all rights reserved. 2008. Diss.

326. SINTONEN, SANNA. Older consumers adopting information and communication technology: Evaluating opportunities for health care applications. 2008. Diss.

327. KUPARINEN, TONI. Reconstruction and analysis of surface variation using photometric stereo. 2008. Diss.

328. SEPPÄNEN, RISTO. Trust in inter-organizational relationships. 2008. Diss.

329. VISKARI, KIRSI. Drivers and barriers of collaboration in the value chain of paperboard-packed consumer goods. 2008. Diss.

330. KOLEHMAINEN, EERO. Process intensification: From optimised flow patterns to microprocess technology. 2008. Diss.

331. KUOSA, MARKKU. Modeling reaction kinetics and mass transfer in ozonation in water solutions. 2008. Diss.

332. KYRKI, ANNA. Offshore sourcing in software development: Case studies of Finnish-Russian cooperation. 2008. Diss.

333. JAFARI, AREZOU. CFD simulation of complex phenomena containing suspensions and flow through porous media. 2008. Diss.

334. KOIVUNIEMI, JOUNI. Managing the front end of innovation in a networked company environment – Combining strategy, processes and systems of innovation. 2008. Diss.

335. KOSONEN, MIIA. Knowledge sharing in virtual communities. 2008. Diss.

336. NIEMI, PETRI. Improving the effectiveness of supply chain development work – an expert role perspective. 2008. Diss.

337. LEPISTÖ-JOHANSSON, PIIA. Making sense of women managers´ identities through the constructions of managerial career and gender. 2009. Diss.

338. HYRKÄS, ELINA. Osaamisen johtaminen Suomen kunnissa. 2009. Diss.

339. LAIHANEN, ANNA-LEENA. Ajopuusta asiantuntijaksi – luottamushenkilöarvioinnin merkitys kunnan johtamisessa ja päätöksenteossa. 2009. Diss.

340. KUKKURAINEN, PAAVO. Fuzzy subgroups, algebraic and topological points of view and complex analysis. 2009. Diss.

341. SÄRKIMÄKI, VILLE. Radio frequency measurement method for detecting bearing currents in induction motors. 2009. Diss.

342. SARANEN, JUHA. Enhancing the efficiency of freight transport by using simulation. 2009. Diss.

343. SALEEM, KASHIF. Essays on pricing of risk and international linkage of Russian stock market. 2009. Diss.

344. HUANG, JIEHUA. Managerial careers in the IT industry: Women in China and in Finland. 2009. Diss.

345. LAMPELA, HANNELE. Inter-organizational learning within and by innovation networks. 2009. Diss.

346. LUORANEN, MIKA. Methods for assessing the sustainability of integrated municipal waste management and energy supply systems. 2009. Diss.

347. KORKEALAAKSO, PASI. Real-time simulation of mobile and industrial machines using the multibody simulation approach. 2009. Diss.

348. UKKO, JUHANI. Managing through measurement: A framework for successful operative level performance measurement. 2009. Diss.

349. JUUTILAINEN, MATTI. Towards open access networks – prototyping with the Lappeenranta model. 2009. Diss.

350. LINTUKANGAS, KATRINA. Supplier relationship management capability in the firm´s global integration. 2009. Diss.

351. TAMPER, JUHA. Water circulations for effective bleaching of high-brightness mechanical pulps. 2009. Diss.

352. JAATINEN, AHTI. Performance improvement of centrifugal compressor stage with pinched geometry or vaned diffuser. 2009. Diss.

353. KOHONEN, JARNO. Advanced chemometric methods: applicability on industrial data. 2009. Diss.

354. DZHANKHOTOV, VALENTIN. Hybrid LC filter for power electronic drivers: theory and implementation. 2009. Diss.

355. ANI, ELISABETA-CRISTINA. Minimization of the experimental workload for the prediction of pollutants propagation in rivers. Mathematical modelling and knowledge re-use. 2009. Diss.

356. RÖYTTÄ, PEKKA. Study of a vapor-compression air-conditioning system for jetliners. 2009. Diss.

357. KÄRKI, TIMO. Factors affecting the usability of aspen (Populus tremula) wood dried at different temperature levels. 2009. Diss.