ESPOO 2003 VTT PUBLICATIONS 509

Mikko Sallinen

Modelling and estimation of spatial relationships in sensor-based robot workcells

VTT PUBLICATIONS 509

Modelling and estimation of spatial relationships in sensor-based robot workcells

Mikko Sallinen

VTT Electronics

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in the Auditorium YB 210 at University of Oulu (Oulu, Finland) on 28th of November, 2003, at 12 o'clock noon.

ISBN 951–38–6247–X (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6248–8 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)

JULKAISIJA – UTGIVARE – PUBLISHER

VTT, Vuorimiehentie 5, PL 2000, 02044 VTT
puh. vaihde (09) 4561, faksi (09) 456 4374

VTT, Bergsmansvägen 5, PB 2000, 02044 VTT
tel. växel (09) 4561, fax (09) 456 4374

VTT Technical Research Centre of Finland, Vuorimiehentie 5, P.O.Box 2000, FIN–02044 VTT, Finland
phone internat. + 358 9 4561, fax + 358 9 456 4374

VTT Elektroniikka, Kaitoväylä 1, PL 1100, 90571 OULU
puh. vaihde (08) 551 2111, faksi (08) 551 2320

VTT Elektronik, Kaitoväylä 1, PB 1100, 90571 ULEÅBORG
tel. växel (08) 551 2111, fax (08) 551 2320

VTT Electronics, Kaitoväylä 1, P.O.Box 1100, FIN–90571 OULU, Finland
phone internat. + 358 8 551 2111, fax + 358 8 551 2320

Technical editing Maini Manninen

Otamedia Oy, Espoo 2003

Sallinen, Mikko. Modelling and estimation of spatial relationships in sensor-based robot workcells. Espoo 2003. VTT Publications 509. 218 p.

Keywords intelligent robots, parameter estimation, spatial uncertainties, pose estimation, sensing planning

Abstract

Requirements for verifying spatial relations in robot workcell in terms of accuracy and repeatability are increasing. Improvements in the performance of industrial robots have extended the range of applications in to new fields in which flexibility, high payload, accuracy and repeatability are needed. To satisfy the requirements of overall geometric performance and flexibility in a robot system, a sensor-based, intelligent robot can be used.

One of the goals of this thesis was to develop a flexible, CAD–based robot system. Modern applications and cost-effective production require off-line programming, and the difference between off-line programming systems and actual robot workcells has to be illustrated somehow in order to verify the gap between simulation models and actual robot systems.

A method for modelling spatial uncertainties in a robot system is presented here, based on Bayesian-form estimation of model parameters and of the spatial uncertainties in the resulting parameters. The calibration of the robot workcell consists of several phases: hand-eye calibration, localization of the work object and estimation of model parameters for the work object surface. After localizing the work object, a finalization task can be carried out, e.g. inspection, manufacturing or assembly. A synthesis method of sensing planning that uses the same form of modelling spatial uncertainties is also presented. The deviation between covariance propagation models and actual systems is reduced by using detailed noise models of the robot system, including measured noise, at different phases in the calibration.

The methods developed here were tested with simulation and extensive actual tests in each phase. The evaluation criteria used were eigenvalues in the directions of eigenvectors of the error covariance matrix. A careful analysis of

spatial uncertainties was carried out to test the reliability of the covariance propagation method when the level of noise is changing, the results suggesting that the method is also applicable in such cases. The sensing planning method was compared with different types of sets of samples and the results analysed by considering the a posteriori error covariance matrix for the estimated parameters.

Preface

This work was carried out at the Technical Research Centre of Finland (VTT) and the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan, during the years 1998–2003. The related projects were financed by the Technology Development Centre of Finland (Tekes), VTT, the Finnish Work Environment Fund (TSR) and the companies participating in these projects.

I would like to express my gratitude to my supervisor, Docent Tapio Heikkilä, for his guidance, discussions and support during all these years. I especially thank him for his enthusiastic and encouraging attitude towards the world of robotics.

I wish to thank Prof. Aarne Halme and Dr. Hannu Lehtinen for reviewing this thesis, and I am also grateful to Mr. Markku Järviluoma, Prof. Heikki Ailisto and Prof. Pentti Vähä for their comments on its draft version. I would like to thank Dr. Fumiaki Tomita of AIST for allowing me to work in his group and Mr. Toshio Matsushita for his guidance and for providing a good working atmosphere in the 3D Vision Systems Group.

I am also grateful to my co-workers in the Mechatronics Group who contributed to this work. Thanks are also due to Mr. Matti Sirviö and Mr. Jukka Väinölä of VTT Industrial Systems for our fruitful co-operation during these years, and to Malcolm Hicks for revising the English text of this manuscript.

The work was supported financially by the Technology Development Centre of Finland, VTT, TSR, the Tauno Tönning Foundation and the Foundation for Technology in Finland (TES). Their support is gratefully acknowledged.

Finally I would like to express my thanks to my wife Kati and my daughter Siiri for their patience and understanding during my studies.

Oulu, the 22th September 2003

Mikko Sallinen

Contents

Abstract 3

Preface 5

List of symbols 9

1. Introduction 11
   - 1.1 Background 11
   - 1.2 Scope of the thesis 14
   - 1.3 Contribution of the thesis 18
   - 1.4 Outline of the thesis 20

2. Overview of parameter estimation methods for use with robot workcells 22
   - 2.1 Introduction 22
     - 2.1.1 Modelling the pose parameters 23
     - 2.1.2 Deterministic and stochastic estimation 24
     - 2.1.3 Estimating the pose parameters 25
   - 2.2 Hand-eye calibration methods 26
     - 2.2.1 Methods based on deterministic estimation 26
     - 2.2.2 Methods based on stochastic estimation 31
   - 2.3 Work object localization methods 34
     - 2.3.1 Methods based on deterministic estimation 35
     - 2.3.2 Methods based on stochastic estimation 37
   - 2.4 Work object surfaces 38
     - 2.4.1 Parametrized surfaces 38
     - 2.4.2 Free-form surfaces 39
   - 2.5 Discussion of the overview of parameter estimation methods for use with robot workcells 43

3. Modelling the spatial uncertainties in a robot system 45
   - 3.1 Introduction 45
   - 3.2 Methods for modelling spatial uncertainties 46
     - 3.2.1 Factors affecting spatial uncertainty 47
     - 3.2.2 Deterministic modelling 47
     - 3.2.3 Stochastic modelling 49

3.2.4 Simulation modelling 59
3.3 Combining estimates and uncertainties from different measurements … 60
3.3.1 Control strategies 61
3.3.2 Sensor fusion in different ranges 62
3.4 Discussion of the modelling the spatial uncertainties in a robot system 64

4. Planning of sensing actions 66
4.1 Introduction 66
4.2 Generate-and-test 68
4.2.1 Condition number 69
4.2.2 Observability index 71
4.3 Synthesis 74
4.3.1 Error covariance matrix 75
4.3.2 Surface properties 78
4.3.3 Occlusion 80
4.3.4 Other synthesis methods 80
4.4 Other sensing planning methods 83
4.4.1 Expert systems 83
4.4.2 Sensor simulation systems 84
4.5 Discussion of planning of sensing actions 85

5. A comprehensive framework for the modelling, estimation and planning of parameters and uncertainties in robot workcells 87
5.1 Introduction 87
5.2 Estimation of parameters in robot workcells 88
5.2.1 Hand-eye calibration 88
5.2.2 Work object localization 96
5.2.3 The robot base-wrist relation 100
5.2.4 Work object surfaces 104
5.3 Modelling and estimating the spatial uncertainties in a robot system 111
5.3.1 Modelling and transforming spatial uncertainties 111
5.3.2 Propagating the uncertainties into the TCP frame 115
5.3.3 Uncertainties in hand-eye calibration 118
5.3.4 Uncertainties in work object localization 120

5.3.5 Uncertainties in surface models 122
5.3.6 Combining estimates and uncertainties from different measurements 130
5.4 Planning of the measurements 134
5.4.1 Error covariance matrix and SNR 134
5.4.2 Verification of the planning criterion 143
5.4.3 Planning the measurements for work object localization 149
5.5 Discussion a comprehensive framework for the modelling, estimation and planning of parameters in robot workcells 155

6. Experimental tests 159
6.1 Uncertainties in simple examples 160
6.1.1 Estimating the model parameters 160
6.1.2 Uncertainties in the estimation of the pose of a cube 164
6.2 Uncertainties in robot geometry 169
6.3 Uncertainties in hand-eye calibration 170
6.4 Uncertainties in work object localization 174
6.5 Uncertainties in work object surface models 181
6.5.1 Parametric surfaces 181
6.5.2 Patched surfaces 187
6.6 Sensing planning 191
6.6.1 Planning measurement of the localization of a square 191
6.6.2 Calculation of the pose of a cube 195
6.7 Other experimental tests 198
6.8 Discussion of the experimental tests 199

7. Discussion 201

8. Summary 205

References 207

List of symbols

e	error function
H	homogenous pose matrix including rotation matrix and translation vector
ΔH	homogenous increment matrix for the pose matrix
J	jacobian matrix
K	partial derivative of error function with respect to input parameters
m	pose parameter vector
Δm	increment for pose parameter vector
O	observabilty index
P	error covariance matrix
R	noise matrix for the input data
T	translation vector
V	Rotation matrix
Q	weight matrix
δ	gaussian form noise in pose parameters
σ	singular value
θ t	rotation of the joint t

φ	transformation of each joint in D-H parameters
λ	eigenvalue of the parameter
Ψ	planned set of samples
$(t_x, t_y, t_z, \phi_x, \phi_y, \phi_z)$	full pose parameters (translation and orientation)

A, B and C	forms of spatial uncertainties in coordinate transformations
CAD	Computer Aided Design
D-H	Denavit-Hartenberg, a notation of method for coordinate transformations
DOF	Degrees Of Freedom
EKF	Extended Kalman Filter
GA	Genetic Algorithm
KF	Kalman Filter
Monte Carlo	simulation method based on generation of random numbers
MCL	Monte Carlo Localization
pose	posture, translation and orientation information
SA	Simulated Annealing
SNR	Signal-to-Noise Ratio
STL	Stereolithography exchange format
TCP	Tool Center Point

1. Introduction

1.1 Background

There is an increasing need for verifying the accuracy and reliability of the kinematic properties of industrial robot systems. Due to improvements in the performance of microcomputers, robot manufacturers are able to design controllers which have better performance in computing robot paths more accurately than before. This combined with higher accuracy in the manufactured parts of the manipulator has meant that the whole geometric system of an industrial robot has taken several steps forward during the last few years, and the result is the development of accurate, powerful robots suitable for many purposes (ABB Robotics 2003a, KUKA Robotics 2003). Analysis of geometric relationships and estimation of spatial uncertainties is nevertheless a fundamental problem in several robot-related application areas: product inspection, assembly, manufacturing, machining and mobile robotics. A desirable situation as far as flexibility and cost efficiency in a manufacturing line is concerned would be that the same robot should perform both manufacturing operations and quality inspection. In that way it would be possible to save process time, device investments and space on the factory floor. Improved accuracy increases costs, but costs can be held at a reasonable level when using a robot system. The constantly increasing off-line programming also places requirements on the accuracy and repeatability of a robot. Off-line models are nominal ones, but programs have to work in actual robot cells. This difference also has to be realized somehow. One solution is to analyze the spatial uncertainties off-line in order to verify that deviations between the nominal model and the actual workcell are within an acceptable range, determined as the working space in which the dimensions of the robot cell can vary but the robot programs still work.

Improvement of the accuracy, repeatability and performance of a robot system requires both external sensors and intelligence in the controller. Sensors help a robot to observe its environment and, using its intelligence, process the observed data and make decisions and changes to control its movements. The term *intelligent robotics,* or *sensor-based robotics,* is used for an approach of this kind. Such a robot system includes a manipulator (arm), a controller, external sensors and software for controlling the whole system. The movements of the

robot are controlled using a closed loop system. For this to be successful, the bandwith of the control sensors has to be much greater than that of the actuators of the joints, but usually the external sensors are still much less accurate than the internal sensors of the robot (Bernhardt & Albright 1993). The types of sensors that the robot uses for observing its environment may include vision, laser-range, ultrasonic or touch sensors. The availability of data varies between different sensors, and it is important when designing a robot system to consider what its requirements will be. The combining of information from several measurements or sensors is called *sensor fusion*.

The calibration of robot systems to improve accuracy can be divided into two main types:

- Calibration of static parameters, in which the aim is to estimate and improve the parameters used in the kinematic model of the robot. A typical application is static positioning in an assembly task.

- Calibration of dynamic parameters, in which the dynamic parameters of the robot model are estimated and improved. This improves the accuracy of the motion paths and other dynamic actions.

This thesis focuses on the first type of calibration, the goal being to estimate the transformations between different coordinate frames, evaluate the goodness of the transformations and reduce the uncertainties in the estimated parameters by planning the measurements such that the effects of noise, i.e. uncertainty in the system, are minimized.

There are always inaccuracies and flexibilities in the joints and links in a robot system, and improvement of robot performance in the sense of accuracy must include the following methods: calibration of robot joints and link lengths, calibration of the relationships between sensors and the robot base, calibration of the relationships between sensors and the robot TCP and visual servoing. Calibration of robot joints and link lengths means estimation of the offsets for robot joint angles and link lengths and using these for improving the kinematic model of the robot. This kind of calibration can be carried out using external sensors. When a sensor is located in a robot cell, the transformation between the sensor coordinate frame and the robot base has to be calibrated. This is called

hand-eye calibration. The sensor may also be attached to the TCP of the robot, so that the transformation from the robot TCP to the sensor coordinate frame has to be determined. Visual servoing means that the accuracy of the robot is improved by observing its location using an external vision sensor and moving the TCP of the robot on the basis of that visual information. The largest inaccuracies and errors in a robot system nevertheless depend on the use of the robot. Mechanical stress on the manipulator arm causes deviation in the TCP in the case of tasks with a high payload (Mooring et al. 1991). Robot manufacturers nowadays provide manipulator arms with as much as a 500 kg payload for this eventuality, but unfortunately they have poorer accuracy (ABB Robotics 2003a, KUKA Robotics 2003).

Calibration of the kinematic model of a robot includes the following three steps: modelling the kinematic behaviour of the robot, determining the parameters to be estimated and collecting the measurement data. Static calibration can be carried out in two ways depending on the measurement approach: first, a large number of points around the working space can be measured with a sensor having a high level of noise, or second, only a few, accurate and carefully selected points can be measured (Bernhardt & Albright 1993). This is because the accuracy and reliability of the calibration is highly dependent on the quality of the measurement data. There are two main methods for generating the set of measurement motions that will satisfy the requirements: generate-and-test and synthesis (Sheng et al. 2000). Generate-and-test is straightforward and easy to implement, but the disadvantage is the computational cost. This is critical only in real-time applications, however. Synthesis requires a deeper understanding of the relationships of parameter estimation to the goal to be achieved. In addition to these two methods, there are knowledge-based methods and sensor simulation methods which are not used so frequently in sensing planning.

When running a robot in an unknown evironment, i.e. treating objects whose geometric properties are not known, a method for representing surface models is needed. In addition to this, when trying to model all the possible error sources in a robot system, uncertainties in the surface models have to be taken into consideration. Another requirement for modelling is to model surfaces using simple forms which are easy to use in any calibration of the robot system.

There are several mathematical methods for calculating the parameters for calibrations. In many cases linearized models can be used, e.g. least-squares estimation. A very commonly used method is a Kalman filter (KF) or an Extended Kalman Filter (EKF). Some researchers have used a Levenberg-Marguard method (Motta & McMaster 1997) and obtained promising results. In order to improve the localization in a global frame, researchers have used a Monte Carlo Localization (MCL) in mobile robotics (Dellaert et al. 1999). The other branches of non-linear parameter estimation are Grid-based Markov Localization and Topological Markov Localization (Dellaert et al. 1999). Most of the parameter estimation methods enable the sensor information to be fused with the initial estimate after it has been computed, in order to reduce the uncertainties in the estimated parameters. Especially in mobile robotics, sensor fusion is important when the location of the robot is to be updated while it is moving.

Simulation and off-line programs offer a flexible approach for using a robot system efficiently. Nowadays product design is based on CAD models, which are also available for simulation and off-line programming purposes. When the robot is working, a new path can be designed and generated, but there is still a gap between the simulation model and an actual robot system, even if the dynamic properties of the robot are modelled in the simulation model. When using intelligent robots, this gap is bridged using sensor observations on the environment and motions are corrected according to sensor information from the control system. This kind of interaction improves the flexibility of the robot system and makes it cost-effective in small lot sizes as well.

1.2 Scope of the thesis

The purpose of this thesis is to present methods for estimating the geometrical relationships between coordinate frames and the spatial uncertainties in estimated model parameters. The calibration and running of a robot system requires three main steps:

- Planning and programming the robot motions
- Executing the motions and collecting measurement data with sensors
- Analysing the results.

Off-line programming includes calculation of the required set of motions (specific position and orientation values) and testing them in a simulation tool. This step is also called the *sensing planning and simulation step*. The second step, executing the actual calibration, includes making the measurements using the robot system. The calibration of geometrical transformations can be divided into two main tasks: hand-eye calibration, i.e. determining the transformation between the robot TCP and the origin of the sensor, and work object localizing, i.e. determining the relationship between the robot base frame and the origin of the work object. Analysis of the results includes calculation of spatial uncertainties attached to the estimated parameters or some specific feature and verifying that the accuracy requirements of the robot system will be fullfilled. The accuracy requirements for robot systems are typically between 0.1 and 0.8 mm.

The phases in the estimation of spatial relations in a robot-based workcell, as illustrated in Figure 1, may be taken to include hand-eye calibration, work object localization and a finalization task, which can be inspection, deburring etc. The initial input to the system is CAD information on the product to be handled. A priori information on the cell or environment should also be included. Using the sensor / sensors of the system, hand-eye calibration is first carried out, including the estimation of spatial uncertainties attached to the estimated parameters. After determining the hand-eye transformation, the location is measured, i.e. the uncertainties associated with the work object are estimated and a pose is calculated. Spatial uncertainties are also calculated in this phase, and then analyzed to check the fullfilment of the accuracy requirements. The last phase is the handling of the work object. This may include inspection, manufacturing operations, e.g. deburring or polishing, or assembly. These are typical applications of the robot system developed in this thesis.

Since the effects of noise and unmodelled sources of error can disturb parameter estimation, the quality of the coordinate transformations has to be verified somehow. The evaluation of system performance will be more reliable if efficient models are used to describe the spatial uncertainties, which are regarded in this thesis as affecting the estimated model parameters.

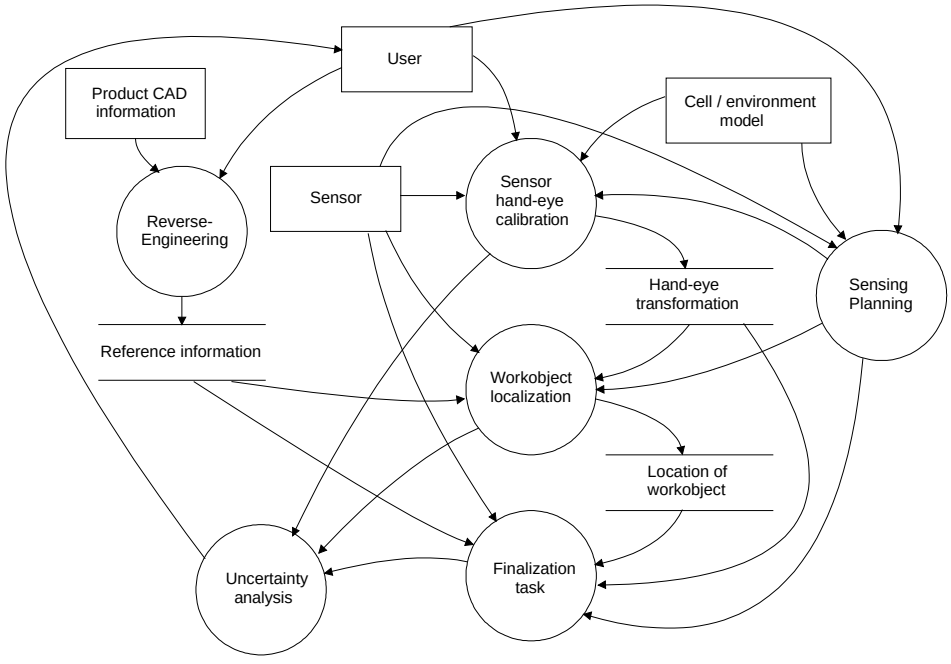


Figure 1. Process flow for the estimation of spatial relations in a robot-based workcell.

The spatial uncertanties are analysed by studying the error covariance of the estimated parameters a posteriori. When estimating the uncertainties of a pose with six degrees of freedom, the error covariance matrix includes both translation and orientation information. Computation of eigenvectors and eigenvalues from an error covariance matrix which includes several units does not give clearly interpretable results, but instead, the uncertainties attached to a single feature, e.g. a point, will allow computation of an error covariance matrix with single units. Other criteria for analysing the goodness of the set of samples are the condition number (Sakane et al. 1987), observability index (Borm & Menq 1991) and noise amplification index (Nahvi et al. 1996).

To obtain model parameters with a low level of uncertainties in the estimation process, the data set used for estimation will have to be carefully selected. This kind of data set can basically be generated in two ways: generate-and-test or synthesis. In the generate-and-test method, the parameter space of the sensor is discretized and candidates for different positions and orientations are generated.

The set of points is then evaluated somehow based on the given criteria. Building up a system of this kind is not very complicated, but if the parameter space is very large or the criteria do not sort the set of samples effectively, the result will be inaccurate. The other disadvantage of this method is that it ignores internal sensor parameters in the planning process and usually fails to take uncertainties into consideration (Gu et al. 1999).

The second method, synthesis, adopts an analytical approach to solving the problem, by calculating the valid sensor positions and orientations directly. Its best solutions allow the measurements to be planned so as to take internal sensor parameters into consideration. The disadvantage of the synthesis method is that if one tries to optimize too many parameters the result is an algorithm with poor performance. This thesis descibes an algorithm which belongs to the second category but without considering the internal parameters of the sensor. The method developed for this purpose is simple and easy to implement for application to different parameter estimation problems.

When estimating model parameters in work object localization and surface models, the Verification Vision approach (Shirai 1987) is considered in this thesis. This means that all the objects and their shape and form are known beforehand and no special attention is paid to recognition of an object by the sensor in advance.

One goal of this thesis has been to develop a flexible, CAD-based robot system. The requirements of flexibility are suitability for production of single or small lot sizes and changes of deviations in work objects. To fulfil the requirements of flexible operation of the system, all the robot motions are programmed off-line with interaction between an operator and the system. Various kinds of auxiliary tools have been developed for different types of motion, including selection of the calibration points, orientation and selection of the commands for the run-time sensor measurements. In addition to using CAD models produced by a designer, a Reverse Engineering approach is proposed here. In a case when there is no CAD model available, or the form of the work object differs considerably from that presupposed in the CAD model, a parametric or patched surface model can be generated from a cloud of measurement points. When using the same kind of estimation method as for coordinate transformations, it is possible to estimate the uncertainties attached to the model parameters.

The methods and algorithms presented in this thesis have been tested by simulation and in an actual test environment. In the simulation tests, the algorithms were implemented using MATLAB™ software and robot simulations were run using the ENVISION™ off-line programming tool. In the actual tests, several types of robots and sensors were used. The main tests were carried out using an industrial robot with six rotational joints and a 3D laser-range sensor.

No analysis of the computational efficency of the proposed methods has been included in this thesis. There are a number of real-time applications in the literature (Davison & Kita 2001, Dissanayake et al. 2001, Nilsson & Nygårds 1996) that use the Kalman Filter, for example, which is very similar to the method proposed here, and the methods presented here can be implemented in real-time applications as well.

1.3 Contribution of the thesis

The idea of developing methods for estimating coordinate transformations and spatial uncertainties in a robot system is not new, as the fundamental work of modelling spatial uncertainties was performed by Durrant-Whyte (1988), who defined the uncertainties based on Gaussian noise. Interest in robotics in the early 1990s focused on mobile robotics, and most of the research into the modelling of spatial uncertainties took place in that field. Hardly any attention was paid to the a posteriori analysis of error covariances, however, as presented in this thesis, even though methods for composing covariance matrices had been presented.

The contribution made by this thesis to methods for modelling spatial uncertainties in the context of a robot system are the following:

- Modelling the spatial uncertainties in a robot-based workcell covering a wide range of error sources. Several examples of modelling uncertainties in one task have been proposed in the literature using different modelling methods e.g. (Moreno et al. 2002, Su & Lee 1992, Taylor & Rajan 1988), but not in whole systems with accumulations of spatial uncertainties including several phases: hand-eye calibration, work object

localization, surface model generation and manufacturing operations such as surface inspection.

- Careful evaluation of covariance-based uncertainty approximations (Monte Carlo simulations, experiments with a real robot system). The spatial uncertainties are evaluated by studying the behaviour of an a posteriori error covariance matrix derived from the estimated parameters. The properties of the covariance matrix that are studied are both eigenvalues and eigenvectors, and also an uncertainty ellipsoid, which is one geometrical illustration of the covariance matrix.

- Analysis of the level of input noise affecting the a posteriori uncertainties of the estimated parameters. It is important in a noisy system to find the limits of performance within which it can be used, i.e. the limits within which the parameter estimation gives robust results. This analysis should be carried out in each estimation phase.

- A simple, novel sensing planning method for generating a set of samples that give a low level of uncertainties for the paramaters to be estimated. User of this method also enables the reliability of the estimator to be improved when the properties of the a posteriori error covariance matrix can be predicted within certain limits.

- Development of a flexible, CAD-based robot system. The system is programmed off-line, using simulation and CAD data as a priori information. The simulation model is updated off-line and movements of the robot are adjusted on-line based on sensor measurements from the environment. This kind of system is able to operate in manufacturing tasks with individual or short series production.

The main focus with regard to modelling spatial uncertainties is placed on studying the usability and reliability of the covariance propagation method when modelling the robot system. Linearized models are quite straightforward to use, and it is assumed that they remain reliable even if several calibration steps are taken (hand-eye calibration, work object localization and surface modelling). The usability and reliability are studied by comparing the difference in orientation and the lengths of the main axis of the uncertainty ellipsoid between

covariance propagation, Monte Carlo simulation (Karimäki 1993) and actual tests. In addition, the effect on the goodness of the set of samples is tested, to assess whether the difference between the uncertainties in the covariance propagation and actual tests decreases as the set of samples becomes better.

The methods developed in this thesis are designed in such a way that they should be easy to implement in the context of other estimation problems and to carry out even in a factory environment or remotely, using applications built on a web browser. They have been implemented in a different types of sensors, including a laser rangefinder, a tactile sensor, an optical range sensor and a camera system in a different projects.

1.4 Outline of the thesis

The chapters 2 and 3 are similar in structure, as they both include an overview of the literature regarding methods of hand-eye calibration, work object localization and the estimation of work object surface model parameters.

Chapter 2 presents methods for estimating the model parameters, together with approaches for solving the coordinate transformations required in hand-eye calibration, work object localization and the generation of surface models, as found in the literature. The estimation methods found in the literature are classified into deterministic and stochastic methods in order to illustrate the different approaches.

Methods for modelling the spatial uncertainties in a robot system are considered in chapter 3. A review of the methods used in the literature is presented, and this is again classified into deterministic and stochastic approaches. A brief overview of methods of sensor fusion used in robotics is also given.

Chapter 4 discusses sensing planning. Both main methods, i.e. generate-and-test and synthesis, are explained. In addition to these, expert systems and sensor simulation systems for sensing planning are introduced.

A methodological framework developed for parameter estimation, modelling of spatial uncertainties and the planning of measurements is given in chapter 5,

which focuses on presenting the contribution of this thesis to different phases of calibrating a flexible robot.

The results of the experimental tests are given in Chapter 6. First, a simple illustrative simulation example involving the estimation of a cube is presented, after which the results of the simulation and actual tests of hand-eye calibration, work object localization and surface model estimation are given.

The results and methods presented in the thesis are discussed in Chapter 7, and Chapter 8 states the conclusions and considers directions for future work.

2. Overview of parameter estimation methods for use with robot workcells

2.1 Introduction

One of the fundamental problems in robot automation is the requirement for generating a robot program off-line for a certain task, running it on several robots and obtain similar results with each. Usually this is not possible even if the robots are from the same manufacturer and should be similar. The reasons lie mainly in manufacturing deviations between manipulators and the performance of the robot controllers. To overcome these inaccuracies the kinematic model and coordinate transformations can be corrected in order to improve the accuracy of the robot systems.

The kinematic modelling of a robot can be divided into two levels (Mooring et al. 1991). First, corrections for joint level parameters are identified. These parameters include properties of encoders, signals and transducers and the lengths of the links, and they are usually updated directly in the controller of the robot. The second level determines the parameters for joint angles and the position of the end effector of the robot. These parameters can also describe homogeneous transformations from one coordinate frame to another. The most common means of modelling the transformations between robot links is the Denavit-Hartenberg method, which is also used here.

The computation of model parameters consists of the following steps: 1) determination of the system model and the parameters to be estimated, 2) retrieval of a priori information on the estimated parameters (this is not always required, depending on the method of calculation), and 3) calculation of the parameters. In the estimation of the parameters certain evaluation criteria are usually used to determine the optimal solution. The methods available for parameter estimation can be classified based on the type of model or assumption regarding noise (Mooring et al. 1991):

1. Deterministic or Stochastic

2. Recursive or Nonrecursive

3. Linear or Nonlinear.

The first criterion is divided, depending whether or not probabilistic models are used for system and noise models. The second criterion depends on the way in which the measurement data are used, which determines whether the whole body of data is collected at once or sequentially, with recursive computation. The third classification is based on the mathematical model that is used in modelling the system. A short overview of parameter estimation methods will be given in the next paragraphs, and ways of considering uncertainties when using different estimation methods will be introduced in Chapter 3.

2.1.1 Modelling the pose parameters

The full pose in a robot system modelled in a Cartesian space consists of six degrees of freedom, i.e. three translation and three orientation parameters. A state vector containing these parameters can be modelled in basically two different ways, deterministic and stochastic, as described above. The literature review contained in this thesis includes solutions to the existing problem that use both approaches to the modelling of parameters (where pose includes both translation and orientation parameters). In deterministic modelling, the variables are fixed constants that are assumed to be noise-free. In many cases this is enough, but if the level of system noise is high, the model may be inaccurate. A typical approach in the deterministic modelling is least-squares estimation.

In stochastic modelling, parameters are modelled in terms of their mean value and variance. Just as in deterministic modelling any parameter is assumed to be constant or of some exact value, each parameter in stochastic modelling has a mean and probability. The stochastic approach is closer to nature, because mathematical models are always approximations and no parameter can be determined absolutely (Maybeck 1979).

2.1.2 Deterministic and stochastic estimation

Typical methods for both deterministic and stochastic estimation are explained here. A typical deterministic example is least-squares estimation, the equations for which can be composed as follows. Let matrix J be the measurement, the vector m the parameter estimate and y the output of the system:

$$y = J^T m \tag{1}$$

Estimates for parameters m in equation (1) can be written as follows:

$$m = (JJ^T)^{-1} Jy \tag{2}$$

This equation (2) can be used recursively or non-recursively as classified in paragraph 2.1. A well known stochastic estimation method is Kalman filtering, the equations for which can be composed as follows. Let vector m at time *k* be the parameters to be estimated, while the matrix J includes the measurements, matrix $y(k)$ the output of the system, matrix $\delta(k)$ the input noise and matrix P the covariance of the measurements, as follows:

$$y(k) = J^T(k)m(k) + \delta(k) \tag{3}$$

The Kalman filter equations for parameter estimation are as follows:

$$L = \frac{P(k-1)J(k)}{r_2 + J^T(k)P(k-1)J(k)} \tag{4}$$

$$m(k) = m(k-1) + L(k)\left[y(k) - J^T(k)m(k-1)\right] \tag{5}$$

$$P(k) = P(k-1) - L(k)J^T(k)P(k-1) \tag{6}$$

where

L is the correction matrix

m is the estimate of the parameters

P is the covariance matrix of the estimated parameters

r_2 is the noise matrix of the measurement.

The result is an estimate of the state parameters m with its covariance P. The estimation methods will be classified in terms of the deterministic and stochastic approaches later in this chapter.

2.1.3 Estimating the pose parameters

The input and output data to be used in parameter estimation are illustrated in Figure 2. The estimated parameters include these of the homogeneous transformation from the base frame to the target frame. The result is a coordinate transformation and spatial uncertainties, i.e. the uncertainty type of that transformation. One exception concerns the estimation of surface model parameters of the work object, where no homogeneous transformation is provided, but instead there are model parameters which may include translation, orientation and parameters defining the shape or size of the object, e.g. the radius of a surface. Spatial uncertainties are uncertainties of the estimated parameters.

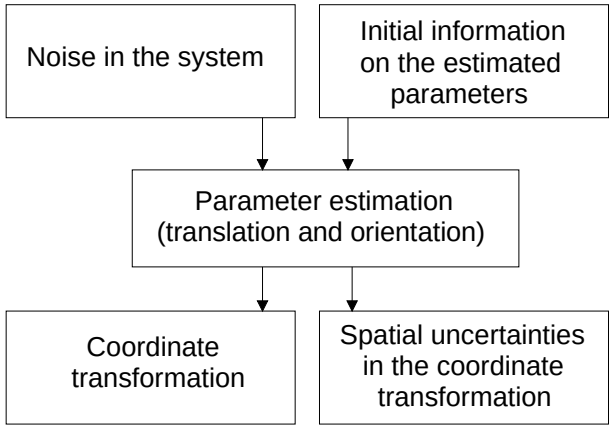


Figure 2. Estimation of the parameters.

When using robots equipped with external sensors for various measuring and manipulating tasks, the following three calibrations should be performed: calibration of the internal parameters of the sensor, hand-eye calibration and calibration of the link lengths and joint offsets of the robot. It is assumed in this thesis that the internal parameters of the sensor are already calibrated, and also the joint level parameters of the robot itself. Two main configurations can be given in hand-eye calibration: static cameras in the environment and on-the-hand cameras. In the case of static cameras, they are looking at a robot which is moving a calibration tool with reference points used in calibration. In on-the-hand systems the camera is attached to the TCP of the robot and it is used to measure a calibration object.

Estimation of model parameters is also required for work object localization and surface model generation. These two methods use the same estimation principle as hand-eye calibration and are presented later in this chapter.

2.2 Hand-eye calibration methods

The purpose of hand-eye calibration is to determine the coordinate transformation from the origin of the sensor to the TCP of the robot in on-the-hand situation. This transformation includes five to six parameters depending on the type of sensor, and is usually calculated by making movements with the robot and measuring the positions using a sensor. The transformation can then be calculated using the sensor information and information on the pose of the TCP of the robot. In following, methods for solving problem of hand-eye calibration is reviewed from the literature.

2.2.1 Methods based on deterministic estimation

A least-squares method can be used for calculating the parameters by fitting the measured set of samples into the parameters to be estimated. This is done by means of fitting criteria, i.e. an error function aimed at finding the best solution. Here two main solutions for hand-eye calibration using least-squares are presented: standard LSQ and recursive LSQ.

Standard linear least-squares estimation

The fundamental problem of hand-eye calibration is the transformation between the wrist of the robot and the sensor. This can be characterized by means of equation (7):

$$AX = XB \tag{7}$$

where

A is the change in position of the wrist of the robot

B is the change in the coordinate frame of the sensor

X is the transformation between the wrist of the robot and the coordinate frame of the sensor.

Both A and B are known, and X is unknown and has to be solved. The most fundamental work in hand-eye calibration has been carried out by Tsai & Lenz (1989), in a paper in which they propose a new method for calculating the hand-eye transformation with a much smaller number of unknown parameters, which is important especially in noisy cases. In this calibration the camera is attached to the wrist of the robot and robot makes a series of motions by measuring a calibration block with an array of target points on the top surface. The position of each target point on the calibration block is accurately known. The method is based on first defining the orientations of the hand-eye transformation in the form of a rotation axis and then solving the translation. The calculation is carried out by comparing the measurements using linear least squares estimation.

Other basic work for solving the homogeneous transformation equation (7) was contributed by Shiu & Ahmad (1989), who used an approach based on geometric interpretations of the eigenvalues and eigenvectors of a rotation matrix. Both translation and orientation values are calculated simultaneously using least-square fitting. Because no unique solution to equation (7) can be found with only one measurement, the parameters are estimated with two robot positions, but the orientations of the wrist cannot be zero or a multiplication of a π value, which

would guarantee a unique solution. The solution is verified with practical tests and the paper also includes a noise analysis of the estimated parameters by adding noise to the measurement and simulating the results. In conformity with the results illustrated in this thesis, Shiu & Ahmad state that noise is dependent on the configuration of the robot.

Hand-eye calibration of a stereo-camera system has been described by Takahashi & Tomita (1988), who propose a method for solving both external and internal parameters. They use a previously determined calibration pattern. Because the orientation parameters are the most critical and easiest to change, these are iteratively solved first, and after that the translation parameters, as done by Tsai and Lenz. The fitting criterion for calculating the camera location is a unique point measured by both of the cameras, and the solution is found using the least-squares method. The problem of parameter estimation is to measure the same point with both cameras even when the intrinsic parameters are not correct. These parameters are updated during the estimation process. To solve this problem the authors proposed to use segment matching to evaluate the correspondence, and also matching of vertices. These methods make it much easier to find the solution even in noisy situations.

A method for the hand-eye calibration of a vision sensor has been presented by Wang (1992), who proposes three cases of sensor calibration, each of which requires different information on the calibration objects and measurement positions. Class A requires the target object to be in a precalibrated location, class B requires a reference object with no precalibration and class C requires only a simple point in a non-calibrated location as the reference object. The classes presuppose different solutions for determining the transformation. A solution for class A can be found directly, using one measurement, whereas in class B calculation is a little more complicated and is resolved by first calculating the rotation matrix and then solving the translation vector using the least-square method. In class C the rotation matrix is also calculated first and after that a translation. The paper provides simulation and experimental results for all three classes. The experimental tests show that Wang's method and the Tsai-Lenz method have smaller standard deviations than Shiu-Ahmad's method. The method proposed in the paper provides a suitable solution for different needs and is a noteworthy contribution in that sense.

The first means of solving robot-world and hand-eye transformations simultaneously was formulated by Zhuang et al. (1994b) in terms of the following equation (8):

$$AX = YB \tag{8}$$

where

A is the change in position of the wrist of the robot,

B is the change in the coordinate frame of the sensor,

X is the transformation between wrist of the robot and the coordinate frame of the sensor,

Y is the transformation between the base frame of the robot and the world frame.

Both hand-eye and robot-world transformations may have to be determined frequently or rarely, depending on the application. It is always an advantage, however, if both calibrations can be carried out within a single set of measurements. The calibration is performed in the same way as proposed by Tsai, i.e. first solving the rotation part of the transformation and then the orientation part, both using linear models. Contrary to the original proposal, the orientation of the transformations is solved using unit quaternions, which means that a minimum of three measurements are required for determining the orientation and three more for translation. The method was tested under the effect of uniformly distributed noise added to the translation and orientation parameters of the measurements. In another test, noise was added to the joint values of the robot. The kinematic model of the robot was constructed using the Denavit-Hartenberg notation. These authors mention as one drawback of the method the two-stage computation, which means that errors from the first stage propagate into the second stage. In addition, the method seems to be quite sensitive to the completeness of the measurements.

Assembly robots used in the electronics industry have usually a reduced number of degrees of freedom. Typical examples of these are robots with a SCARA arm. A means of hand-eye calibration for such situations is proposed by Zhuang (1998), the basic solution being very similar to that used in cases of a full six degrees of freedom, although some differences do exist. The solution is divided into two phases, solving of the rotation matrix and then of the position vector. The types of motion can be divided into two forms, restricted motions and random motions. Both cases are solved using the least squares method.

Use of structure-from-motion combined with known robot motions has been illustrated by Andreff et al. (2001). Basically, the method is based on solving a slightly modified version of the homogenous equation (7), namely

$$A(\lambda)X = XB \tag{9}$$

where

λ is an unknown scale factor.

To solve this problem they propose a new equation which can solve the hand-eye transformation even in special cases, including pure translations, pure rotations or planar motions. The result of the parameter computation is fairly robust due to the structured motions. This makes the method very reliable.

Recursive least-squares

The recursive version of hand-eye calibration presented by Angeles et al. (2000) uses the least-square method on-line, which means that they compute parameter estimates using common recursive least-square estimation (RLS). They demonstrate that the method works well in on-line applications even when not optimized. The recursive manner also behaved quite well, and the accuracy of the estimates improved during estimation.

2.2.2 Methods based on stochastic estimation

Kalman filtering

A means for localizing the pose of a robot wrist using an ultrasound sensor has been presented by Wehn & Belanger (1997). Even though the ultrasound sensor is not usually very accurate, they succeeded in reducing the kinematic error of the robot system. They used a Kalman filter in the estimation and modelled the noise of the system, especially properties related to the ultrasound sensor, e.g turbulences and convection currents in the test room. When using the Kalman filter, they divided the estimation process into two separate subfilters, a reference filter which uses a fixed reference sender at a known location and a main filter which uses a moving sender attached to the robot's wrist, and estimated the position of the wrist. The idea is that information from the reference filter is fed to the main filter in order to improve the estimates of the position of the wrist. A more detailed description of the Kalman filter is given in section 3.2.3.

Gradient methods

The Newton method is an iterative method for solving a non-linear function which has a continuous derivative $\partial e/\partial m$ around the linearization area. After selecting initial quess (x_0, y_0, z_0) for the parameters to be estimated, the iteration runs by calculating a derivative around the zero. The intersection of the tangent of the curve with the given function gives the next quess (x_1, y_1, z_1), which are closer to the final estimate. In the next step, the derivative is calculated in this new location (x_1, y_1, z_1), and the intersection of the tangent with the function again gives a new estimate (x_2, y_2, z_2). This is continued until the tolerance ε small enough and desired accuracy is achieved. The eventual estimate (x_n, y_n, z_n) is taken as the result of the iteration (Kreyszig 1993).

Of the various non-linear methods, gradient methods are the most often used for parameter estimation in hand-eye calibration. In their description of the hand-eye calibration of a laser-range sensor attached to the hand of a robot Wei et al. (1998) formulate the problem using linear functions. The novelty in their work is that they include systematic error when building up the system model. The hand-

eye transformation is calculated using the Newton-Rhapson method in an iterative manner. The proposed method does not need any a priori information, but instead the algorithm first searches for the initial values itself. This is done by using only translational values and first calculating the orientation of the calibration plane. After that, the orientation of the sensor in the wrist of the robot is calculated, and finally the translation of the sensor. The method was tested with Monte Carlo simulations and an actual system with 320 tests, using 13 planes for calibration. The variances of the calculated parameters were also studied, and the results were promising.

Wei et al. (1998) also applied the proposed method to calibrating cameras in the wrist of the robot and proposed an approach in which the first initial calibration is carried out without any motion planning and the actual motion planning is carried out after that. The method is compared with that of Tsai and the rms error and maximum error are shown to be smaller than with Tsai's method. The drawback of the proposed method is the computational cost, which could not be reduced because of the need to estimate multiple object coordinates.

Horaud & Dornaika (1995) proposed an extension of Tsai's method for solving the hand-eye equation, equation (7). Their new form of that equation was

$$MY = M'YB \tag{10}$$

where

M is a 3x4 perspective matrix of first positions

M' is a 3x4 perspective matrix of second position.

The main advantage of this new method is that a pinhole model of camera is no longer needed and the method can then be applied to a laser stripe sensor, laser rangefinder etc. Adding to the method proposed by Tsai & Lenz, Horaud & Dornaika present two solutions for equation (10): first a linear solution in which the rotation is solved first and then the translation, and second, simultaneous estimation of the rotation and translation parameters using the non-linear Levenberg-Marquardt method. They tested the algorithms with simulations that

assumed uniform random noise in the first case and Gaussian–form noise in the translation and orientation parameters separately in the second case. The results showed that the non-linear method gave better performance for the translation vector, sometimes at the cost of the rotation vector. Overall, the non-linear method was more accurate than the linear method. Its disadvantage is that it needs quite accurate initial assumptions for the iteration.

The method for the calibration of a robot and a hand-mounted camera using a Gauss-Newton approach presented by Zhuang et al. (1995) involves three levels of calibration, first simultaneous calibration of the intenal parameters of the camera and its hand-eye transformations, second calibration of the internal parameters of camera, a hand-eye transformation and a base-world transformation, and third calibration of the entire robot in addition to the above. By comparing this multi-step calibration with simultaneous calibration, they were able to evaluate the error propagation between the different cases. The parameters were estimated by the Gauss-Newton method, which was shown to work more robustly than a Levenberg-Marquardt algorithm. Jacobian and estimation error vectors were composed for an iterative computation of the estimated parameters.

Following Zhuang's proposal for a linear solution for solving the hand-eye and robot-world transformations (Zhuang et al. 1994b), Dornaika & Horaud (1998) presented closed-form and non-linear solutions for the same transformations in workcell. The main differences between the linear and closed-loop methods are (Dornaika & Horaud 1998): 1) The linear method solves the components of quaternions linearly and then normalizes the quaternions so that they represent rotations, while in the case of a closed-form solution the unit quaternions are solved directly. 2) The linear method is not feasible for certain special configurations, whereas a closed-form system is. The advantages of non-linear methods are that the parameters, including translation and orientation, are estimated once, the method is not perturbed in the presence of noise as in the linear case, and both closed-loop and non-linear methods allow evaluation of the quality and confidence of the solution. For non-linear estimation they use the Levenberg-Marquardt minimization method. The results of the estimation of the transformation are equal to those obtained by hand-eye calibration, i.e. the non-linear method provided the most accurate solution for estimating the homogeneous transformation.

Other stochastic estimation methods for hand-eye calibration

Zhang & Ji (2001) used genetic algorithms (GA) for camera calibration. The advantage of their method compared with linear estimation methods is that they do not need an initial guess at the calibrated parameters, and accurate calibration can be carried out using only 7 measurement samples, even if both internal and external camera parameters are calibrated, whereas in linear models the minimum number of measured points is equal to number of parameters to be estimated. All three operations of genetic algorithms, evaluation, selection and recombination, are included in the generation of a solution space. The contribution of this work lies in the new mutation scheme provided, which comprises two steps: determination of the direction of the search, and determination of the step size in that direction. The selection of step size determines the amount of perturbation affecting a point generated in the given interval and has an effect on the convergence. Too small a perturbation will usually lead to sluggish convergence, but too large a perturbation may give erroneous or oscillating convergence. When selecting the position of a new current point in each generation, the authors use "golden selection" to control the direction of search. This improves the convergence of the GA considerably, leads to a close to optimal solution and simultaneously saves computation time. The algorithm was shown to be robust and accurate, especially in situations with a high noise level relative to that allowed for in Tsai's commonly used method.

2.3 Work object localization methods

Work object localization is based on fitting the measurements from the surface of the work object into the nominal model of the same surface. The work object to be localized may include surfaces of several forms, and some of these have to be selected for fitting.

The localization of work objects with different surface forms can be managed in basically two ways: each surface form has its own model, which is used in localization (e.g. a plane, cylinder, or B-spline surface), or else all the different forms are managed using patched surface forms. The advantage of the first method is that fitting is always effective even in the case of noisy data. The disadvantage is selection of a suitable surface form in each case, as this usually

requires a tool that will carry out that selection automatically. One advantage of modelling of all surfaces using patched surfaces is that no sorting is required between surface forms, but this kind of system is more sensitive to noise, and localization does not produce such accurate result as with separate surfaces.

Very similar methods are used in work object localization as in hand-eye calibration. An overview of parameter estimation techniques is presented below.

2.3.1 Methods based on deterministic estimation

Least-squares method for work object localization

A fundamental paper on determining forms of represention for surfaces and locating work objects is that of Faugeras and Hebert (1986). Where representation is traditionally divided into two main topics: hierarchical representation, in which several resolutions are used, and homogeneous representation, in which only one resolution is used, they use homogeneous representation. They model surfaces by deviding them into several subfeatures, which include a plane, quadrics and a B-spline surface. This is done even when the surface has a complicated form, e.g. a B-spline surface. The method has several advantages: 1) it is robust with respect to occlusions, so that a planar surface can be found even if it is partly hidden, 2) planar or quadric surface models are fairly robust in terms of the location of measuring points when measured from the surface, i.e. it does not matter where the point is located on the plane surface, and 3) it is quite easy to choose the type of surface representation for simple surface models. Objects are localized using recursive least-squares estimation in which the distance between the reference surface and the point to be measured consists of error functions, i.e. the distance is to be minimized. When estimating the rotation parameters Faugeras & Hebert (1986) use both an orthonormal matrix and quaternions.

The principle used by Gunnarsson and Prinz (1987) for managing complicated surfaces, including B-spline surfaces, in their fundamental work on localization was to linearize curved surfaces locally, i.e. to assume that a curve or surface consists of small planes. The surface is fitted by calculating the smallest distances from points measured to a plane approximation of each surface

segment. The calculation is done in an iterative manner using the least-squares method. When estimating the position and orientation of work objects, they noticed that it is quite easy to determine translation parameters but orientation requires rather complicated surface forms, i.e. high gradients. This was also noted by Sallinen and Heikkilä (2000b), when localizing the very flat surface of a ship's propeller, in that the estimation easily converged to a local minimum instead of a global one.

A CAD-based method of surface profile measurement has been studied by Menq et al. (1992), who propose a solution for surface inspection by first determining the accurate location of the surface to be inspected and then measuring the reference points from the surface. They were able to localize complicated surfaces based on minimizing the squared sum of distances between measured points and reference points. Before the actual estimation, the optimal points were identified in order to speed up the localization process. To describe the goodness of the set of samples, they propose a sensitivity measure which is a product of normal vectors and transformation errors, which are equal to dimensional errors. They use the standard deviation of the measured surface points for assessing the tolerances of the surface to be inspected.

To avoid the requirement of an a priori initial estimation, Boley et al. (1996) proposed the use of a Recursive Total Least Squares method, computing the TLS solution by ULV Decomposition, which is much faster than the frequently used Singular Value Decomposition method and can also be easily updated when new data are received by the system. Comparison of the proposed algorithm with the Kalman filter pointed to promising results.

Other deterministic methods for work object localization

Localization of free-formed objects using an iterative closest point (ICP) algorithm based on the Newton method has been proposed by Besl and McKay (1992). The method relies on minimizing the distance between measured points and a surface. It can be applied to different forms of surfaces and is robust with respect to noise within a reasonable level. The disadvantages, however, are easy convergence to a local minimum and failure of the computation when complicated object forms are used.

The method proposed by Besl and McKay was also used by Schutz and Hugli (1995), who improved the ICP algorithm by reducing the time required for matching. They took a number of measurements and selected the best ones by evaluating the quality of matching. Quality was computed from the squared distances between the closest points.

Localization of work objects, including commonly used symmetric data features such as planes, cylinders and spheres, has also been discussed by Gou et al. (1998). The localization is identified with a homogeneous space in a Euclidean group, and mimization is achieved using a Tangent algorithm or Hong-Tan algorithms. The authors expanded their work to cover all three cases of localizing work objects: general localization of 3-D objects, localization of symmetric work objects and localization of partially machined workobjects. Hybrid localization / localization of partially machined objects focuses on the localization of objects with unmachined surfaces and methods for extracting points measured from unmachined surfaces. The symmetric localization problem includes localization of objects with symmetric geometries, while general 3-D localization is for typical normal cases. Illustrations were provided of results obtained with all three proposed methods.

2.3.2 Methods based on stochastic estimation

A simple method for localizing a polyhedral object using light stripes has been proposed by Kemmotsu and Kanade (1995). This includes three identical laser rangefinders, and the pose of the object is computed by minimizing the distance between points on the surface and a reference model of the object by minimizing the sum of the integrals of the squared distances along each line segment over all pairings of a feasible interpretation. The uncertainty of a pose is determined as a covariance matrix of points at the ends of each stripe. The noise in the system includes uncertainties coming from the sensing tool as well as noise from inaccuracies in the object to be localized.

Other work using stochastic estimation in work object localization has been carried out by Sanderson (1999) and in localization of a mobile robot by Nygårds and Wernersson (1998). These papers are explained more detailed in chapter 3.

2.4 Work object surfaces

Surface models can be divided into two main forms: parametric and patched surfaces. The model parameters of parametric surfaces can be estimated by the same methods as presented for hand-eye calibration and work object localization. Patched surfaces are non-linear surfaces, e.g. spline surfaces. Due to their more precise description of the surface, parametric surfaces are more reliable for use in object localization than patched surfaces. The modelling of both forms is illustrated in this section and equations are presented for computation of the model parameters.

Surfaces are divided by Faugeras and Hebert (1986) into forms of representation, which they call primitives. These include planes, quadrics and B-spline surface patches, with a residue of more complicated surfaces. In the generation of models, they first divide the surface into smaller regions and repeat the division until the required accuracy is achieved. This also works in reverse, which means that small regions can be merged into larger areas when such a high level of accuracy is not required. This is very useful when finding the locations of a workobject surface for use in localization, as only the surface of a restricted region needs to be modelled accurately.

Three types of surface form for work objects are considered by Li et al. (1998), namely 1) a general form of a surface, 2) a symmetric, i.e. parametric, surface, and 3) a surface with unfinished areas. The surfaces in the first two cases are very close to a CAD model, or at least the deviations between the reference model (design model) and the actual model (model of the real work object) are very small. In the third case the surface of the work object is divided into parts representing finished and unfinished surfaces and the unfinished surfaces are constrained out in localization. This type of modelling is very important in manufacturing applications, for even though efforts are made to localize work objects using accurate surfaces, it is not always possible.

2.4.1 Parametrized surfaces

An efficient method for modelling parametric surfaces is presented by Ahn et al. (2002), who estimate different surface forms (plane, cylinder, sphere, cone or

torus) from a point cloud using the Gauss-Newton method, taking the shortest distance as a fitting criterion. The main goal of their work was to develop a system that automatically generates the right surface forms. The current state of the method is that the user interactively identifies points on different surfaces of the CAD model and the algorithm does the segmentation and surface model fitting. This is a good step, as mentioned previously, as one of the most critical tasks in the use of multiple surface forms for work object localization or the estimation of surface model parameters is to separate the surfaces from the set of samples measured.

2.4.2 Free-form surfaces

When surfaces are more complicated, they have to be modelled using patched surfaces, because a parametric surface would not provide sufficient accuracy. Patched models are more complicated than those for parametric surfaces, and several types of surface models have been developed for such purposes, including triangular facets, B-splines and Bezier curves. The differences between these surface types lie mainly in the representation of the surface. B-spline surfaces are patched and have some control points and/or knots that define their form, the representation of this patch structure varying from one method to another.

Surface modelling using Bezier curves

A Bezier curve has a set of control points describing its form. Surfaces are then regarded as consisting of several curves with a given mutual density.

The modelling of surfaces using Bezier curves has been explained by Knopf & Abouhossein (2000). The surface model is generated based on a cloud of points and the control points are decribed using a Bernstein Basis Function network that gives weights for the Bezier functions. The advantage of Bernstein Basis Functions is that they can be used directly for modelling the polynomials used for describing Bezier curves.

Surface modelling using B-Spline surfaces

The modelling of surfaces using B-Spline surfaces also calls for the determination of control points on the basis of a set of points on the surface. A typical algorithm for generating such a surface has been presented by Liu and Hasegawa (2001). This is illustrated in Figure 3:

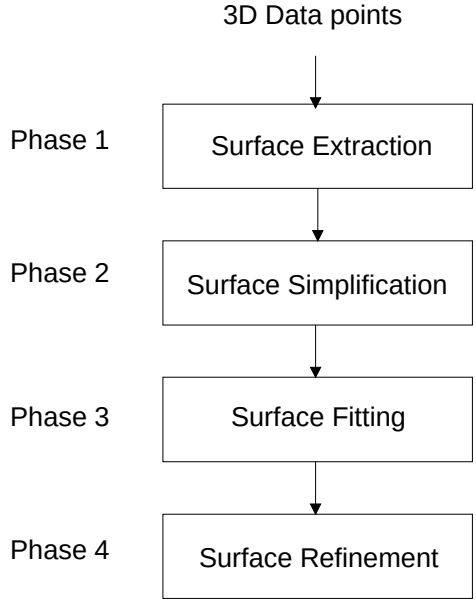


Figure 3. Flow chart for the generation of a B-spline surface, according to Liu & Hasegawa (2001).

The first phase, surface extraction, includes the generation of a large mesh of vertices that may become parts of the B-spline surface. In the second phase the surface is simplified by reducing the number of facets in the mesh. In the third phase, a B-spline surface is generated by finding control points that describe it. In the fourth phase, surface refinement, the surface model is scanned through and checked for the required accuracy. If this is not satisfied, the surface is divided into smaller parts so that the requirements will be fullfilled.

B-spline surfaces have been used by Huang and Cohen (1993) in an approach in which cubic polynomials are calculated for describing the surface. A transformation of a curve is calculated by using moments of the curve, which means that the sizes of the changes in the surface profiles are considered. In this way the pose of each surface can be calculated.

Gunnarsson and Prinz (1987) also modelled complicated surfaces using a B-spline surface patch and found it to be simple and usable for localization purposes.

Charlebois et al. (1997) modelled different kinds of parametric surfaces using B-spline surfaces and analyzed the goodness of the fit of the set of points to the surface in each case. They used two kinds of criteria in the evaluation, the first describing the curvedness of the patch and the second its shape. Uncertainties in surfaces can be evaluated by calculating the means and standard deviations for these parameters. They tested the criteria for plane, cylinder and sphere surfaces with an actual robot system and were able to evaluate the surfaces successfully. The tests showed that a B-spline surface patch can be used for modelling parametric surfaces, but it is obvious that the computation requirement for surface model generation is much larger than when using parametric representation directly.

Modelling of the whole environment of a mobile robot using B-spline surfaces has been described by Drapikowski and Nowakowski (2002), who also showed that all forms can be modelled using only B-spline surfaces.

Surface modelling using triangular facets

A popular form of modelling for free-form surfaces is triangular faceting, in which the surface consists of facets connected to each other. For flexible use, the accuracy of the CAD model has to be controlled somehow for each purpose. In order to be able to use this approach in robotics, there is no time for heavy computation. A flexible Reverse-Engineering approach with user-controlled density has been proposed by Chen et al. (1999). The facet model is generated by calculating the surface normal using the cross-product of two vectors calculated from three points and normalising the result. They propose two

approaches to data reduction: 1) by reducing the facets based on small differences between the normal vectors, and 2) reducing the number of triangles by means of a kind of filter which converts the regions with small triangles into larger triangles. Both of these methods also work in reverse, which is necessary when accurate information is required in some particular area.

The modelling of free-form surfaces in a robot environment has also been used by Sallinen et al. (2001), who measured the surface profile of a work object using a laser rangefinder and generated a facet model using an octree method, the advantages of which are effective representation in an unevenly distributed point cloud and the ability to represent very complicated forms.

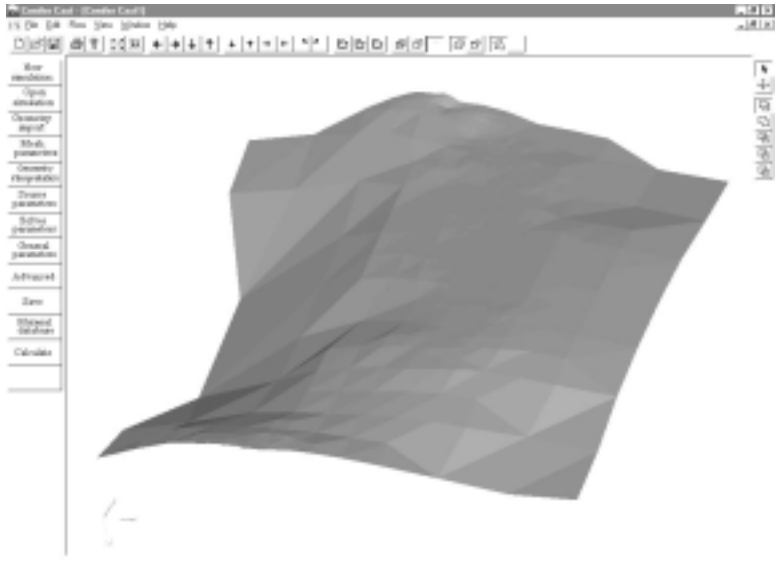


Figure 4. A facet model for the surface of a work object (Sallinen et al. 2001).

A typical surface modelled using triangular facets is illustrated in Figure 4. Basically, a facet surface can be used for work object localization by calculating the shortest distance from the set of measured points to the reference surface using the same point-to-point fitting as presented for the localization of a B-spline surface. Instead of searching for the closest point, one will be searching for the closest facet.

2.5 Discussion of the overview of parameter estimation methods for use with robot workcells

The estimation of coordinate transformations in a robotic system is essential in order to be able to carry out tasks that require high accuracy. The methods for doing this can be divided mainly into deterministic and stochastic approach for the state model and linear and non-linear cases depending the mathematical model used in modelling the system. Deterministic models does not include noise in the models whereas in stochastic models noise is modelled in the system. The advantages of linear methods is that they are quite straightforward to use and can be employed easily in many applications. A typical example of a linear model is standard least-squares estimation. The disadvantage compared with the non-linear methods is sensitivity to noise (e.g. Newton-Rhapson). Both linear and non-linear models require a priori information on the parameters to be estimated, which may in some cases be difficult to obtain. In addition to these methods, genetic algorithms (GA) have been used to determine the parameters for coordinate transformation. The advantage of these is that calculation does not need any a priori information and the number of set of samples required is fairly small. This is due to their principle of operation.

There are several methods for hand-eye calibration described in the literature in addition to the classical Tsai method. Several authors propose simultaneous calibration of the sensor to TCP transformation and the location of the robot in the world frame using linear models (e.g. Zhuang et al. 1994b) and non-linear models (e.g. Dornaika & Horaud 1998, Heikkilä et al. 1999). This will improve flexibility and verify the performance of the robot system if calibration is carried out frequently enough.

Localization of the pose of the work object depends on the surface form of the object. Its surfaces can be divided into parametric and patched surfaces, where the parametric surface forms considered here are planar, cylinderical and spherical and the patched surfaces considered are B-spline surfaces. The localization of parametric surfaces is more straightforward than that of work objects with patched surfaces. On the other hand, patched surfaces provide a much larger variety of surface forms. When using parametric surfaces, the feature selection that follows data acquisition plays an important role, although this task is not considered here. The localization of work objects with parametric

surfaces gives a more accurate result, however, due to the form of the surface model. Problems are often encountered in estimating the orientation parameters of objects with patched surfaces, because flat surfaces cause large orientation errors even with a low level of noise.

Input data for surface generation can be obtained directly from the CAD model or a Reverse Engineering approach. Another form of modelling surfaces in addition to the two methods mentioned above is the triangular facet method. This gives a flexible approach to modelling even in the case of complicated surfaces.

3. Modelling the spatial uncertainties in a robot system

3.1 Introduction

Methods for modelling the spatial uncertainties in a robot system are closely related to the parameter estimation problem presented in the previous chapter. Forms of modelling the uncertainties can be divided into the three classes:

- Stochastic models
- Deterministic models
- Simulation models

It is assumed in stochastic models that the noise in the initial state and during motion and all the measurement models are distributed according to some known distribution. When considering the use of stochastic models used in parameter estimation, the most common stochastic-based estimator used in linear models is the classical Kalman filter (Maybeck 1979), there being numerous examples of the use of this approach in the literature (Su & Lee 1992, Nygårds & Wernersson 1998, Sanderson 1999). The Kalman filter gives a robust and efficient method for estimating both parameters and their respective uncertainties, and it can be used recursively, i.e. to predict the state of the next cycle by using the last estimate and a new measurement to produce new values for the state parameters. This recursive behaviour improves the speed and computational requirements of the method (Moreno et al. 2002). The advantage of covariance propagation methods is the relatively simply way of combining the existing uncertainties between coordinate frames and error sources. On the other hand, the disadvantage is that if the actual noise models are not Gaussian-distributed the covariance propagation gives a rather biased result. Due to the demand for modelling more complex systems, there are several modifications of the Kalman filter, including an Extended Kalman filter developed for cases where the system model has non-linear dynamics or a non-linear pattern of observations.

In addition to the Kalman filter, there are Bayesian forms of modelling which apply when using a Gaussian form of noise distribution. The main user of Bayesian-form modelling is Durrant-Whyte (1988). There are many similarities between the Kalman filter and the Bayesian form of modelling, but there is a difference in the presentation of probabilities.

Deterministic modelling gives a fairly straightforward way of modelling spatial uncertainties. The two main methods are tolerance propagation (Taylor & Rajan 1988) and grid-based modelling / dynamic Markov localization (Burgard et al. 1998). In these methods a solution space is divided into deterministic elements, polytopes in tolerance propagation and a probability density map in grid-based modelling.

There are several search algorithms which uses simulation methods or known as sampling-based methods for modelling spatial uncertainties. The most common method used in robotics is Monte Carlo localization, the idea of which is to present theoretical probability distributions in the state space by simulated random measurements (Moreno et al. 2002). The probability function itself is not described (Dellaert et al. 1999).

The second type of simulation methods consists of genetic algorithms, which presuppose that the uncertainties are considered to be stochastic. Due to the high computational cost, such algorithms are usually calculated off-line, although examples of on-line calculation can be found in the literature, e.g. Moreno et al. (2002), who improves the performance of localization by combining the Kalman filter with genetic algorithms.

3.2 Methods for modelling spatial uncertainties

The main methods for modelling the spatial uncertainties are described in detail in this chapter, together with a review of the literature. The methods are divided into stochastic, deterministic and simulation models, and each subtopic is presented under these headings. Before the methods are presented, the factors affecting spatial uncertainty will be described.

3.2.1 Factors affecting spatial uncertainty

Spatial uncertainty is affected by noise, which can be divided into geometric and non-geometric errors (Mooring et al. 1991). Geometric errors are errors in parameters that define the geometric transformations in the robot system, and can arise from the following sources: resolution of the sensors at the joints in the robot, resolution of the measuring sensor (e.g. laser rangefinder) and manufacturing accuracy of the manipulator, i.e. deviations in the lengths of the links. Non-geometric errors are computational errors made by the robot controller when carrying a heavy payload, for example. When calibrating the robot system, i.e. estimating the spatial uncertainties of the transformations in different coordinate frames, geometric errors pay a major role and non-geometric errors a minor one.

The behaviour of geometric errors can be described in terms of homogeneous distribution and continuous existence. A Gaussian distribution is a suitable assumption for modelling such noise, but the effect or amplification may change depending on the parameter, so that an ideal Gaussian distribution does not give a good enough solution. Also, some error sources are not uniformly distributed and are better represented in deterministic modelling.

3.2.2 Deterministic modelling

Tolerance propagation

The use of multidimensional linear programming, as proposed by Taylor and Rajan (1988), represents a different approach to modelling the uncertainties in a robot system. The advantages of the method are that it is a straightforward way of modelling non-probabilistic constraints and that it does not require knowledge of the probability distributions, because redundant sensor information does not contribute to the reduction of uncertainty. The computation method gives more conservative results than methods based on a Gaussian distribution, and it can also be easily applied to geometric problems in order to solve proper configurations. The uncertainty spaces are solved for areas called polytopes, which are equal to tolerance spaces. To simplify the interpretation of high-

dimensional polytopes, they are projected to a lower dimensional space such as a 2D or 3D space.

The idea emphasized by Taylor & Rajan is illustrated in Figure 5, where the orientation of the inner square is determined to be within the given segment α, and the locations of the corners c_1, c_2, c_3 and c_4 within certain limits. The determination of uncertainty limits by this method is a quite simple and still effective way of presenting uncertainties.

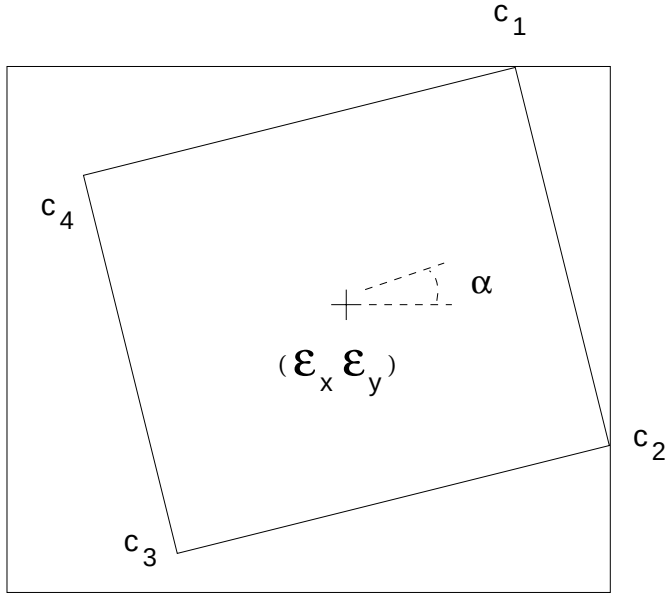


Figure 5. Constraint of a small part within a feed tray, according to Taylor & Rajan (1988).

Grid-based modelling / Dynamic Markov Localization

The basic idea of Markov Localization is to maintain a probability density over the whole state space of the estimated parameters (Burgard et al. 1998). This probability density grid map is updated when new information is obtained e.g.

on coordinate relations. The dynamic properties of the method are expressed when the resolution of the probabilty space is dynamically enhanced. The new information is integrated into the state space using a Bayesian update formula. To manipulate the probability space efficiently, they use an octree principle for representing the probabilities. The advantages of the method are that the space requirement is adopted automatically and the resolution of the state space can be increased or reduced dynamically.

Grid–based modelling has also been used by Olson for estimating the location of a mobile robot (Olson 1999). He used probability distribution functions (PDF) to express the uncertainties. A scene called an occupancy grid is created to represent the forms of the environment of the robot and the respective uncertainties. He used the maximum likelihood method for the estimation itself.

3.2.3 Stochastic modelling

Gaussian distribution: Kalman filter

The Kalman filter is a well-known method for taking the model with following conditions into consideration when estimating parameters (Kalman 1960): error is zero-mean, white noise and error covariances are known. When estimating the transformations between coordinate frames, Smith and Cheeseman (1986) became the first to use a Kalman filter in robotics for modelling the spatial uncertainties between a nominal transformation and the expected error of the estimated parameters. They proposed a general model of coordinate transformations which could be generalized into a six degrees of freedom model, including the uncertainties attached to the translation and orientation parameters. The uncertainties in each coordinate frame were studied by considering the uncertainties of a point in that frame. Uncertainty is described in terms of an error covariance matrix for the estimated parameters, which geometrically takes the form of an uncertainty ellipsoid. The geometrical presentation is illustrated in Figure 6, where the coordinate transformations of the four coordinate frames $frame_1, frame_2, frame_3, frame_4$ with respect to the world are illustrated.

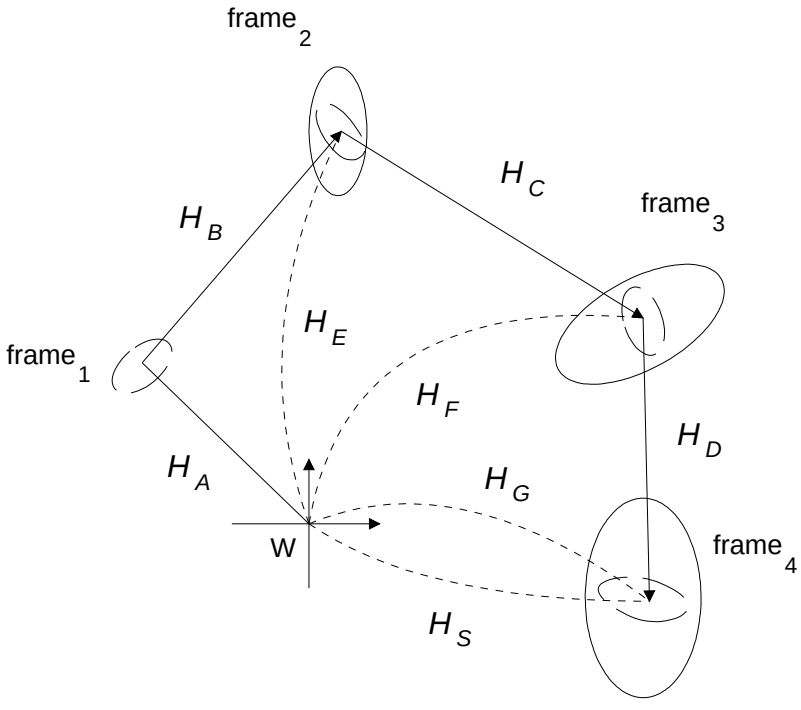


Figure 6. A sequence of approximate transformations, according to Smith and Cheeseman (1986).

Smith and Cheeseman also introduced two methods for manipulating spatial uncertainties and compounding and merging approximate transformations. By compounding they mean propagation of the uncertainties associated with different coordinate frames into a one combined uncertainty which represents the total uncertainty attached to the coordinate transformations and is always larger than the separate uncertainties. This is the typical case when calculating the uncertainties affecting a manipulator arm and determining the coordinate transformation using the forward geometrics of the arm. When combining two two-dimensional coordinate transformations $H_A\ (X_1, Y_1, \theta_1)$ and $H_B\ (X_2, Y_2, \theta_2)$ into a single transformation $H_C\ (X_3, Y_3, \theta_3)$, which is given in terms of H_A and H_B, the covariance of the transformation H_C must also be estimated. The given transformation is approximated using first-order Taylor series expansion around the means of the variables, and can be written in matrix form as follows:

$$\begin{bmatrix} \Delta X_3 \\ \Delta Y_3 \\ \Delta \theta_3 \end{bmatrix} \cong J\left[\Delta X_1 \quad \Delta Y_1 \quad \Delta \theta_1 \quad \Delta X_2 \quad \Delta Y_2 \quad \Delta \theta_2\right]^T \tag{11}$$

where

J is the (3x6) Jacobian matrix of the transformation around the mean values.

The covariance of the estimated parameters is defined as follows:

$$C_3 \cong J\begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix}J^T = HC_1H^T + KC_2K^T \tag{12}$$

where

C_1 is the covariance matrix of the coordinate frame H_A

C_2 is the covariance matrix of the coordinate frame H_B

H is the partial derivative of the coordinate frame H_A

K is the partial derivative of the coordinate frame H_B

The merging of coordinate transformations means that the uncertainties attached to some coordinate transformations are reduced by using information from other similar or parallel transformations. The initial situation for merging the transformations is that the covariances C_1 and C_2 of the individual transformations H_A and H_B must be known. When merging two transformations in which C_I is the one to be computed, the Kalman gain K has first to be calculated:

$$K = C_1\left[C_1 + C_2\right]^{-1} \tag{13}$$

The covariance of the merged coordinate transformations can then be calculated from the Kalman gain as follows:

$$C_3 = C_1 - KC_1 \tag{14}$$

and the merged mean value for the transformation as

$$X_3 = X_1 + K[X_2 - X_1] \tag{15}$$

Both compounding and merging methods need some initial assumptions in order to work reliably. First, the errors are assumed to be independent in both cases, and the noise in the parameters is also assumed to be zero-mean Gaussian distributed with no bias. In the case of merging, the coordinate transformations have to be in the same coordinate system. Also, when combining similar or parallel transformations of corresponding uncertainties one must be careful to select the appropriate weights for the linear combinations of the two estimates. If these assumptions are satisfied, the given formula should produce a reliable estimate.

The Kalman filter and Gaussian distribution of noise has been used for modelling spatial uncertainties by Su and Lee (1992), who present methods for manipulating uncertainties that include the fusion of uncertainties arising from different coordinate frames and the transforming uncertainties from one frame to another. They also apply the manipulation of uncertainties to an assembly task and develope methods for different kind of tasks: moving actions, perception actions and contact actions. A moving action consists of the movement of a robot arm from one location to another, which changes the uncertainties in the pose due the different configuration of the robot, while a perception task is a sensing task aimed at reducing the uncertainties in the world state by improving knowledge about it. They propose a solution for the consistency problem of defining the spatial relationships in the world state, and maintain that when a coordinate transformation gains more information and the original location of the coordinate frame is changed, this also changes other transformations in that frame. The principle is illustrated in Figure 7:

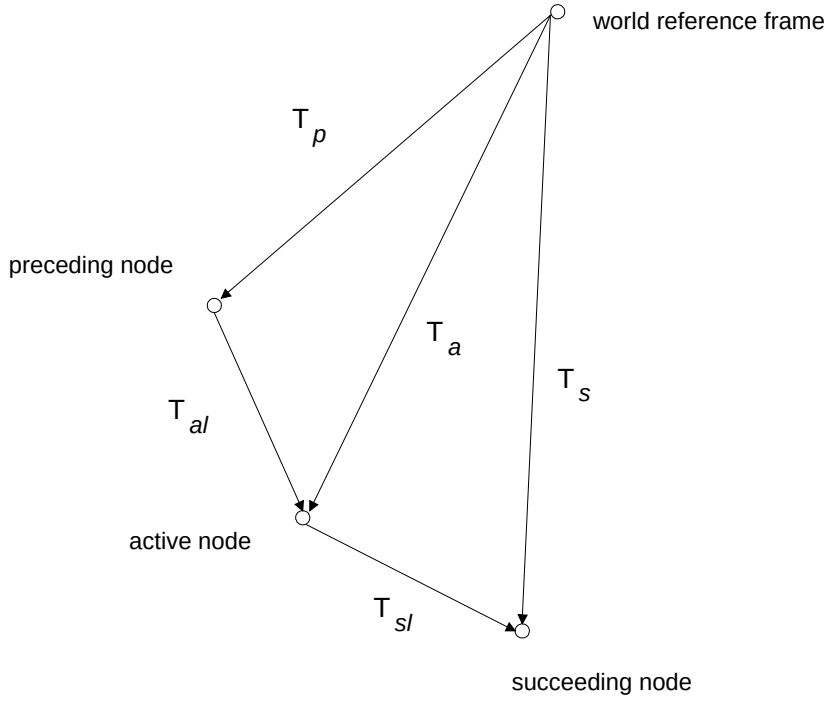


Figure 7. Coordinate transformations when uncertainties in the world state are updated, according to Su & Lee (1992).

The preceding node mentioned in the figure is the original frame before updating of the world state, the active node is the updated coordinate frame and the succeeding node is the new target frame, i.e. the frame which will be approached next. The third form of action is a contact action, in which they assume that there are no modelling errors in the surfaces, so that two objects become fully dependent in their location. When contact occurs in some directions of the surfaces but not all, uncertainties are reduced only in these parameters.

The Kalman filter has also been used together with other methods. The modelling of spatial uncertainties using the maximum likelihood method has been proposed by Sanderson (1999), who uses the Kalman filter for parameter estimation and also models the uncertainties of parts using a Gaussian distribution when assembling known parts using a robot. The main question with respect to this application is: even if the parts fit nominally with each other, what is the probability that they actually fit? He regards the uncertainties as

tolerances, because in practice designers specify the uncertainties as relations between two parts.

Gaussian-distributed covariance modelling for noise and uncertainties has been used by Nygårds and Wernersson (1998), who employ explicit covariance matrices and a Kalman filter for parameter estimation. They have applied their method to mobile robot scanning planes around the working space.

In cases where the coordinate systems are non-linear, better results can be achieved using the Extended Kalman Filter (EKF), which is a non-linear estimator. The difference between the Kalman filter and the Extended Kalman filter is that in EKF the linearization is performed around an estimated trajectory instead of a previously calculated nominal trajectory.

An iterative version of the Extended Kalman filter was proposed by Moreno et al. (2002), in a work in which they combined genetic algorithms (GA) with EKF, proposing three localization methods: first, generating the initial values using GA, second, making a genetic search for suitable values and selecting them using GAs, and third, updating the state using EKF. The proposed method combines the advantages of EKF and the parameter search used in GAs. They call the new method the Restricted Genetic Optimization Filter (RGO). The uncertainties are modelled as probabilities in GAs and then transformed into EKF, so that the uncertainties are reduced in the iteration and can be used in both GAs and EKF.

Gaussian distribution: Bayesian-form modelling

Another stochastic form of modelling is Bayesian modelling, which has been developed in the field of robotics by Durrant-Whyte. Bayesian-form modelling uses Gaussian featured assumption in the modelling of the noises. Durrant-Whyte started his work with mobile robotics, but the method has been applied to almost every field of robotics, including ground, underwater and flying robots. The fundamental work was reported in Durrant-Whyte (1988), where he introduced basic methods for modelling the spatial uncertainties of coordinate transformations using Bayesian-form uncertainty modelling.

According to Durrant-Whyte, the uncertain geometry includes two main elements: the geometry itself and the overall topology between the geometric structures. The geometry describes the geometric uncertainties of basic features such as points, curves and planes in a three-dimensional space, and the topology defines the probabilistic geometric transformations between these basic features. When the transformation of a parameter vector between coordinates *c* and *c'* is written $p'=h_g(p)$, the perturbed transformation can be written as follows:

$$p'+\delta p' = h_g(p+\delta p) \tag{16}$$

where

δp is the noise at the point

As in the case of Kalman filtering, equation (16) can be written using the Taylor series expansion:

$$\delta p' \approx J_p h\big|_p \delta p \tag{17}$$

where

$J_p h$ is the Jacobian matrix describing the change in δp viewed from different coordinate systems.

Durrant-Whyte also illustrates methods for transforming spatial uncertainties from one coordinate frame to another and mecanisms for the consistent interpretation of relations between uncertain features. If the transformation between coordinates i and j is ${}^j h_i$ and ${}^i\Lambda_p$ is the variance of the point p in the coordinate frame i, the variance in the coordinate frame j can be written

$${}^j\Lambda_p = \left(\frac{\partial {}^j h_i}{\partial p_i}\right) {}^i\Lambda_p \left(\frac{\partial {}^j h_i}{\partial p_i}\right) \tag{18}$$

This equation is valid provided that the matrix $\partial {}^j h_i / \partial p_i$ is not singular or ill-conditioned. The validity of this can be proved using verifying tools such as

condition number or computation of the eigenvalues of the error covariance matrix.

Bayesian modelling was also used for representing spatial uncertainties by Nakamura & Xu (1989), who proposed a geometrical fusion method for sensor information based on Bayesian probabilities. For the geometrical representation of uncertainties they proposed an uncertainty ellipsoid inside which the resulting parameter vector is located in parameter space (Figure 8). A similar uncertainty ellipsoid was presented earlier by Smith and Cheeseman (1986). The ellipsoid was calculated on the basis of an error covariance matrix determined in such a way that its eigenvectors are the principal axis of the ellipsoid and eigenvalues represent the lengths of the axis. As a criterion for describing the extent of the uncertainties attached to the estimated parameters they use the volume of the uncertainty ellipsoid, which can be written as follows:

$$\Psi_{ellipsoid} = \frac{\pi^{p/2}}{\Gamma(1+p/2)}\sqrt{\det(P)} \qquad (19)$$

where

p is the (x,y,z) point of the covariance to be calculated in the estimated coordinate frame

Γ is the gamma function

P is the covariance of the parameters.

Computation of the uncertainty ellipsoid as a criterion for evaluating the goodness of the set of samples has been also used by Heikkilä et al. (1999) and it is similarly used in this thesis.

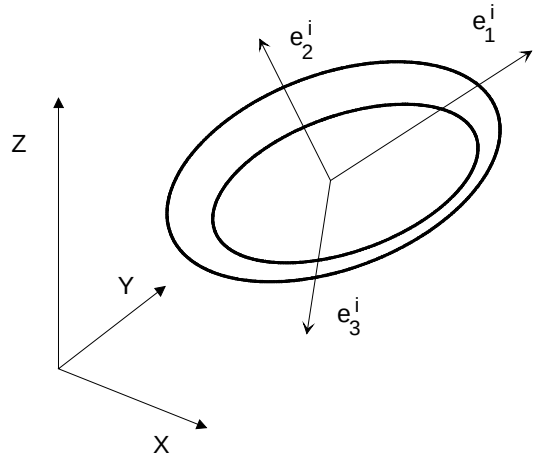


Figure 8. The uncertainty ellipsoid of Nakamura and Xu (1989).

To minimize the uncertainties when fusing measurements from several sensors in the system, Nakamura and Xu tried to minimize the volume of the uncertainty ellipsoid by minimizing the determinant of the error covariance matrix P. This was done by using Lagrange multipliers for solving the proper weight matrix in order to calculate the error covariance matrix.

Probability functions

The modelling of uncertainties by means of probability functions was proposed by Hager (1990). To model the uncertainties in complicated cases such as non-linear ones, he presents probability densities that employ piecewiseconstant density functions. In that way he is able to manipulate the density of the grid and improve the accuracy of the uncertainty modelling.

The computational cost of updating the uncertainties of probability functions is much greater than in the Bayesian method, because the probability attached to each element in each partition must be updated. This is the main disadvantage of the method. The resolution of the grid also affects the existence of bias in the system. Such problems can be avoided by using grid size that is smaller than the required tolerance level. Also, the rectangular structure works best when the observations are highly independent. The grid-based method presented by Hager works better if the parameter estimation can be discretized and the sensors can

have quantized outputs. In that way the updating of the probabilties can also be made much faster.

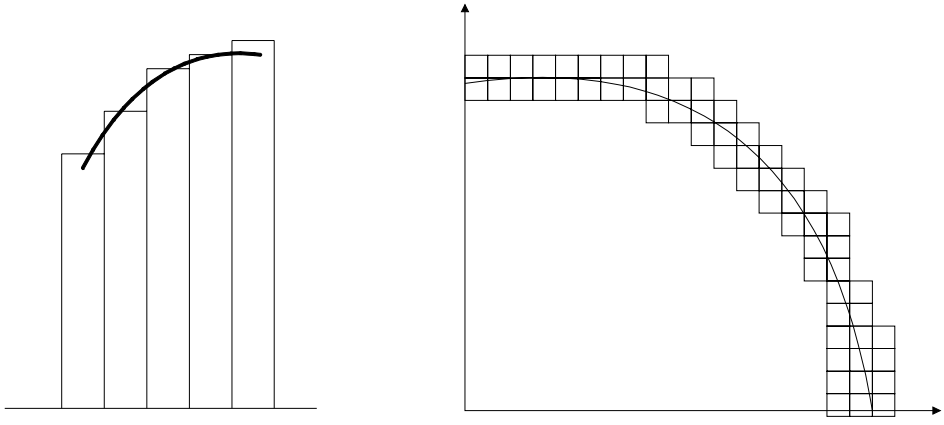


Figure 9. Probability functions according to Hager (1990). Left, a one-dimensional function, and right, a multidimensional approximation of the density function.

Other stochastic methods

The Gauss-Markov random method is a means of modelling Gaussian-form spatial uncertainties. The method has been applied to modelling of the assemblability of parts by Mantripragada and Whitney (1999). To test the probabilities attached to the assembled parts they used variation ellipsoids which accumulated errors from noises in the system and included both translational and orientational errors in the parts. The method offers an effective tool for evaluating uncertainties, especially in assembly.

3.2.4 Simulation modelling

Monte Carlo Localization

In Monte Carlo Localization (MCL) the uncertainties are presented as a probability density with respect to the samples. The method was employed by Dellaert et al. (1999) to improve on their previous work (Burgard et al. 1998). The advantage compared with their previous method is the reduced requirement for computation power and memory. The accuracy is also higher than in Markov Localization and the method is easier to implement. They base these statements on careful comparison between the two methods.

Weaknesses of the methods for modelling spatial uncertainties

Each modelling method entails limitations with regard to accuracy and the possibilities for recovering from failure. In the linearized models, the assumption of modelling the noise as Gaussian is often too far from reality. Usually this becomes significant when the errors in the motion model become too large. These errors are originally mechanical problems involving transmission, gears, encoders and sensors, and have to be considered non-systematic. All systematic errors are considered when a robot system is calibrated, but elimination of the non-systematic ones is a more difficult task. Some sensors have internal parameters with similar behaviour, and their modelling using a Gaussian distribution represents a fair assumption, although determination of the sensor bias should also be included in the calibration process.

Linearized models rely heavily on a priori information about the estimated parameters. If the initial information is unreliable or erroneous, the estimation may very easily fail. In the case where a work object is placed in a robot workcell and no CAD model is available for its localization, it is not always possible to obtain a priori information on the estimator. In the case of the Extended Kalman Filter, when the equations are first-order terms of Taylor series and higher terms are neglected, there is a small deviation between the exact function and the linearization.

Search algorithm-based methods usually suffer from the density of the samples. When using genetic algorithms, the number of samples affects the accuracy of the estimate. Then a compromise has to be found between the number of samples and the time required to calculate the estimates. The number of samples also affects the convergence ratio. The tool required to shorten the computation time is restriction of the solution space. In an actual system, genetic algorithms need four times more computation time than the Extended Kalman filter (Moreno 2002), which means that they cannot be calculated on-line. On the other hand, genetic algorithms are less expensive methods than Monte Carlo localization. So far, most of the search algorithm methods are unfortunately off-line techniques or require part of the calculation to be carried out off-line.

3.3 Combining estimates and uncertainties from different measurements …

Transformation from different measurements or several sensors can be combined, provided it is described in such a way that it can all be evaluated using the same criteria. The form of the information can be different in many ways: the uncertainty in the sensors being based on accuracy and the amount of information (coarse or sparse data), and the time stamps may be different. Forms of merging sensor information can be divided into the following two categories (Luo & Lin 1988):

- Multi-sensor Integration
- Multi-sensor Fusion.

Sensor integration means the use of several sensors to assist the behaviour and movements of the robot without fusing their information, whereas in multi-sensor fusion sensor measurements from different sources are combined or fused together to represent the estimate and its uncertainties in a single format. The error functions and Jacobian matrices can be different, depending on the types of sensors.

3.3.1 Control strategies

The interface by which the sensors communicate with each other is specified in the control strategies. Selection of the most proper strategy depends on the requirements of the system to be developed, i.e. the separate needs and parts of the system and the principles of fusion used for combining the information. Control strategies can be divided as follows (Henderson et al. 1987):

- Sequential / Parallel
- Probabilistic / Deterministic
- Feedback / Open loop
- Modular / Hierarchical
- Goal-directed / Data-driven.

Control of the system may by sequential or parallel depending on the process and the amount of computation. When estimating spatial relationships in this thesis, they are taken as belonging to the sequential control, whereas parallel control is more often used in mobile robotics, when computation time is more critical. The form of modelling the spatial uncertainties in a robot system can be probabilistic or deterministic, and this selection can also be made for control. A large system may also include both types of modelling, so that computation is carried out using deterministic modelling, for example, but global estimation is done using a probabilistic method. The type of sensor may also affect the selection. The principles of a control system can be open or closed when information obtained from the feedback loop is used for controlling it. A control system may also be a modular construction, or else it may be built up in a hierarchical manner, just as it may be either goal-directed or data-driven. This usually depends on the application. The trend in recent years has been to construct robot systems and other machines based on modular architectures. The goal has been ”plug-and-play” thinking, so that parts of the machine can be attached or removed without complicated configuration tasks.

The advantages that can be achieved when using sensor fusion are improvement of the reliablity and accuracy, reduction in uncertainties and the possibility to include sensors of several types in the system. In most of cases the uncertainties in the estimated parameters are also considered when fusing sensor measurements. This is a basic requirement when information from sensors of different types is fused. The completeness of the information gathered on the environment will increase as a result of combining measurements from several sensors, and this may enable objects or whole environments to be measured, which would not otherwise be possible. This may be the case when modelling surrounding objects. When using multiple sensors, the time required for measuring is an important aspect. In this thesis, a laser rangefinder with a laser stripe is used, and with sensor fusion, all the data can be used in the estimation. When designing a multi-sensor robot system, the sensors will usually be selected to complement each other, e.g. a combination of vision, laser rangefinder and touch sensors. Vision or range sensors lack information in cases of occlusion, and if the robot has a tactile sensor attached to the manipulator, information can be obtained from behind occluding objects. The cost of the whole measuring system becomes smaller when collecting the information using several sensors and fusing the information rather than building up separate systems for each observing sensor.

Reliable operation of a multi-sensor system requires high-level flexibility to maintain stable computation. Although modern sensor technology is very advanced, there are still a lot of problems with the sensors. The fusing system has to manage situations where the collecting of information with any of the sensors may be interrupted or may be erroneous. On the other hand, a multi-sensor system should keep the overall functioning of the robot system stable even if one sensor does not work. In that way a multi-sensor system with sensor fusion is more reliable than a robot system with only one sensor.

3.3.2 Sensor fusion in different ranges

Efficient use of multi-sensor fusion is achieved by using the most suitable sensor at each distance from the target. Luo and Kay (1989) present a hierarchical approach for managing sensor fusion in different situations. The model, illustrated in Figure 10, contains four phases: *far away*, *near to*, *touching* and

manipulation. In the lower part of the circle, the most suitable sensors are presented for the proposed distances. The arrow on the right-hand side of the circles represents time, which means that observations are first made far away from the target and come closer and closer to it with time. The range of sensing represents the distance between the target and the sensors, which naturally increases from close to further away.

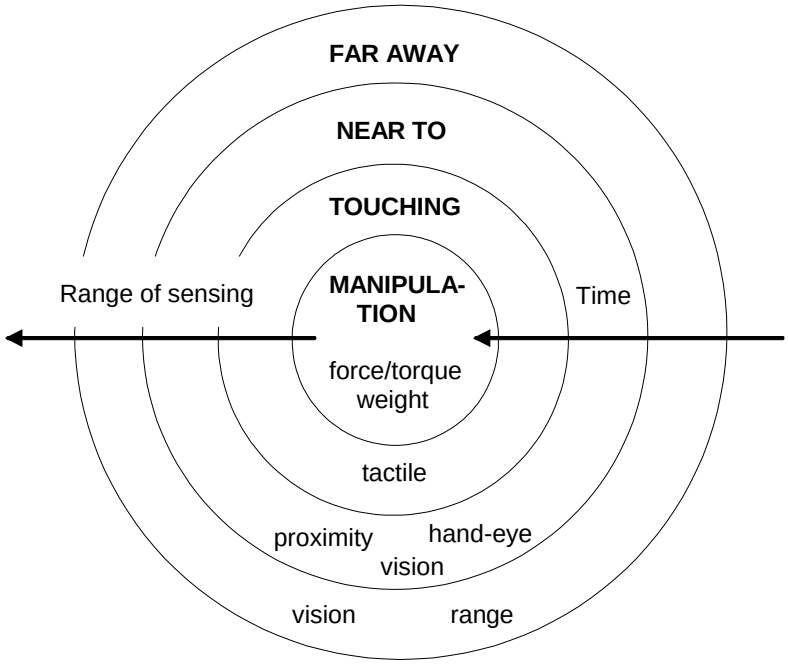


Figure 10. Four phases of the approach to sensor fusion adopted by Luo & Kay (1989).

The concept of a virtual sensor, as presented by Eccles et al. (1989), implies a method for improving the flexibility of a multi-sensor system. It is an arbitrary type of sensor that improves knowledge about the environment and may also include higher-level information. The properties of the virtual sensor also include the fact that the main system does not need to know the type of sensor or form of data acquired. The virtual sensor takes care of all the data transformation processes from sensor reading into submitting the data for sensor fusion.

The combining of measurements into a one large matrix by means of sensor fusion was proposed by Nakamura and Xu (1989). Each new measurement increases the size of the weight and measurement matrix. To keep the computation time at a reasonable level, old measurements are deleted after a given period of time. Fusion is based on Gaussian-form uncertainty representation, which is also used in the estimation procedure.

Minimal representation size (MRS) is a data fusing method in which a minimal representation for the data is selected using certain criteria. Selection includes choosing the properties of the model and the representation size for residuals or errors. The method has been investigated by Joshi & Sanderson (1999) and who use a differential evolution algorithm consisting of a heuristic search and genetic algorithms for finding the solution. MRS is also known as the minimum description length criterion, as it finds the shortest length of program that can be used to construct the data to be observed.

3.4 Discussion of the modelling the spatial uncertainties in a robot system

Methods for modelling spatial uncertainties in robotics have been obtained from the theory of statistics. The most common stochastic method has been to assume that the noise has a Gaussian distribution. Approaches of this kind include Kalman filter -type modelling and Bayesian-form modelling, which are both linearized models. These are very useful when the noise follows the assumed distribution, but quite easily fail when the measurement data are biased.

Another type of stochastic modelling involves probability functions, which also give a noticeable advantage if the measurements can be discretized. Tolerance propagation and dynamic Markov localization represent the deterministic approach and sampling methods used by search algorithms, e.g. Monte Carlo localization, give good alternatives in many applications. Search algorithm methods has been used e.g. in mobile robot navigation, where the movement space has been modelled using a sampling-based technique.

When a great deal of information is available at one moment or as a function of time, there may be a need to combine measurements. Methods for multi-sensor integration or fusion of this kind, as proposed in the literature, are reviewed at the end of this chapter. Most of them include the estimation of spatial uncertainties in the result in the case of fused parameters.

4. Planning of sensing actions

4.1 Introduction

The selection of a versatile set of samples and robot configurations can affect the noise level of the estimated parameters. Tarabanis et al. (1995a) state that the goal of sensing planning is to understand and quantify the relationship between the objects to be viewed and the sensors detecting them. The approach is a model-based, task-directed form of computer vision, but it is employed as a general approach to sensing planning in this thesis. The system for planning computer vision consists of the parts illustrated in Figure 11, including the processing of many of the tasks, sensor models and object models as input information for the system, the main system for sensor planning and output in the form of planned parameters.

The purpose of planning may be divided into two distinct areas (Hutchinson & Kak 1989): how to place the sensor such that the object features can be detected in a best way possible, and how to choose the robot motions in a way that the known target is localized most accurately. The sensing tasks that have to be carried out include object recognition, scene reconstruction and feature detection. These are closely related to camera techniques and are not considered in this thesis. One important purpose of sensing planning lies in applications where computation time is limited, so that the parameters of the model have to be estimated very quickly and often only a few measurements out of a vast amount of data can be used in the calculation.

Thus motions that give a set of samples for estimating the parameters in such a way that spatial uncertainties will be minimized are studied here. It is assumed that a priori information is available from the environment, i.e. sensor models and object models. In addition, planning may also consider internal parameters of the sensor, e.g. focal length. In this thesis a laser-rangefinder is used and the internal parameters are not estimated, i.e. no set of samples is planned for minimizing these parameters.

Sensor planning is an important part of an autonomous robot system, and it includes a method for selecting an optimal set of samples. The methods presented in the literature can be divided into two main types: generate-and-test

and synthesis (Tarabanis et al. 1995a). In addition to these, there are also other sensing planning types, including expert systems and sensor simulation systems. The quality of measurement selection is very important in cases where there is only a sparse set of samples measured using a point-by-point sampling system and when the available data are very noisy. Examples of acquisition systems yielding a sparse set of data are the Coordinate Measuring Machine (CMM), which obtains only a few samples, or a robot system with a tactile sensor. Compared with vision systems, the amount of data achieved using a point laser rangefinder or an ultrasound sensor is much smaller, unless they are scanning sensors. For this reason, sensing planning plays an important role in a robot system when estimating the coordinate transformations and trying to analyze and minimize the spatial uncertainties attached to the parameters of the estimated transformation.

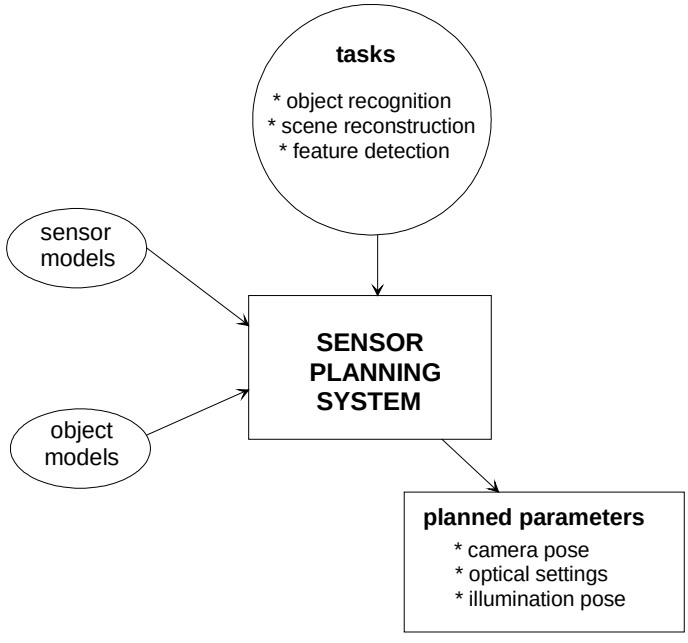


Figure 11. Sensing planning for computer vision (Tarabanis et al. 1995a).

The parameters that the sensor planning system produces in the general case of a vision system are the pose and optical setting of the sensor, and in some cases the pose of an illuminator may also be considered. The main attention in this thesis is paid to calculating the pose of the sensor. The optical settings include

those for internal parameters including visibility, field of view, focus, magnification or pixel resolution and perspective distortion. The illumination parameters include illuminability, the dynamic range of the sensor and contrast (Tarabanis et al. 1995a). These are considered only in the review of the literature.

In the object recognition that precedes the planning phase in general sensing planning systems, the object information is extracted from CAD models, which are nowadays available in manufacturing environments. The required parameters or other geometric information for the planning algorithm will be selected automatically or manually from the CAD models. This selection process will not be considered here, as the information is assumed to be obtained automatically. One special type of sensing planning, known as sensor simulation systems, relies heavily on the CAD model and the model for the whole robot system in a simulation environment. More detailed information on the system is given in section 4.4.2.

This chapter illustrates methods for planning the set of samples, presenting first the generate-and-test method, then the synthesis method, and finally simple examples of two other methods, expert systems and sensor simulation systems.

4.2 Generate-and-test

The most common method for considering the goodness of a set of samples is generate-and-test. It is straightforward to use but usually requires time, due to the iterative manner of selecting the set of candidates. The methods are divided here into two classes, involving the use of either a condition number or an observability index for evaluating the set of samples. One member of the condition number class is a noise amplification index, which weights the difference between the smallest and largest individual values of the identification matrix more than the condition number.

4.2.1 Condition number

One of the most fundamental works in sensor planning is that of Sakane et al. (1987), who introduced the HEAVEN planning system. This uses a spherical representation to illustrate the configurations of a sensor. When planning the positions of the sensor for measuring the object, a geodesic dome is created around the object and divided into small triangular facets. A viewing sphere of radius equal to the distance from the vision sensor to the target object, which is selected a priori, is centred on the object. The first criterion, visibility of the target point, is defined by projecting a ray from the target point of the object to the centre of the appropriate facet, and all the intersections of the object frames are computed. The second criterion, distance, is calculated by computing the distances from the object to the sensor and iteratively selecting the best ones,i.e. those which involve the points closest to the camera. The last criterion, sensor placement, is calculated as an intersection of the candidate facets sorted by the distance criteria. In addition to these criteria, the illumination of the viewpoints is evaluated using the inverse of the condition number, i.e. the illumination for the configuration C_1 can be written

$$C_1 = 1/cond(M_S) \tag{20}$$

where

M_S is the matrix of unit vectors that point from the object surface to the light sources

$cond(M_S)$ is the condition number of the matrix M_S.

The condition number can be calculated when the equations are in the form of a square matrix A $(A^T A)$, the condition number for the matrix A being calculated using the second norm (Kreyszig 1993):

$$cond(A) = \| A \|_2 \| A^{-1} \|_2 \tag{21}$$

The other way is to calculate the condition number from the largest and smallest individual values of the identification Jacobian matrix, as proposed by Zhuang & Roth (1996):

$$cond(J) = \frac{\sigma_{\max}}{\sigma_{\min}} \tag{22}$$

where

$\sigma_{\max}$ is the largest individual value

$\sigma_{\min}$ is the smallest individual value.

The relation between eigenvalues and inidvidual values is that eigenvalues are positive square roots of the singular values to the power of two. The condition number is calculated by dividing the largest individual value in the matrix by the smallest, where the smallest value is theoretically one. In that case all the individual values will be equal to one and all axes of the uncertainty ellipsoid will be of the same length. The geometrical interpretation for this is that the uncertainty ellipsoid has the form of a sphere.

The condition number used by Motta & McMaster (1997) was calculated from the Jacobian matrix of the robot kinematic system. The important aspect when using the condition number is that the parameters and task variables of the Jacobian matrix have to be scaled before calculating the condition number. There are two ways of scaling the data to obtain a better condition number (Nahvi et. al. 1994): task variable scaling, in which both the position and the orientation of the tip have to be scaled to the same precision, and parameter scaling, in which the effects of different parameters at the tip are normalized, i.e. they are translation and orientation parameters. A typical example of parameter scaling is column scaling.

Sallinen et al. (1999) used the condition number to evaluate the goodness of the set of samples after carrying out the measurements. In that case hand-eye calibration of a laser rangefinder was carried out and several sets of samples were generated randomly. The condition numbers for the different sets of samples were then compared with respect to spatial uncertainties. There was

some coherence between the condition number and the spatial uncertainties, but these criteria complement each other rather than sharing the same result.

Calculation of the inverse of the condition number as a criterion for sensing planning was also suggested by Nahvi & Hollenbach (1996), who took the maximum value to be one. They used an inverse condition number to determine the Noise Amplification Index, an indicator which is a product of the inverse condition number and the minimum individual value:

$$O_4 = \frac{\sigma_{\min}}{\sigma_{\max}} \sigma_{\min} = \frac{\sigma_{\min}^2}{\sigma_{\max}} \tag{23}$$

The larger the index, the better the observability and the smaller the difference between the axes of the uncertainty ellipsoid in a geometric sense. The noise amplification index is an indicator of the amplification of the sensor noise and the unmodelled errors. The index was successfully tested with a 3DOF planar robot. As a consequence of the sensitivity of the noise amplification index to the different scales of the Jacobian matrix, the data have to be scaled before computing it, and the measuring conditions should be the same for all the data sets to be compared.

4.2.2 Observability index

An observability index for evaluating variations in the parameters was defined by Borm & Menq (1991). This measure of parameter errors was calculated by multiplying the individual values of the Jacobian matrix, taking the square root of the product and dividing it by the square root of the number of measurements:

$$O = \frac{\sqrt[L]{\sigma_1 \cdot \sigma_2 \cdots \sigma_L}}{\sqrt{M}} \tag{24}$$

where

σ_L is the $L:th$ individual value, and the smallest

σ_1 is the first individual value, and the largest

L is the number of individual values

M is the number of measurements.

The observability measure can be used as a noise minimization criterion when minimizing the index O, i.e. the smaller the index, the lower the noise level in the system.

The parameter estimation results were considered by studying the propagated error covariance matrix over the links in the robot system (Borm & Menq 1991). The test system included a robot with 5 DOF and the position was verified by a coordinate measuring machine (CMM). These authors reached the following three conclusions on the use of the observability measure: 1) the observability measure increases as the upper bound of the RMS value of the residual position error decreases, 2) the selection of configurations is as important as the number of measurements, and 3) the robot calibrations should be carried out with the configurations that have the highest observability measures.

The observability index was later used by Nahvi et al. (1994), who stated the determinant of the Jacobian matrix:

$$\sqrt{\det(J^T J)} = \sigma_1 \ldots \sigma_L \tag{25}$$

where

J is the Jacobian matrix.

This was also called manipulability by Klein and Blaho (1987). Based on equation (25), these latter authors defined the observability index:

$$O = \sigma_L \tag{26}$$

which is equal to smallest individual value. From the geometrical point of view, if the error in estimated parameters is defined as an uncertainty ellipsoid, or a hyperellipsoid as they call it, the semi-axes of that hyperellipsoid are the individual values $\sigma_1 \ldots \sigma_L$. Also, the volume of this hyperellipsoid is proportional to the product of the individual values of the identification matrix $\overline{J}$. As a result, they proposed that both the condition number and the observability index should be used in order to obtain the best results.

The simplification of equation (26) proposed by Klein & Blaho gives a one-sided interpretation of the uncertainties attached to the estimated parameters. If the minimization criterion is an individual value only in one direction, it can easily give rise to a situation in which this one direction is minimized and other directions suffered. Most situations of this kind are much worse than uncertainties distributed homogeneously in the parameter space, because minimization of uncertainties in only one direction leads to a situation in which the uncertainties in other directions are greatly expanded.

A generate-and-test method for planning visual guidance for a robot has been proposed by Heikkilä et al. (1989), who divide the planning task into four phases: selection of the feedback strategy, structure evaluation, selection of the both feature and object evaluation and selection of the viewpoint. The main evaluation consists of selecting a proper feature, which can be an edge or an object. The goodness of the feature information is evaluated in the case of an edge in terms of the length of the edge and the ease of finding the corner. When the object criteria include a combination of edges and possibilities for determining degrees of freedom for the object, the viewpoints of the robot system are planned using two evaluation functions: a goodness factor for the viewpoint, calculated as the product of the goodness factors of the objects in the view, and a goodness value for the path, evaluated by reference to the goodness factors for the different views and the angles between the eigenvectors of the rotation matrices and the normal vector of the plane. The method seems to be suitable for a number of purposes, but its use is restricted to vision systems due to the requirement for detailed feature extraction.

The generate-and-test method is easy to implement and gives a good initial solution for synthesis-based methods (Gu et al. 1999). The potential problems related to it are said by Gu et al. (1999) to be: 1) the computational cost is quite

high in most cases. This problem is expanding, especially as the dimensions of the parameter set are expanding. 2) The constraints on the sensor, e.g. focus, resolution and field-of-view, are often ignored. 3) The solution space for possible candidates is limited to one viewpoint and view direction at the centre of each facet, because of the computation cost, and 4) the uncertainties are not usually considered. In many cases, however, modelling of the evaluation criteria or a cost function for sensing planning is too difficult or even impossible, and in those cases the generate-and-test approach will be the proper solution.

4.3 Synthesis

The synthesis method for selecting the set of samples requires a deeper understanding of a robot and the sensor model than the generate-and-test approach. When building up the system for selecting the samples, the parameters to be minimized have to be selected, togerther with some kind of minimization function or criterion. The minimization criterion can be a condition number, occlusion, uncertainties in the estimated paramaters or a geometrical dimension. The space of the solutions is fairly easy to find when estimating only a few parameters, but as the number of parameters increases, the computation becomes more complicated and it can be very difficult to find the best solution. In some cases in which both external and internal sensor parameters are estimated, the number of parameters to be minimized will reach 14.

One classical synthesis method for generating the set of samples presented by Tarabanis et al. (1995b) is called MVP (machine vision planner). The system takes the object CAD model and sensor information as an input and determines camera poses and settings as outputs. The parameters which are defined are the pose of the camera (6DOF) (r_0, ν) and a principal point on the image plane d, the focal length f and the aperture of the lens a. This means that planning is carried out in an eight-dimensional space, any point in which is called a generalized viewpoint $V(r_0 \quad \nu \quad d \quad f \quad a)$. The optimization function for the estimated parameters can be written as follows:

$$f = \max(\alpha_1 g_1 + \alpha_{2a} g_{2a} + \alpha_{2b} g_{2b} + \alpha_{3b} g_3 + \alpha_4 g_4) \qquad (27)$$

where

g_i is a function of $g_i(r_0 \quad \nu \quad d \quad f \quad a)$

α_i are the weights for the parameters.

The weights are selected initially such that each parameter is weighted equally, but after the optimization a solution for a global minimum will be achieved by searching for the proper value for each weight. Using the general optimization equations presented, the MVP offers a useful tool for synthesis of optimal sensor positions.

4.3.1 Error covariance matrix

Error covariance matrix describes the noise of the estimated parameters around the estimated value. It can be interpreted geometrically by calculating eigenvalues and eigenvectors which describes the uncertainties length of the axis and orientation of the uncertainty ellipsoid. When the error covariance matrix is calculated by propagating all the error sources possible to model, it describes the total uncertainty of the propagated links or estimated parameters. Volume of the uncertainty ellipsoid gives the space e.g. within the end tip of the calibrated tool lies. Therefore it can be used as a criteria for describing the accuracy of the different devices of the robot system.

Uncertainty minimization has been applied to the localization of mobile robots by Borghi & Caglioti (1998), who use position uncertainty as an optimization criterion, i.e. an a posteriori error covariance matrix. Due to the requirement for fast computation in the sensing planning task in the case of a moving robot, the computation is divided into two phases: an off-line phase and an on-line phase. The finding of a solution to the problem follows a grid-based approach. The off-line phase includes dividing the solution space into separate cells, which are then used for calculating the optimal solution in the on-line phase based on the given criterion. In the estimation of uncertainties, they use uncertainties at a known point, which is an advantage when studying an a posteriori covariance matrix with the same units, i.e. the translation parameters of a point.

When using an uncertainty matrix as a criterion, a situation arises in which parameters with different units are selected for the covariance matrix. This problem was solved by Borghi & Caglioti (1998) by using uncertainties at a point as a criterion for sensing planning when the units of the error covariance matrix were translation parameters. In the situation where the 2D reference pose of the robot is represented as $g(X,Y)=0$, the uncertain pose as $g(X',Y')=0$ and the noise of the pose as $(\delta x, \delta y, \delta v)$, the relative non-displaced pose can be written as

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = \begin{bmatrix} \cos\delta\vartheta & \sin\delta\vartheta \\ -\sin\delta\vartheta & \cos\delta\vartheta \end{bmatrix} \begin{bmatrix} X-\delta x \\ Y-\delta y \end{bmatrix} \tag{28}$$

and the covariance matrix for the robot pose as:

$$\Lambda = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\vartheta} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{xx} \\ \sigma_{\vartheta xx} & \sigma_{\vartheta y} & \sigma_{\vartheta\vartheta} \end{bmatrix} \tag{29}$$

The position error of point I is indicated as being $(\delta x_I, \delta y_I)$ and the accuracy $\sigma_{\rho\rho I}$ of the position I is presented by the trace of the covariance matrix Λ_{xyI}, i.e. the sum of the eigenvalues. This quantity is equal to the mean squared distance between the true position of I and the estimated position. The equation can be written:

$$\sigma_{\rho\rho I} = E\left[\delta x_I^2 + \delta y_I^2\right] = \sigma_{\delta xI\delta XI} + \sigma_{\delta yI\delta yI} \tag{30}$$

where

$E[\ ]$ is the covariance of the given parameters.

The problem can now be formulated: using the given estimates of the pose parameters and their covariance matrix, which is the point on Γ (Γ here describes the surface around the robot) to be explored at which the criterion $\sigma_{\rho\rho I}$ will be minimized, equation (30), the minimization is carried out by suboptimal exploration, which relies on a priori information on the environment,

by selecting the measurements with the least uncertainties. The method has proved promising, and is especially suitable for curvilinear environments. In addition, localization works fine with a very few measurement points, so that it can be used in piecewise linear environments.

In his previous work, Caglioti (1994) had used a determinant of the a posteriori error covariance matrix as a criterion for planning the set of samples. In parameter estimation he uses a minimum error variance estimator, which enables him to estimate the error covariance of the estimated parameters. When locating objects, information about their surfaces has to be available, and the minimization in the sensing planning phase is divided into several subproblems for each surface.

A composition of robot movements when the robot approaches a cube has been proposed by Nilsson & Nygårds (1996). The movements are programmed off-line and use measurements made by a laser rangefinder and Gaussian-form noise for modelling the uncertainties in the robot system. The uncertainties are considered by calculating and studying the error covariance matrix for the estimated parameters. The state equation of the moving robot is written as follows:

$$x_{k+1} = x_k + s_k + bdiag(s_k)w1_k + cw2_k \tag{31}$$

where

$x_k = [x_k, y_k]^T$ is the position vector of the robot with respect to the object to be gripped at time t_k

s_k is the unknown planning variable describing the desired incremental change in the robot position

$w1_k, w2_k$ are the sequences of variables

b, c are the model variables.

When the output of the sensor is modelled as $z_k = h(x_k) + v_k$, the cost function for the system can be written as follows:

$$J(s_0,\ldots,s_{N-1}) = E\left[(x_N - x_g)^T (x_N - x_g) \right] \quad (32)$$

where

$J(s_0,\ldots,s_{N-1})$ is a function of the poses of the robot

$s_0,\ldots,s_{N-1}$ are poses of the robot

x_g is the requested final position

x_N is the actual final position

$E[\ ]$ is the covariance of the given parameters.

A solution for the cost function is obtained using the Extended Kalman filter. The planning algorithm is repeated when a new observation is received and the solution is always adjusted with respect to the information available. The final error can be used for optimizing the speed of the approach by considering the criterion used in the planning.

4.3.2 Surface properties

A method for planning viewpoints for a partial dimension inspection has been presented by Sheng et al. (2000), although their assumption that the targets to be inspected have free-form surfaces makes the inspection more difficult. The test setup consisted of a CCD camera system mounted at the TCP of the robot. The internal parameters of the camera system are assumed in the sensing planning to be constants, and only the external parameters, i.e. the pose parameters, are to be calculated. The sensing planning procedure consists of two main steps: 1) finding flat patches on the surface of the object, and 2) determining the viewpoints for each flat patch. A flat patch is defined as a set of connected triangles which correspond to a certain continuous area on a surface, and the

angle between the normal and each triangle within an average normal patch must be within a given threshold. The search for flat patches is carried out by increasing a seed area, selected as being the triangle with largest area on the surface. The neighbouring triangles are tested with the given criteria and the growth process is repeated until there are no more acceptable triangles to add. An improved version of flat patch selection is presented by Sheng et al. (2001), where the flat patches are constructed using the adjacency information on the surfaces and similarity between the surface normals.

Sheng et al. (2001) adopt a ”bounding box” approach to determine the optimal camera viewpoints. This means that a bounding box is constructed on a given patch to project all the vertices in three orthonormal directions and to find the farthest and closest points. The direction with the maximal areal view is chosen as the optimal direction, and the other two directions are selected so that the points can all be covered by the smallest rectangular field of view. The search for viewpoints is carried out recursively by selecting patches and subpatches depending on the success of the search. Experimental tests showed that the system works with real CAD models.

An optimal sensor placement for visual dimension inspection has also been proposed by Gu et al. (1999), who used error measures of two kinds for formulating the placement problem: displacement error and quantization error. The displacement error includes errors of camera placement (6 DOF), which give rise to errors in the image plane that can be computed by calculating partial derivatives of placement with respect to the image in the image plane. The random variables of camera location are assumed to be Gaussian in distribution, and since the displacement error is the sum of several variables with Gaussian distribution, the displacement itself is also Gaussian-distributed. The quantization error arises from the accuracy of the image digitization. It is assumed in the work referred to that a line in the image plane has equal noise in the directions x and y of its endpoints. The noise in the image plane is also Gaussian-distributed. When integrating both the displacement and the quantization error, the result will be Gaussian distributed with a zero mean. The dimensional inspection is carried out by calculating the dimensional error and its deviation and comparing the result with manufacturer’s information. If the resulting deviation is smaller than the given threshold, the test will be passed.

4.3.3 Occlusion

Occlusion has been used as a criterion for computing optimal viewpoints by Tarabanis et al. (1996) and Maver & Bajcsy (1993). Especially in the case of complicated objects, some points on the object may not be visible. In such cases the use of occlusion as a criterion for sensing planning is very effective. The forms of occlusion can be divided into two cases (Tarabanis et al. 1996): self-occlusion, which means that the object which contains the feature to be measured is occluding the view itself, and location of a desired feature in such a way that it is impossible to measure using the given sensor, i.e. the sensor has too limited an angle of view or is too large to account for small hollows in the work object.

Tarabanis et al. (1996) propose two ways of detecting occlusions. They can either be controlled for on the basis of the boundary representation of the objects in the work space, as provided for in the algorithm, the optimal pose values for the camera being determined using a generate-subdivide-select algorithm, or else a decomposition-based algorithm can be used in which the occlusion is considered by checking the boundary faces. The authors compare these two algorithms and show the boundary-based one to be clearly better in terms of both performance and robustness.

Maver & Bajcsy use a two-stage method of occlusion observation. Their first scan provides initial information on the scene, and the resulting data are utilized in the second scanning to check the occlusions from the potential directions of observation.

4.3.4 Other synthesis methods

A Bayesian statistical decision theory is used by Cameron and Durrant-Whyte (1990) for determining the optimal sensing locations in recognition and localization operations. The object and sensor models use the probabilistic approach, i.e. the Gaussian-form noise models proposed by Durrant-Whyte in his previous work (Durrant-Whyte 1988). Cameron and Durrant-Whyte uses probabilistic membership functions to describe the uncertainties attached to the

sensed objects, and they also include information on the shape of the object, its dimensions and uncertainties within the combined uncertainty of the system. Utility functions are used to evaluate the merits of each sensed position, and the sensed positions are collected into a probabilistic interpretation tree and a sequential analysis is used to test the proposed hypothesis. The poses that pass this two-stage test are accepted as sensed positions. The system was tested in a 2D case with an actual robot-camera system and the results were successful.

The planning of sensing strategies for multi-sensor capabilities has been studied by Hutchinson & Kak (1989). Their solution comprises a complete system including object identification based on a hypothesis concerning the object and the surrounding scene. They use the Dempster-Shaver theory for object feature matching, defining the following sets of hypotheses: feature matches, object consistency and aspect consistency. The set of motions is composed of a set of planning motions for several observable features of the target object. Their matching criterion is the difference between the dot products of the measured and modelled surface normals. The sensors they used for measuring the robot workcell included a laser range sensor, a vision sensor and a force/torque sensor. The best sensing strategy for these was evaluated using the following three criteria: 1) viewpoint, which corresponds to the principal viewpoint for a given aspect, 2) features which can be observed from the selected viewpoint, and 3) a hypothesis based on this initial information. This process is repeated until the minization of ambiguity reaches the required level.

Almost all the papers concerning sensing planning have an approach with a static target, but research has also been carried out into sensing planning for moving targets. Methods and results have been described by Abrams et al. (1999). The approach to dynamic sensing planning is basically the same as to static planning, but the sensing operation has to be carried out within a certain time period. There are two types of movement in dynamic sensing planning: one in which the sensing tool is moving and the target objects remain stable, and one in which both the sensing tool and the target objects are moving. The optical parameters of the sensor are planned in the first phase and the measurements for localizing the target object are performed in the second. The system computes all the same parameters as static planning systems and in addition temporal intervals, i.e. the time related to the sensing event. For the dynamic sensing process the system has to have an additional motion model for the environment.

The idea of the planning is to find temporal intervals for viewpoints, compute swept volumes of the target objects that present the movement and then use the viewpoint algorithm to compute sensor positions, orientations and optical parameters that are valid for each interval. The paper also presents an improved version of the viewpoint planning algorithm that includes a new way of integrating the field-of-view and resolution constraints. Even if the system works well, there will be much to be done in the future regarding the time intervals and how to manage them. The temporal interval decomposition method they use is a rough method and they will presumably improve it significantly.

Simulated Annealing is a stochastic optimization method for generating an optimal set of samples derived from Monte Carlo methods (Zhuang et al. 1994a). Its advantages are that it offers an easy way of implementing the joint travel limits in configuration selection and that it avoids a local minimum point in the minimization function in the context of gradient-based methods.

Simulated annealing is implemented with the following procedure: 1) formulation of the cost function F, 2) selection of the initial configurations through the probability distribution $Q(.)$ 3) generation of candidate states by specifying the conditional probability distribution $D(.,.)$ and 4) design of the cooling schedule $\{T_k\}$. To observe the state of identification of Jacobian and kinematic parameters, the authors use the condition number and the observability measure. The first step, definion of a cost function, is carried out using the condition number or the observability index, whileiIn the second step, the probability function is determed by distributing it uniformly over the parameter space. Candidates are generated by random selection of a robot joint variable vector, every new vector being selected in such a way that it does not contain any joint value in common with previous vectors. The cooling schedule can written as follows:

$$T_k = \frac{\Gamma}{\ln(l\ k_s + 2)} \tag{33}$$

where

$\Gamma = T_0 \ln 2$ in which T_0 is the initial temperature

k_s is a constant (>0)

$l = l(k) = truncate(k / L_k)$.

This is a finite-time equation generated using a Markov chain of finite length L_k . Although simulated annealing is a computationally expensive method, it can be recommended for generating an optimal set of samples. The compution is carried out off-line, which makesiteasier to use in real systems.

The synthesis method gives a much more effective way of generating an optimal set of samples than generate-and-test. In addition, a larger parameter space can be covered. The disadvantage is the difficulty of forming the criterion function for the selection of samples. When looking for a flexible and intelligent way of generating a set of samples, synthesis usually has more advantages than other methods.

4.4 Other sensing planning methods

In addition to generate-and-test and the synthesis approach, there are two less common methods for sensing planning, namely expert systems and sensor simulation systems. Less intensive work has been put into developing these methods, but there are several applications for which they constitute the proper solutions.

4.4.1 Expert systems

Expert systems can be included among the sensing planning systems. Their planning relies on a knowledge base derived from a human expert's knowledge of image processing techniques. The idea of the planning system is that it generates a sequence of primitive image processing operations to solve the given vision problem. Expert systems can be divided into four categories (Matsuyama 1988): consultation systems for image processing, knowledge-based program composition systems, rule-based design systems for image segmentation

algorithms and goal-directed image segmentation systems. Expert systems have been developed for a number of tasks (Matsuyama 1988), including signal interpretation, medical diagnosis, circuit design and troubleshooting and plant control. Two knowledge-based systems are presented here.

Kitamura et al. (1990) presented an expert system for a part feeder. The pose of the object is recognized using predetermined criteria concerning its features, including the area of the component and angular features. Both feature values are measured using binary images. The system as presented consists of three modules: an image processing expert (IPE), a decision tree generator (DTG) and a feature selection expert (FSE). The IPE component generates an image processing procedure to measure the values of the shape features, the DTG calculates the effectiveness value of the terminal level shape feature and a threshold value for the classification, and the FSE selects the shape feature to generate the decision tree. This method was applied to their part feeder with successful results.

The other knowledge-based expert system was presented by Matsuyama (1988). The general architecture for expert systems for image processing includes a reasoning engine, knowledge about image processing techniques, a library of image processing operators and a database of characteristics of the input and processed image data. The reasoning engine operates at two levels: generation of an analysis plan and selection of an operator, and adjustment of parameters. The whole system is controlled with a user interface.

4.4.2 Sensor simulation systems

In sensor simulation systems a scene is visualized on the basis of information obtained from different components of the world, including target objects, sensors, the robot and light sources. The planning is carried out visually or may be based on a generate-and-test approach, for example.

The sensor simulation system called VANTAGE proposed by Ikeuchi & Robert (1991) includes model-based vision system that recognize the object based on its appearance. It models the world in 3D, but recognition takes place in a 2D space. To model objects in this 2D space they define sensor detectability as a measure

to describe where the sensor can see the object. The input to the system consists of boundary values for the object, and the result is the set of measurement poses. Objects are modelled using structures at different levels (trees) and object recognition makes use of the illumination-level structure. The uncertainties attached to each feature are taken into consideration in the recognition. The system has been tested and works fairly reliably despite its high computational cost.

The other methods that exist alongside generate-and-test and synthesis offer good solutions for a number of planning applications, but their performance is highly dependent on the user. The results achievable with an expert system depend on the knowledge base, and the sensor simulation method is restricted by the simulation environment and its accuracy. Despite some disadvantages and exceptional features, there are many applications for which these systems offer the best solution.

4.5 Discussion of planning of sensing actions

The objective of sensing planning is to generate a set of samples so that, using the data, the estimates of the coordinate transformation will be reliable and the parameters will decribe it with a low level of uncertainties. The two main approaches to planning are generate-and-test and synthesis. If the estimation process is known, the synthesis method will provide effective tools for generating a set of samples. Depending on the application, there may be a requirement for on-line planning, and in such cases the method will have to be fast to compute, so that synthesis is an obvious solution for these purposes. Generate-and-test is simplier to use and is a more common method for planning samples.

Generate-and-test is a straightforward approach for selecting the set of samples. It is usually computationally heavy, however, and so the calculations are performed off-line. The interpretation of the results is quite obvious and criteria for the set of samples are often taken from applied mathematics, including the condition number and observability index. In the synthesis method, the set of samples is generated based on given criteria, basically of the same kind as in the generate-and-test approach. Calculation of the set of samples is quite fast and

this means that the method is often used in time-critical applications. Other criteria used in the synthesis approach are the properties of surface profiles and surface occlusion. Typical surface properties are flatness and field of view. Occlusion is very useful when the object includes features which cover some areas of its surface. In such cases it is imperative to select the viewpoint on the basis of occlusion.

There are applications, especially in industry, which use other sensing planning methods, including expert systems and sensor simulation systems. In addition, modern CAD / simulation softwares have similar properties (Deneb 2000). The demerit of these methods is their heavy dependence on both the human operator of the system and a database consisting of expert rules defining the system base.

5. A comprehensive framework for the modelling, estimation and planning of parameters and uncertainties in robot workcells

5.1 Introduction

The estimation of model parameters in a robot workcell includes hand-eye calibration, work object localization and surface model parameter estimation. A stochastic method for estimating all these three phases is introduced in this chapter. The method is based on Newton-Rhapson iterative estimation and uses Bayesian-form modelling of spatial uncertainties in the robot workcell. The kinematic modelling of the robot is performed by the Denavit-Hartenberg method, which enables a flexible approach to be adopted for both solving the kinematics and modelling the kinematic structure when propagating spatial uncertainties.

This chapter also presents a method for modelling spatial uncertainties using covariance propagation that can be applied to every phase of the calibration of a robot system, including hand-eye calibration, work object localization, estimation of surface model parameters and sensor fusion (section 5.3), in which measurements from several sources are combined. Means of modelling the spatial uncertainties involved in localization and the surface uncertainties of the work object with B-spline surfaces are presented here which have not been reported in the literature in this form.

A synthesis-based method for planning measurements is introduced in section 5.4. This seeks to minimize the a posteriori error covariance matrix and uses the geometric distance between consecutive points as a minimization criteria when measuring points on the surface of the work object. A verification of the criteria is provided, and equations for planning the set of samples for work object localization are given.

5.2 Estimation of parameters in robot workcells

The calibration of a robot means here estimation of the spatial relations between its wrist and the origin of the sensor (hand to eye), between the work object and the world and between the robot base and the wrist, together with the form of the work object surface. The equations are composed for a 6DOF robot and a 5DOF sensor and are explained in detail.

5.2.1 Hand-eye calibration

Here a method for estimating the parameters for hand-eye calibration using the Newton-Rhapson method is presented. The method is based on iterative Newton-Rhapson method and the same method is used also in work object localization and estimating the surface model parameters in this thesis. The sensor used in this work is a laser rangefinder, modelled as a transformation from the wrist (TCP) of the robot to the origin of the sensor frame, so that the measurement value is the distance from sensor origin to the measured point. This distance is obtained only in one dimension, but a versatile set of data can be obtained when using robot motions to move the sensor. One-dimensional measurement also relaxes one degree of freedom and allows several measurement positions and orientations. It means that only five pose parameters have to be estimated instead of the full six. The state vector describing the transformation between the sensor origin and the robot TCP can be written as follows:

$$m_S = \begin{bmatrix} x & y & z & \phi_x & \phi_y \end{bmatrix} \tag{34}$$

where the first three terms describe the translation and the next two the orientation of the estimated transformation in euler angles. The calibration is carried out by measuring the distances between the sensor origin and the surface of the calibration plane. The state of this plane can be determined using three parameters, as follows:

$$m_{CP} = \begin{bmatrix} z & \phi_x & \phi_y \end{bmatrix} \tag{35}$$

where the first parameter describes the distance from the origin in the direction z of the plane and the next two the orientation of the plane. Hand-eye calibration can be performed more reliably using parametric surfaces, due to the more accurate model obtained as compared with patched surfaces. A plane surface is used here, and the error function used as a minimization criterion is given only for a plane surface. The measuring system is based on measurement motions generated off-line, a rough estimate of the sensor to TCP transformation and a rough estimate of a position and orientation of the calibration plane, Figure 12. The position and orientation of the calibration plate are also determined during estimation of the hand-eye transformation, an idea first introduced by Järviluoma (1991). Parameter estimation is more sensitive to error in the a priori information on orientation values than are the translation values of the estimated parameters. The same behaviour has been noted in the literature (Gunnarsson & Prinz 1987, Dornaika & Horaud 1998, Heikkilä et al. 1993). The advantages of the method proposed here is the simplicity of the calibration plane required for calibration and flexibility of movements of the robot for collecting set of samples.

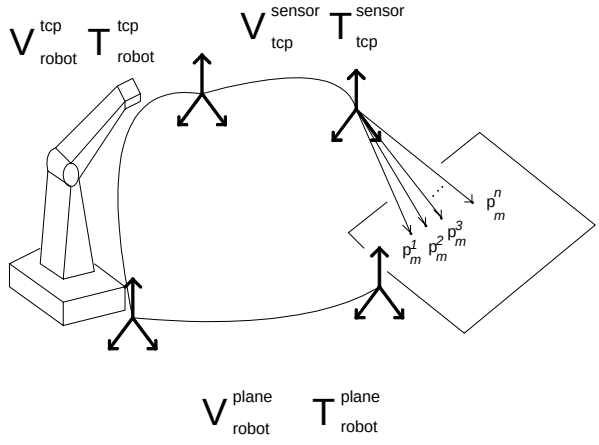


Figure 12. Coordinate transformations in hand-eye calibration.

The following coordinate transformations are used in hand-eye calibration:

- plate frame in the world (robot base) frame: H_P, $V_{robot}^{plane}, T_{robot}^{plane}$, δ_P
- robot's TCP in the world frame: H_R, $V_{robot}^{tcp}, T_{robot}^{tcp}$, δ_R
- sensor frame in the robot's TCP frame: H_S, $V_{tcp}^{sensor}, T_{tcp}^{sensor}$, δ_S and
- measured range value of the sensor d.

where

H's are 4x4 homogenous transformation matrices including translation and orientation

V's are 3x3 rotation matrices,.

T's are 3x1 translation vectors, and

δ's are 6x1 noise vectors of the pose parameter. $\begin{bmatrix} x & y & z & \phi_x & \phi_y & \phi_z \end{bmatrix}$.

The direction of measurement of the sensor is set here to the direction z. When obtaining only a distance value d, the measured point in the sensor frame can be written as follows:

$$p_S = \begin{bmatrix} 0 & 0 & d \end{bmatrix}^T \tag{36}$$

When using a priori information on the hand-eye transformation, the point can be expressed in the robot TCP frame as follows:

$$p_{TCP} = V_{tcp}^{sensor} p_s + T_{tcp}^{sensor} \tag{37}$$

The transformation $V_{tcp}^{sensor}, T_{tcp}^{sensor}$ is constant and is estimated during the iterative estimation process. The next transformation is from the robot TCP to the robot base frame. This is different in each motion measurement and an accurate value

for this transformation is measured from each joint encoder in each measurement. This means that in each pose the joint values are read from the controller of the robot and converted into pose form using the Denavit-Hartenberg model, including the rotation matrix V_{robot}^{tcp} and the translation vector T_{robot}^{tcp}:

$$p_R = V_{robot}^{tcp} p_{TCP} + T_{robot}^{tcp} \tag{38}$$

Finally, the measured point is transferred to a calibration plate frame. An initial estimate for the transformation is used at the beginning, and the result is updated during the estimation process:

$$p_P = (V_{robot}^{plane})^T p_R + (V_{robot}^{plane})^T (-T_{robot}^{plane}) \tag{39}$$

A fitting criterion, i.e. an error function, is required for estimating the coordinate transformations. Since the measurements are on the plane surface and the coordinates of the calibration plane surface have been chosen in such a way that the plane x-y coincides with the plane surface and z direction z points upwards, the error function can be written as the z-coordinate of the measured point on the plane surface:

$$e_{plane} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} p_P \tag{40}$$

where

p_P is the measured point transformed into the calibration plate frame.

The system is non-linear and has to be linearized around zero, i.e. small change of transformation $\Delta m = \bar{0}$. This means that the measurements are fitted to a system model and a Jacobian matrix is calculated by composing a partial derivative of the error function e_i with respect to the parameters to be estimated, m:

$$\frac{\partial e_i}{\partial m} = \frac{\partial e_i}{\partial p_i}\frac{\partial p_i}{\partial m} \tag{41}$$

This equation consists of the following partial derivatives: $\partial e_i / \partial p_i$ is a partial derivative of the error function with respect to a point in the sensor frame, and $\partial p_i / \partial m$ is a derivative of a point in the estimated pose frame with respect to all six pose parameters. The method proposed by Lowe improves the performance of the computation of a solution for the derivative given here (Lowe 1985). According to his principle, small rotations around the orientation axis (ϕ_x, ϕ_y, ϕ_z) can be marked as translations in the coordinate axis. This assumption produces only translation values in the Jacobian matrix, and computation is simpler.

When estimating parameters for the transformation from the TCP frame to the sensor frame, the a priori information on these is assumed to be quite close to the final estimate. Small corrections are calculated for each pose parameter in the course of the iteration and added to the values obtained from the previous iteration step. The partial derivatives for the estimated parameters m_S are calculated as partial derivatives from the error function with respect to the estimated parameters:

$$\frac{\partial e}{\partial \overline{m}_S} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} (V_{robot}^{plane})^T V_{robot}^{tcp} V_{tcp}^{sensor} \begin{bmatrix} 1 & 0 & 0 & 0 & d & 0 \\ 0 & 1 & 0 & -d & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{42}$$

where

d is the range measurement of the sensor

V_{world}^{plane} is the rotation matrix from the world frame to the calibration plane frame

V_{world}^{wrist} is the rotation matrix from the world frame to the robot wrist (TCP) frame

V_{wrist}^{sensor} is the rotation matrix from the wrist frame to the sensor frame.

Parameters for the height and orientation of the calibration plane are estimated when it is assumed that the state vector of the pose parameters consists of a full set of six DOF pose parameter vectors:

$$\frac{\partial e}{\partial m_P} = \begin{bmatrix} 0 & 0 & -1 & -p_y & p_x & 0 \end{bmatrix} \tag{43}$$

where

p_x, p_y are points in the calibration plane frame.

Both of the equations above include zero columns in the estimation, and these are removed from the Jacobian matrix, which in general consists of matrices for the TCP to sensor transformation m_S and matrices for the robot base to plane transformation m_P, as follows:

$$\frac{\partial e}{\partial m} = \begin{bmatrix} \frac{\partial e}{\partial m_S} & \frac{\partial e}{\partial m_P} \end{bmatrix} \tag{44}$$

Each measurement in the Jacobian matrix is inserted into equation (44), to give one line with eight columns. The total Jacobian matrix has a size of Nx8, where N is the number of measurements, as follows:

$$J_m^e = \frac{\partial e}{\partial m} \tag{45}$$

The corrections Δm for the sensor and plane transformations can be calculated using the equation (46):

$$\Delta m = -\left(J_m^{e^T} Q^{-1} J_m^e + P^{-1}\right)^{-1} J_m^{e^T} Q^{-1} e \tag{46}$$

where

Q is the weight matrix of the measurements, see below

J is the Jacobian matrix described above

P is the a priori error covariance matrix.

This equation is close to the observer part of the Extended Kalman Filter, or without the weight matrices and a priori error covariance matrix, close to LSQ estimation. The weight matrix Q is used to weight the measurements with respect to their uncertainties. More information on this is given in the next paragraph. The initial assumption of the error covariance matrix P refers to a Bayesian form of estimation and can be taken from the previous estimate.

The parameters are updated using the method proposed by Lowe (1985), in which the correction parameters are not directly added to the initial parameter values but instead are added using a corresponding correction transformation. The correction Δm in equation (46) includes correction parameters for translation and orientation in Euler form (Paul 1983), so that they are converted into a homogeneous transformation, i.e. $\Delta H_{\Delta} = (\Delta V, \Delta T)$

$$H_{n+1} = H_n \Delta H \tag{47}$$

where

H_n is the coordinate transformation of the estimated parameters before the correction step,

H_{n+1} is the transformation including the estimated parameters after correction,

ΔH is the correction transformation based on the estimated parameter invariants of LSQ.

The correction transformation ΔH is calculated in each iteration step.

The parameter estimation procedure has the following steps:

1. Obtain measurement points from the reference target object, determine the error function and the parameters to be estimated.

2. Give initial values m_{init} for the estimated parameters m_n.

3. Compute the points (x_0, y_0, z_0) in the reference frame and the error function e related to these points.

4. Compute the Jacobian matrix J_m^e related to the estimated parameters.

5. Use Bayesian estimation to compute the correction values Δm for the estimated parameters.

6. Update the values for the estimated parameters with the resulting correction transformation matrix ΔH.

7. Repeat steps 3–6 until the correction parameters Δm are small enough.

8. When the correction parameters Δm are small enough, the updated parameters m_n may be taken as the resulting estimated parameters m.

Depending on the noise in the system, the estimation should converge to its minimum after four to six iteration steps. In the updating of translation and orientation parameters, the translation parameters are updated first in each round and the orientation parameters after that. When using the weight matrix, the measurements with less uncertainties are weighted more than the uncertain ones. When modelling the noise in the system, the uncertainties from each possible source are taken into consideration.

The advantage of the method is its relatively simple way of manipulating the uncertainties in the estimation process. This is due to the fact that uncertainties are modelled using simple geometric features such as points, so that the weight matrices and Jacobian matrices do not become too complicated. It would be a hard task to compose equations for fitting two complicated surfaces, for

example, without splitting them into simple features. In addition, the iterative method for parameter estimation also works in cases where the measurements include noise, which is typical in actual systems.

5.2.2 Work object localization

Parametric surfaces

Before the manufacturing task, the work object is located in a robot working area and is usually fixed with fastening jigs. The location of the work object is roughly known, and off-line programmed task measurements are generated based on that information. For accurate surface treatment tasks, however, e.g. surface inspection or assembly, the location of the work object has to be known accurately with respect to the base of the robot. In the case of single objects or small lot sizes, there is no reason for building up hard jigs to clamp the objects, and the most flexible solution in such cases is to localize each work object separately. This gives an accurate transformation for each pose and the overall accuracy is not disturbed by cumulative errors brought about by using one localization for all work objects. The work object is localized by measuring it from different directions depending on its surface form. The idea of localization is to minimize the distance between the set of measured points and a model of a reference surface on the work object, Figure 13. An error function is used as a fitting criterion in this minimization.

The pose of the work object is determined by six parameters:

$$m_{wo} = \begin{bmatrix} x & y & z & \phi_x & \phi_y & \phi_z \end{bmatrix} \tag{48}$$

As in the case of sensor hand-eye calibration, the measured point is transformed first from the sensor frame to the TCP frame and then from the TCP frame to the robot base frame and from the robot base frame to the work object frame, in which the error function is calculated.

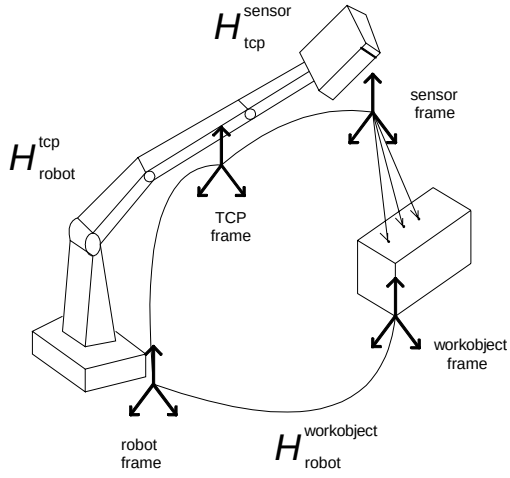


Figure 13. Coordinate transformations in work object localization (Sallinen & Heikkilä 2002).

The error function depends on the surface form of the work object. If the surface is a plane, it is described with a surface normal n and the direct distance l from the origin of the work object. The error function can then be written as a dot product of a normal vector with a point on the surface

$$e_{wo} = n.p - l \tag{49}$$

where

n is the surface normal vector of the reference surface,

p is the measured point on the surface,

l is the shortest distance from the reference surface to the origin of the work object.

The error function in equation (49) is non-linear and is related to the estimated parameters, which are estimated iteratively by linearizing the error function around the nominal values of the work object location parameters within each iteration step. As in the case of sensor hand-eye calibration, the corrections are

calculated in each iteration step using linear models for the error function. Linearization around $\Delta m = \bar{0}$ gives for the localization of the work object

$$\frac{\partial e}{\partial m_{wo}} = \frac{\partial e}{\partial p}\frac{\partial p}{\partial m_{wo}} \tag{50}$$

which can be written further as

$$\frac{\partial e}{\partial m_{wo}} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} (V_{robot}^{wo})^T V_{robot}^{tcp} \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \tag{51}$$

where

n_x, n_y, n_z are the components of the normal vector of the reference surface,

p_x, p_y and p_z are the measured point in the work object frame.

The measurements are collected from at least three sides of the work object, so that the requirements for estimating all six pose parameters are fullfilled. The measurements compose a Jacobian matrix J_m^e, with each measurement producing one line. The corrections for the estimated parameters are calculated using equation (51), as determined previously.

The iteration proceeds in the same way as in the calculation of sensor hand-eye calibration and takes 4–8 steps to converge to the minimum depending on the condition, e.g. level of noise.

Patched surfaces

Localization of patched surfaces requires a slightly more complicated algorithm. The principle is the same, fitting of the measured point cloud into the reference model of the surface. Complicated surfaces are modelled in the present work using a B-spline surface patch as proposed by Choi (1991). The criterion for fitting is the shortest distance between the measured point and the closest point calculated on the surface patch:

$$e = p_{ref} - p_{meas} \tag{52}$$

where

p_{ref} is the reference point on the nominal surface

p_{meas} is the measured point.

The reference point on the surface is calculated by means of an iterative search throughout the surface, the distances from 9 reference points being calculated in each iteration step and the next location being taken to be around the shortest distance. This is repeated up to four times, by which point a satisfactory level of accuracy is usually achieved, Figure 14.

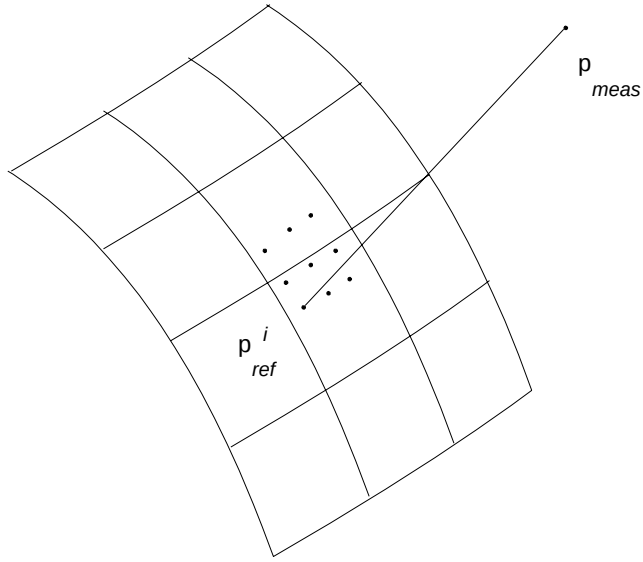


Figure 14. Search for a reference point on a B-spline surface.

In this case the error function is three dimensional and it produces three lines in the Jacobian matrix, i.e. errors in the x, y and z dimensions.

The partial derivatives of the error function with respect to the parameters to be estimated can be written as follows:

$$\frac{\partial e_i}{\partial m_{wo}} = \left(V_{robot}^{plane}\right)^T V_{robot}^{tcp} V_{tcp}^{sensor} \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \qquad (53)$$

where

p_x, p_y and p_z are measured points in the work object frame..

The selection of surface points is more sensitive when localizing patched surfaces than when using parametric surfaces. Especially when localizing surfaces that are close to planar in form, the number of points has to be large, and there should also be measurements from other directions. Otherwise several problems will arise with convergence.

5.2.3 The robot base-wrist relation

The robot movements are based on a model of the robot, with the different parts of the model determining how the encoder values of the servos of the joints are transformed into TCP values for the robot. The parts between the encoders and TCP consist of four models (Bernhardt & Albright 1993): an actuator model, a kinematic model, a deformation model and a target model. The actuator model determines the mechanical relationship between the motors and joints of the robot, which affects the errors is reflected in pose deviations. The kinematic model describes the motion of the robot, and should be accurate in all configurations and at all velocities. The deformation model characterizes the compliance in the joints and links of the robot and takes care of the elasticity and backlash in the movements of the robot so that the robot model behaves correctly, including its dynamic properties. The target model specifies the TCP with respect to the flange of the robot. This relation remains quite stable during the lifecycle of the robot. The robot model is illustrated in Figure 15.

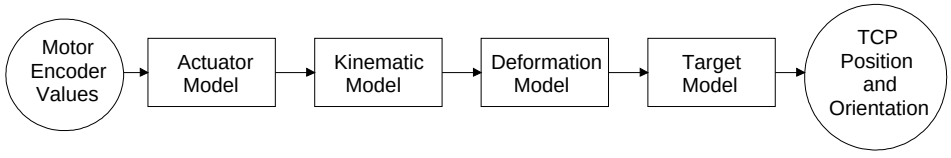


Figure 15. The robot model, according to Bernhardt & Albright (1993).

When modelling the spatial uncertainties of a robot system, the background of noise sources has to be ascertained. The model described above gives a short overview of what models are involved in controlling a robot system which is affected by uncertainties that can be observed by the user.

To model the kinematics of a robot system, the relation between the robot base frame and the wrist of the robot has to be determined somehow. The transformation from the base of the robot to its TCP consists of joints and links connected together to form a manipulator arm. To model the kinematics of the system, a method proposed by Denavit & Hartenberg (in Paul 1983) is used in this work. This enables the TCP to be calculated using a forward geometry model of the manipulator arm.

The transformation of a point between two coordinate frames in the robot system can be written as follows (Paul 1983):

$$H_i^i = H_i^{i-1}({}_i^m p) \tag{54}$$

where

H_i^i is the target coordinate frame,

H_i^{i-1} is the base coordinate frame of the joint coordinate frames,

${}_i^m p$ is a vector consisting of the parameters D–H, defining the relationship between joint i and point m.

The parameter vector can be written in Denavit–Hartenberg form as follows:

$$ {}^{m}_{i}p = [\theta, a, d, \alpha]^T \tag{55} $$

where

θ is the joint rotation angle,

a is the translation of a linear joint,

d is a constant linear transformation,

α is a constant rotational transformation.

The transformation between the robot base frame and the robot TCP frame consists of several transformations generated from links in the robot. When modelling the robot, these links are used to calculate the forward geometry. If each joint φ_i is written in matrix form with translation and orientation parameters in Euler angles, i.e. $\varphi_i = [x \;\; y \;\; z \;\; \phi_x \;\; \phi_y \;\; \phi_z]^T$, the geometry equations of the one specific robot system can be written as follows:

$$ \varphi_1 = \begin{bmatrix} -a_1 & d_1 & 0 & -\alpha_1 & 0 & -\theta_1 \end{bmatrix}^T \tag{56} $$

$$ \varphi_2 = \begin{bmatrix} -a_2 & 0 & 0 & 0 & 0 & -\theta_2 \end{bmatrix}^T \tag{57} $$

$$ \varphi_3 = \begin{bmatrix} -a_3 & 0 & 0 & -\alpha_3 & 0 & -\theta_3 \end{bmatrix}^T \tag{58} $$

$$ \varphi_4 = \begin{bmatrix} 0 & d_4 & 0 & -\alpha_4 & 0 & -\theta_4 \end{bmatrix}^T \tag{59} $$

$$ \varphi_5 = \begin{bmatrix} 0 & 0 & 0 & -\alpha_5 & 0 & -\theta_5 \end{bmatrix}^T \tag{60} $$

$$ \varphi_6 = \begin{bmatrix} 0 & 0 & -d_6 & 0 & 0 & -\theta_6 \end{bmatrix}^T \tag{61} $$

where

θ_i is the joint rotation angle of i:th joint,

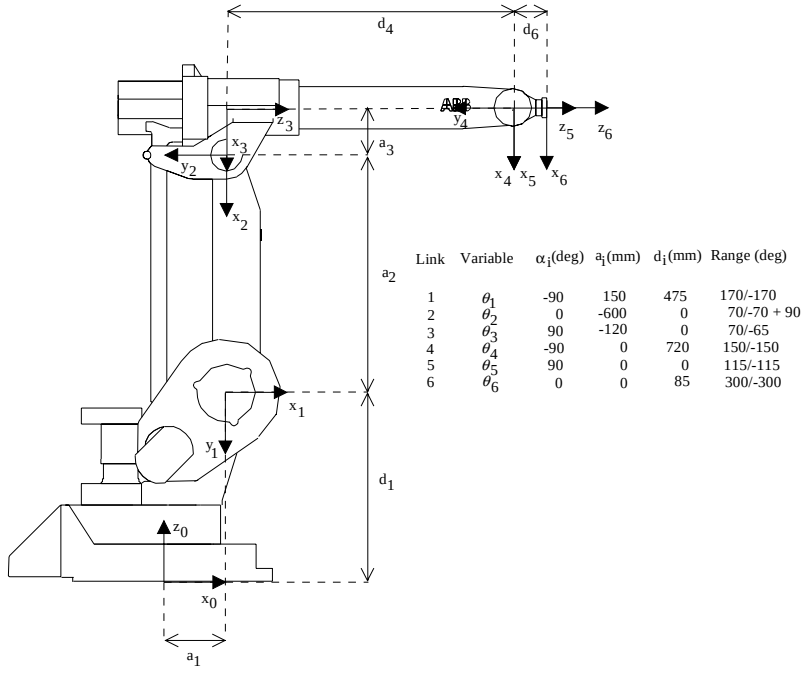
a_i is the translation of i:th linear joint,

d_i is the constant linear transformation of i:th link,

α_i is the constant rotational transformation of i:th link.

These transformations have been specified for the ABB IRB 1400 industrial robot used in some tests in this thesis. The matrix equations (56)–(61) will change in form when the kinematics of another robot manipulator is considered.

The transformations decribed above are illustrated in Figure 16 (Sallinen 2001):



Link	Variable	α_i(deg)	a_i(mm)	d_i(mm)	Range (deg)
1	θ_1	-90	150	475	170/-170
2	θ_2	0	-600	0	70/-70 + 90
3	θ_3	90	-120	0	70/-65
4	θ_4	-90	0	720	150/-150
5	θ_5	90	0	0	115/-115
6	θ_6	0	0	85	300/-300

Figure 16. The ABB IRB1400 robot with Denavit-Hartenberg link representation (Sallinen 2001).

The TCP value of the robot is calculated by transforming the equations (56)–(61) into a homogeneous transformation matrix and multiplying them.

The geometric model described in equations (56)–(61) represents the transformation from the robot base to the TCP of the robot. This model includes all the models presented previously, depending on how the parameters θ, a, d, α are defined, i.e. the actuator model, kinematic model, deformation model and target model. The uncertainties arising from these models are combined and the resulting uncertainty is close to Gaussian in its distribution.

5.2.4 Work object surfaces

Parametrized surfaces

We present here a means for identifying parametric surface forms and estimating the location of the surfaces in a coordinate, the surfaces being divided into three forms: plane, cylinder and sphere. All three forms are suitable for cases where only sparse data are available. The number of measurement points required for estimating each parameter model is roughly twice the number of parameters to be estimated, even in cases where the measurements include a lot of noise. The main attention in surface modelling is paid to simple, robust representation and the ability to include the estimation of spatial uncertainties in surface model parameters. Details and more information can be found in Sallinen & Heikkilä (2001b).

Planar surface

The simplest form of surface is a plane surface. This has three degrees of freedom, i.e. it is defined using translation from the origin to the plane and orientation around two axes, *x* and *y*, giving the parameters $m_{plane} = \begin{bmatrix} p_z & \phi_x & \phi_y \end{bmatrix}$. The surface model parameters can be estimated by a similar method to that used in hand-eye calibration and work object localization, LSQ Newton-Rhapson iteration. This means that the pose of the surface is estimated but it contains a different number of parameters depending on the form of the surface. Estimation requires an initial assumption for the parameters

to be estimated, as in the case of hand-eye calibration and work object localization. In the case of plane surface estimation, the error function is scalar, i.e. the shortest distance between the plane surface and the measured point:

$$e = -p_z \tag{62}$$

where

p_z is z-coordinate of the point in the plane frame.

When estimating the parameters of the surface model, the measured point is transformed from the measurement point frame, i.e. sensor frame, to the estimated surface frame and the error function is calculated in that frame. This is done in the case of a Reverse-Engineering approach, whereas in the other case, points are estimated directly in the work object frame, not by transforming them from the sensor frame. The coordinate frame of the plane surface is located on a surface of the plane, but the position is not determined. Transformation requires a knowledge of the hand-eye transformation and the robot base to TCP transformation as well as of the location of the work object. If the points are measured locally using points digitized from a CAD model, transformations of this kind do not need to be carried out, but instead the surface can be estimated in a local surface frame.

The Jacobian matrix for a plane surface is calculated from the error function with respect to the estimated parameters, equation (63):

$$\frac{\partial e}{\partial m_{plane}} = \begin{bmatrix} -1 & -p_y & p_x \end{bmatrix} \tag{63}$$

where

p_x, p_y are the measured point values in the workobject frame

$m_{plane} = \begin{bmatrix} p_z & \phi_x & \phi_y \end{bmatrix}$, estimated parameters.

The parametrization of a plane surface is illustrated in Figure 17.

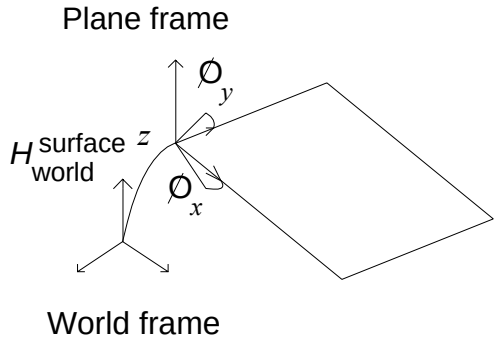


Figure 17. A plane surface described using three parameters (Sallinen & Heikkilä 2001b).

Cylinderical surface

The second form of parametric surface is a cylinderical one. The parameters of the model are estimated from a set of points distributed in the form of a cylinder or part of such a form. A priori information on the parameters to be estimated has to exist before estimation can be carried out. The cylinder surface is described using the position of the centre of the cylinder, the orientation of its centre line and its radius $m_{cyl} = \lfloor p_x \quad p_y \quad \phi_x \quad \phi_y \quad r \rfloor$. The coordinate frame is located in the middle of the cylinder and the height is set to an arbitrary value (Sallinen & Heikkilä 2001b).

The error function used to minimize the distance between the set of measured points and the reference model is as follows:

$$e = \sqrt{p_x^2 + p_y^2} - r \tag{64}$$

where

p_x, p_y are the measured points in the cylinder frame,

r is the estimated radius of the cylinder.

The partial derivatives for the estimation of the surface model parameters are constituted as follows:

$$\frac{\partial e}{\partial m_{cyl}} = \frac{1}{\sqrt{p_x^2 + p_y^2}} \begin{bmatrix} -p_x & -p_y & p_y p_z & -p_x p_z & -\sqrt{p_x^2 + p_y^2} \end{bmatrix} \qquad (65)$$

where

p_x, p_y, p_z are the measured points in the cylinder frame

$m_{cyl} = \begin{bmatrix} p_x & p_y & \phi_x & \phi_y & r \end{bmatrix}$ are estimated parameters for the cylinder pose.

As can be seen, the radius of the cylinder is estimated simultaneously with other parameters. The parametrization of the cylinder surface is illustrated in Figure 18.

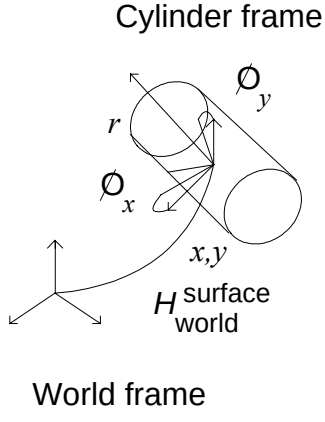


Figure 18. A cylindrical surface with estimated parameters (Sallinen & Heikkilä 2001b).

Spherical surface

Model parameters for a spherical surface can be generated from a point cloud in which the points are approximately in a spherical form. As in the previous

forms, a set of measured points are transformed from the sensor frame to the surface frame. The spherical surface is determined using the vector $\overline{m}_{sphere} = \begin{bmatrix} p_x & p_y & p_z & r \end{bmatrix}$, where r is the radius of the sphere and p_x, p_y, p_z are its centre points. The surface model is generated by an iterative process. The error function for estimating the surface model parameters of a sphere is as follows:

$$e = \sqrt{p_x^2 + p_y^2 + p_z^2} - r \tag{66}$$

where

p_x, p_y, p_z are the measured points in the surface frame.

The partial derivatives for the estimation of the surface model parameters are as in equation (67):

$$\frac{\partial e}{\partial m_{sphere}} = \frac{1}{\sqrt{p_x^2 + p_y^2 + p_z^2}} \begin{bmatrix} -p_x & -p_y & -p_z & -\sqrt{p_x^2 + p_y^2 + p_z^2} \end{bmatrix} \tag{67}$$

where

p_x, p_y, p_z are the measured points in the surface frame

$m_{sphere} = \begin{bmatrix} p_x & p_y & p_z & r \end{bmatrix}$ are the estimated parameters.

As in the case of the estimation of cylinder model parameters, the radius of the sphere is also estimated. The parametrization of a spherical surface is illustrated in Figure 19. The pose of the sphere includes only the position of its centre point and its radius, so that no orientation is described.

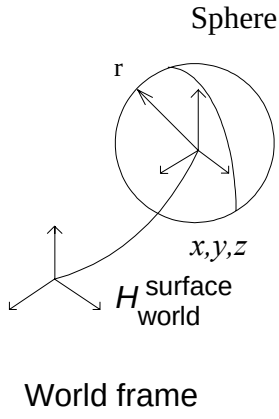


Figure 19. A spherical surface with estimated parameters (Sallinen & Heikkilä 2001b).

Free-form surfaces

B-spline surfaces using tensor products

The patched surfaces considered in this thesis are modelled using B-splines. The parametric forms illustrated were generated on the basis of estimated surface parameters, and patched surfaces by measuring a set of points on the surface of the work object using a robot and generating or selecting a patch to represent the surface. The size of patch will be selected by reference to the size of the whole surface and its gradients. The whole surface will be taken to consist of small local surface patches, i.e. the surface will be determined in terms of a grid of B-splines which describe its properties. The bicubic uniform B-spline patch $r(u,v)$ is expressed as a tensor product in equation 68, as given by Choi (1991):

$$r(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} N_i^3(u) N_j^3(v) V_{ij} \qquad (68)$$

where

$$N_0^3(u) = (1 - 3u + 3u^2 - u^3)/6 \tag{69}$$

$$N_1^3(u) = (4 - 6u^2 - 3u^3)/6 \tag{70}$$

$$N_2^3(u) = (1 + 3u + 3u^2 - 3u^3)/6 \tag{71}$$

$$N_3^3(u) = \frac{1}{6}u^3 \tag{72}$$

And the B-spline surface patch defined by 16 control vertices and expressed in matrix form, equation (73):

$$r(u,v) = UNBN^T V^T \tag{73}$$

where

$$\begin{aligned} U &= \begin{bmatrix} 1 & u & u^2 & u^3 \end{bmatrix} \\ V &= \begin{bmatrix} 1 & v & v^2 & v^3 \end{bmatrix} \end{aligned} \tag{74}$$

$$N = \frac{1}{6}\begin{bmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \tag{75}$$

The control vertices V_{00} to V_{33} are 3D points which define the surface patch. They form a control point net which can be evenly or non-evenly distributed and of a density that depends on the accuracy requirements. The size of the control net can vary depending on the accuracy requirements and the a priori information. A 4x4 size is used for work object localization as presented here.

The localizations of work objects with patched surfaces were calculated by Gunnarsson and Prinz (1987) using point to plane fitting, but this did not

produce such an accurate result as point-to-point fitting. The authors assumed the patched surface to consist of local plane surfaces, which can sometimes a very rough assumption, especially when the surface is changing rapidly. Point-to-point fitting operates well when the surface has large gradients. The novelty of the method presented here relative to that of Gunnarsson & Prinz is that it considers spatial uncertainties in the measuring system.

5.3 Modelling and estimating the spatial uncertainties in a robot system

When modelling the spatial uncertainties using stochastic models and a Bayesian-form noise distribution, some initial remarks should be made. One concerns the consistency of the system model and the validity of the set of samples for measurement. If these settings are not valid, e.g. the model parameters differ from the actually estimated ones, the estimates may biased. This is a common problem in noisy systems and attempts are made here to avoid this by planning the optimal measurements and weighted estimation.

5.3.1 Modelling and transforming spatial uncertainties

The definition of a homogeneous transformation between the joints and links of a robot is presented in previously in this chapter. When modelling noise in this thesis, errors are considered only for joint rotations θ_i and this error can be transformed into the coordinate systems of another link have the errors in the other link added to it etc. with the help of the other D-H parameters of the joint. Thus a formalization is needed for this transformation and accumulation of uncertainties.

Three means of modelling the spatial uncertainties between two coordinate frames are presented in the following, in each in which the spatial uncertainty is modelled as zero mean Gaussian-distributed noise, as first introduced by Järviluoma & Heikkilä (1994). The different forms of uncertainties are designated here as A, B and C and may be illustrated as follows:

A: The uncertain transformation is an additional transformation after the nominal one.

The spatial uncertainties are followed by a nominal transformation. This form is used when the nominal transformation is constant or very close to constant and the uncertainty appears after that transformation. The transformation can be written in matrix form as:

$$H(m)H(\delta) \tag{76}$$

where

$H(m)$ is the nominal transformation, including translation and orientation parameters

$H(\delta)$ is the uncertainty transformation

m is the nominal parameter vector

δ is a zero mean random pose parameter vector

B: The uncertain transformation is an additional transformation before the nominal one

The spatial uncertainty occurs before the nominal transformation. This form is used when the error source for the transformation is close to the origin of the coordinate frame. It can be written in matrix form as:

$$H(\delta)H(m) \tag{77}$$

C: The uncertain transformation is added directly to the parameters, and the orientations are expressed in the base coordinate frame before the nominal transformation

Here the uncertain transformation is included in the whole transformation and can be expressed in matrix form as:

$$H(m+\delta) \tag{78}$$

The form of presentation for spatial uncertainties also depends on the method used for estimating these. When using an Extended Kalman filter–type estimator, the results of uncertainties are given in form A or B depending on the initial assumptions. An example of the use of forms A and B is given in the work of Durrant-Whyte (1988). Generating a large number of samples by adding noise to nominal transformations gives form C uncertainties, a typical example of which is Monte Carlo simulation. Form C has also been used by Smith and Cheeseman (1986). Different forms of presenting a transformation with spatial uncertainties are illustrated in Figure 20 below.

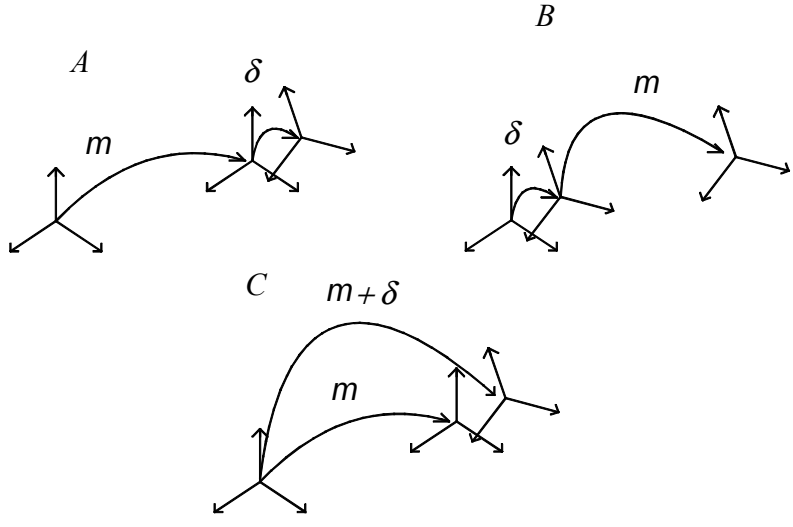


Figure 20. Forms of spatial uncertainty, modified from Järviluoma & Heikkilä (1994).

The spatial uncertainties in a robot workcell are presented in several coordinate frames, and to be able to evaluate the uncertainties of the whole system in one frame, they have to be propagated into a one compound coordinate frame. This requires the combination of uncertainties from different coordinate frames, which is done by transforming them from one form to another and combining the local uncertainties. The procedure is carried out here by transforming the uncertainties between different forms (A, B and C) so that they are represented in the same form in the coordinate frame in which they are summed together. The transformations between forms can be written as follows:

$$H(\delta')H(m) = H(m)H(\delta) \quad (\text{A -> B}) \tag{79}$$

$$H(m)H(\delta') = H(\delta)H(m) \quad (\text{B -> A}) \tag{80}$$

$$H(m+\delta') = H(m)H(\delta) \quad (\text{A -> C}) \tag{81}$$

$$H(m)H(\delta') = H(m+\delta) \quad (\text{C -> A}) \tag{82}$$

where

m is the vector of the pose parameters,

δ represents the uncertainties attached to the pose parameters,

δ' represents the transformed uncertainties attatched to the pose parameters.

These equations are obtained by deriving the transformations from one to another, e.g. A to B, A to C and C to A. As the transformations are non-linear, linearization is performed at $\delta = \overline{0}$. Let P be the covariance of a known coordinate frame in one of the three forms. The covariance P' in some other form then becomes:

$$P' = J_m^{m'} P J_m^{m'^T} \tag{83}$$

where

$J_m^{m'} = \frac{\partial \delta'}{\partial \delta}$ is the Jacobian matrix between the forms, computed at $\delta = 0$ from equations (61)–(64).

The transformations between the forms of uncertainties and the errors caused by the linearization are verified by Järviluoma & Heikkilä (1994) using Monte Carlo simulation. According to the test examples, the differences between transformation and Monte Carlo simulation were quite small, although transformation errors close to the singular point ($\phi = \pi / 2 + n\pi$) increased due to linearization. Järviluoma & Heikkilä (1994) studied only the diagonal terms of the error covariance matrix, which introduces uncertainties in the direction of the coordinate axis of the coordinate frame.

5.3.2 Propagating the uncertainties into the TCP frame

To combine the uncertainties arising from the joints of the robot, the noises in each joint have to be propagated into one frame. The noises are modelled in the form C, which is based on the principle of measuring the noises from the actual system by carrying out repeated tests. The uncertainty in each coordinate frame is represented using a 6x6 pose error covariance matrix, as follows:

$$P_{i,C} = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & {}_i\sigma^2 \end{bmatrix} \tag{84}$$

where

${}_i\sigma$ is the first standard deviation of joint *i.*

The noise in each joint is defined using pose error covariance, which means a total number of six coordinate frames with uncertainties in the form C. These uncertainties are combined by first transforming the consecutive transformations in form C to forms A and B, so that they represent the uncertainties in the same frame and the covariances can be summed together:

$$P' = P_i + P_j \tag{85}$$

where

i and j denote respective points.

The procedure is continued by transforming consecutive transformations into same coordinate frame and summing them together. Finally a situation in which all the uncertainty transformations are in one coordinate frame is achieved, and the outcome represents the transformation from the base of the robot to the TCP of the robot in form A. Due to the geometric properties of the coordinate transformations, the transformation from form A to form B is equal to the inverse of the transformation from form B to form A.

$$J_{AB} = J_{BA}^{-1} \tag{86}$$

where

J_{AB} is the Jacobian matrix for transformation A -> B, equation (77),

J_{BA} is the Jacobian matrix for transformation B -> A, equation (78).

Point uncertainty in a coordinate frame

There are several needs for evaluating the uncertainties of particular features in a coordinate frame. The feature considered here is a point which can be determined or measured almost anywhere around the work object. Consideration of the uncertainties attached to this point provides several advantages:

- The error covariance matrix includes values with only one unit
- The representation of a point is understandable
- Computation of the uncertainties is quite straightforward.

The motivation for using the same units in the error covariance matrix is strong. Almost in every case of estimation, uncertainties attached to different parameters are mixed together, the geometric interpretation for this being that the uncertainty ellipsoid has been rotated. If the uncertainty ellipsoid includes parameters with several units, computation of the uncertainties will not necessary give the correct result. This is because the parameters of the error covariance matrix have cross-relations with each other and the different units have different scales, which disturbs the result. By considering the uncertainties attached to a point in terms of the same units should enable reliable uncertainties to be achieved.

Representation of a point is quite easy to illustrate compared with an error covariance matrix with six degrees of freedom, and can be presented in a normal 3D space. The third advantage means that if the uncertainty of a coordinate frame is known, the uncertainty of any point in that frame can be calculated as follows:

$$P_p = J_p P_m J_p^T \tag{87}$$

where

P_m is the covariance matrix of the coordinate frame

$J_p = \frac{\partial p}{\partial m}$ is the 3x6 Jacobian matrix of the known point p evaluated around $m = 0$.

The Jacobian matrix J_p for the estimation of the uncertainties of a point in a coordinate frame can be written as follows:

$$\frac{\partial p}{\partial m} = \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \tag{88}$$

where

p_x, p_y, p_z are the coordinates of the point in the nominal coordinate frame.

5.3.3 Uncertainties in hand-eye calibration

To verify the goodness of hand-eye calibration, the uncertainties in the homogenous transformation from the TCP of the robot to the coordinate origin of the sensor must be determined. The uncertainties that affect this transformation come from the joint of the robot and the resolution of the sensor. In the following chapters, methods for modelling the spatial uncertainties using Bayesian form modelling is applied into hand-eye calibration, work object localization and estimation of surface model parameters.

When composing the weight matrix for simultaneous hand-eye calibration and pose estimation of a calibration plane, the input parameters for the system have to be considered. The weight matrices are calculated by linearizing the estimated parameter values around the nominal values. The input measurements for the system includes $m_i = \begin{bmatrix} j_1^i & j_2^i & j_3^i & j_4^i & j_5^i & j_6^i & d^i \end{bmatrix}$, i.e. the joint values and a sensor range for i:th pose, cf. Figure 16. When calculating the weight matrices, the TCP of the robot is calculated using the forward geometrics proposed in the previous chapter and then the input values are presented in a Cartesian frame, namely $m_{S,mea}^i = \begin{bmatrix} x & y & z & \phi_x & \phi_y & \phi_z & d \end{bmatrix}$. The noise matrix R consists of various measurements in Cartesian space, set as diagonal values to the noise matrix, including both robot TCP and sensor range values. These noise values are propagated based on noises in the joints in each pose.

$$R_i = \begin{bmatrix} R_{TCP} & 0 \\ 0 & R_{range} \end{bmatrix} \tag{89}$$

Each measurement produces a 7x7 matrix R_i and the total noise matrix has a size of i*(7x7), where i is the number of measurement samples.

Weight matrices for the measurements (measurements = values of the error function *e* of the measured samples) are derived from the original TCP pose and the range measurements by linearization around the nominal values for the estimated parameters. The Jacobian matrix which is here referred as K for the measurements is seen to be

$$K_h = \frac{\partial e}{\partial h} = \begin{pmatrix} \frac{\partial e}{\partial h_{TCP}} & \frac{\partial e}{\partial h_{sensor}} \end{pmatrix} \tag{90}$$

and the robot pose parameters, i.e. $\partial e / \partial h_{TCP}$, can be written as

$$K_{TCP} = \frac{\partial e}{\partial p_s} \frac{\partial p_s}{\partial h_{TCP}} \tag{91}$$

and further, on the first derivative of the weight matrix $J^e_{p_s}$ when the error function is $e_{plane} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} p_p$, in the form of equation (36).

$$K_{p_s} = \frac{\partial e}{\partial p_s} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{92}$$

the partial derivative K_h

$$K_h = \frac{\partial p_s}{\partial h_{TCP}} = \left(V^{plane}_{robot} \right)^T V^{tcp}_{robot} \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \tag{93}$$

where

p_x, p_y, p_z are the points in the TCP frame.

The input parameters for translation and orientation of the robot TCP pose are $h_{TCP} = \begin{bmatrix} x & y & z & \phi_x & \phi_y & \phi_z \end{bmatrix}$, and the respective value for the sensor measurement range h_{sensor} is d. The partial derivative matrix K_{sensor} for a sensor parameter, i.e. $\partial e / \partial h_{sensor}$, is simply

$$K_{sensor} = V_s \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \tag{94}$$

This is included in the uncertainties of sensor hand-eye calibration. The covariance matrix for the estimated state vector is obtained by multiplying the Jacobian natrix J_h^e by the weight matrix of the input parameters. Let R be this noise matrix, i.e. the weight matrix Q of the input parameters.

The weight matrix Q for the measurements K_h is then

$$Q = K_h R K_h^T \tag{95}$$

which is finally used in the estimator, equation (46).

The eye-in-hand calibration algorithm was implemented and convergence was tested by simulations with a varying number of measured points. The number of iterations needed was typically found to vary in the range 10–12, depending on the variability in the data and the initial values chosen.

5.3.4 Uncertainties in work object localization

Estimation of the spatial uncertainties in the location of a work object is the next phase after hand-eye calibration. When localizing the work object using a hand-eye calibrated sensor, the uncertainties of the calibration have to be taken into consideration. These are added to the joint noises and range measuring noise. As in the case of sensor hand-eye calibration, the noise in the robot joints is propagated into the TCP noise in a Cartesian frame.

The noise matrices for the work object localization include the noise arising from the joint of the manipulator, the noise of the sensor measurement and sensor hand-eye calibration, i.e. the uncertainty in the transformation from the TCP to the origin of the sensor. The noises from these sources are placed in a block diagonal matrix similar to that used in the case of sensor hand-eye calibration:

$$R = \begin{bmatrix} R_{TCP} & 0 & 0 \\ 0 & R_{range} & 0 \\ 0 & 0 & R_{hand-eye} \end{bmatrix} \tag{96}$$

Each measurement produces 13x13 matrices, and the total size of the matrix is i*13x13, where i is the number of measurement points.

The partial derivative of the error function with respect to the input parameters is used to calculate the weight matrices:

$$\frac{\partial e}{\partial m_{in}} = \frac{\partial e}{\partial p}\frac{\partial p}{\partial m_{in}} \tag{97}$$

where

m_{in} are the input values in Cartesian form, i.e.

$m_{in} = \left[\left(T_{TCP}, V_{TCP}\right), d_{range}, \left(T_{hand-eye}, V_{hand-eye}\right) \right]$ input parameters.

The partial derivative $K_{TCP,range}$ of the error function $e = n.p - l$ with respect to the pose parameters is as follows:

$$\frac{\partial e_S}{\partial m_{TCP,range}} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} \left(V_{robot}^{plane}\right)^T V_{robot}^{tcp} \begin{bmatrix} 1 & 0 & 0 & 0 & p_{s,z} & -p_{s,y} \\ 0 & 1 & 0 & -p_{s,z} & 0 & p_{s,x} \\ 0 & 0 & 1 & p_{s,y} & -p_{s,x} & 0 \end{bmatrix} V_{tcp}^{sensor} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{98}$$

where

p_s is the measured point in the TCP coordinate frame,

$n_{x,} n_y, n_z$ are components of the reference surface normal vector,

V is the rotation matrix.

In addition, there are uncertainties coming from the sensor hand-eye calibration that also have to be considered. The partial derivative $K_{hand-eye}$ for this can be written as follows:

$$\frac{\partial e}{\partial m_{hand-eye}} = \begin{bmatrix} n_x & n_y & n_z \end{bmatrix} \left(V_{robot}^{plane}\right)^T V_{robot}^{tcp} V_{tcp}^{sensor} \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \quad (99)$$

where

p is the measured point in the wrist frame.

The weight matrix Q is calculated using the equation (93).

5.3.5 Uncertainties in surface models

The uncertainties in the work object surface parameters may also need to be taken into consideration when modelling the spatial uncertainties in a robot workcell. There are many cases in which there is no CAD model available of the work objects to be localized. In those cases the surface parameters have to be estimated in order to obtain a CAD model and verify its accuracy. Applications in which very high accuracy is needed, e.g. surface inspection or assembly, have to take the spatial uncertainties between the object inspected or assembled and the target object into consideration.

Uncertainties can appear in surface models from different sources. When a surface model is generated from a CAD model, the accuracy of the latter determines the uncertainties of the surface, but even if a CAD model is available, the manufacturing accuracy of the actual surface increases the surface inaccuracies and the result differs from the CAD model. The Reverse Engineering approach, in which surface models are generated from the actual surface, means that uncertainties arise from the accuracy of the measuring system. Such systems are usually highly non-linear, but their uncertainties have somehow to be propagated into single point uncertainties, which are used to estimate surface uncertainties. When surface model parameters are estimated from a set of points, the number of points affects these uncertainties. As in the previous estimation cases, the level of uncertainties can be reduced in most cases by increasing the number of measurement points.

Uncertainties in surface models affect the accuracy of work object localization, as has been shown by Sallinen and Heikkilä (2001). Not least for this reason, levels of uncertainty in surface models should be kept low. If the noise level of surface uncertainties is relatively high, the uncertainty of work object localization will increase dramatically.

Parametric surfaces

Equations for estimating the spatial uncertainties attached to surface parameters are presented in the following subchapters. The noise introduced into the estimation is modelled as the noise attached to the measured points. Depending on the form of the particular surface, the points are distributed in a manner of a plane, cylinder or sphere.

Planar surface

Estimation of the spatial uncertainties in planar model parameters is based on modelling the noise of the input parameters of the system and first propagating the uncertainties and then combining them into one uncertainty matrix. The weight matrix K includes partial derivatives of the error function e with respect to the measured parameters m, equation (82).

$$\frac{\partial e}{\partial m_p} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} V_{plane}^{world} V_{world}^{tcp} \tag{100}$$

where

$m_p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}$, includes the position of the measured point in TCP frame

e is the error function, $e = p_z$

V_{plane}^{world} is the rotation matrix from the world frame to the plane frame

V_{world}^{tcp} is the rotation matrix from the TCP frame to the world frame.

When estimating model parameters and spatial uncertainties for the various surface forms, the noise matrix R_i must also be considered. Here the input parameters for estimation are the uncertainties attached to a point, so that the size of each R_i will be 3x3. The total noise matrix will consist of a number of R_i matrices equal to the number of points used in estimation.

Cylindrical surface

The weight matrix K for estimating uncertainies in the model parameters for a cylindrical surface is a partial derivative of the error function with respect to the measured parameters:

$$\frac{\partial e}{\partial m_p} = [dx \quad dy \quad 0] V_{cyl}^{world} V_{world}^{tcp} \tag{101}$$

where

$dx = \frac{\partial e}{\partial x}$ is a partial derivative of the error function with respect to parameter x

$dy = \frac{\partial e}{\partial y}$ is a partial derivative of the error function with respect to parameter y

$e = \sqrt{p_x^2 + p_y^2} - r$ is the error function for the estimation of cylindrical model parameters.

Spherical surface

The weight matrix K for estimating the uncertainty in spherical surface model parameters is calculated from the error function respective to the measured parameters.

$$\frac{\partial e}{\partial m_p} = \begin{bmatrix} dx & dy & dz \end{bmatrix} V_{cyl}^{robot} V_{robot}^{tcp} \tag{102}$$

where

$dx = \frac{\partial e}{\partial x}$ is a partial derivative of the error function with respect to the parameter x

$dy = \frac{\partial e}{\partial y}$ is a partial derivative of the error function with respect to the parameter y

$dz = \frac{\partial e}{\partial z}$ is a partial derivative of the error function with respect to the parameter z

$e = \sqrt{p_x^2 + p_y^2 + p_z^2} - r$ is the error function for the estimation of spherical model parameters

Patched surfaces

The uncertainties in surface patches are modelled using the uncertainties attached to a single point, but unlike parametric models, uncertainties are not estimated iteratively when modelling the surface itself. Instead, they are considered when the work object with a B-spline surface is localized, by including the surface uncertainties in the combined uncertainties of the system.

Figure 21 shows a 2D B-spline curve determined using three points. Each point has a nominal value and an uncertainty, which is illustrated as an ellipsoid around it. The forms of the ellipsoids are determined based on the uncertainties in the points. The orientation and area of an uncertainty ellipsoid depends on the input uncertainties of the system.

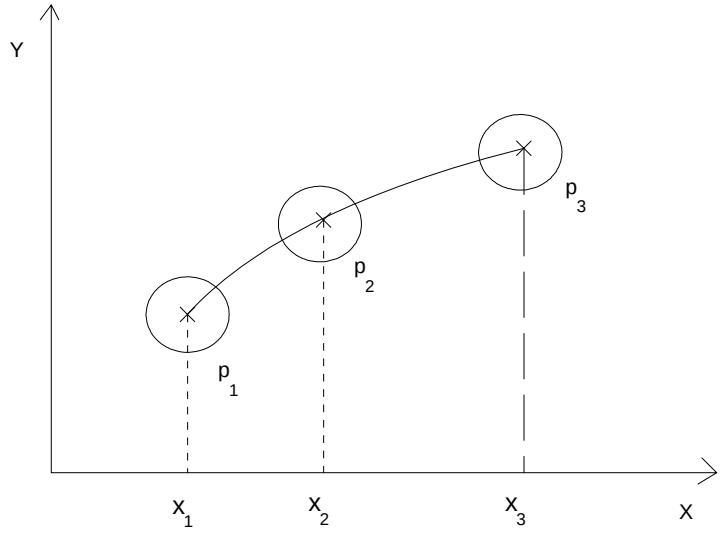


Figure 21. A 2D spline curve determined using three points and their uncertainties.

According to this assumption, each point on the B-spline curve in the coordinate frame can be determined as a nominal point with Gaussian-distributed zero mean noise, as follows:

$$p_1 = p_1^n + \delta p_1 \tag{103}$$

$$p_2 = p_2^n + \delta p_2 \tag{104}$$

$$p_3 = p_3^n + \delta p_3 \tag{105}$$

where

p_1^n is the nominal value of a point

δp_1 is the uncertainty of each point around its nominal value.

The density of a B-spline surface patch has to be selected depending on its accuracy requirements. By increasing the number of points determining the surface, the uncertainty of the surface will be reduced. These methods have been

developed for a sparse set of points, however, and it is assumed that they do not have to be selected too close to each other. Expanding this principle to the 3D case, each point requires 3D uncertainty, describing its spatial uncertainty. An uncertain surface patch can be illustrated in terms of Figure 22, in which there is a nominal transformation with a nominal surface and an uncertain transformation with an uncertain surface:

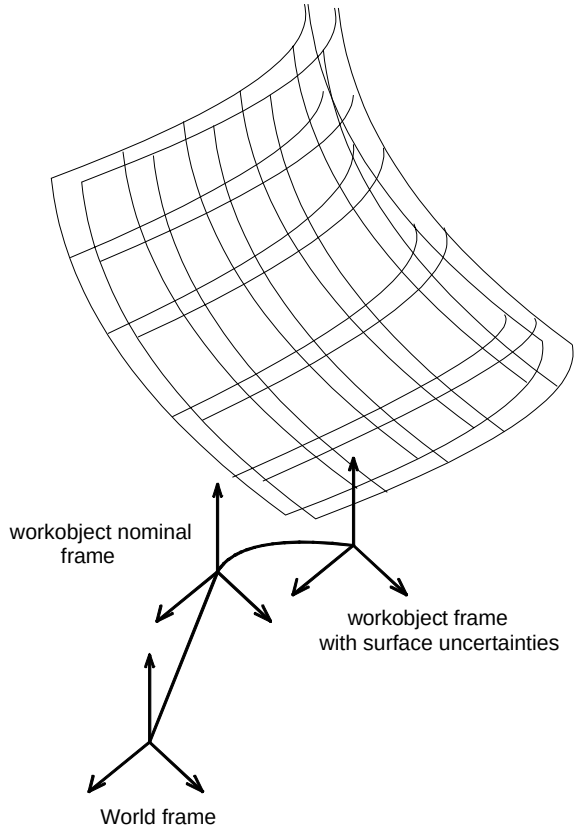


Figure 22. Coordinate frames in the B-spline surface model.

Estimation of spatial uncertainties in a B-spline surface patch

The method presented here assumes that uncertainties in the measurement process are stochastic, i.e. they have a noise value in each pose that is approached. To model robot uncertainty in different poses, a joint-based noise

model is used which is based on Denavit-Hartenberg notation, setting the noise for each joint and propagating these into pose 6 DOF uncertainties as decribed in the previous chapter. These pose values are used to estimate the remaining uncertainty for hand-eye calibration of the sensor. In the same way, the pose uncertainty is combined with work object pose estimation and surface treatment.

In the estimation of spatial uncertainties for estimated parameters, it is possible to calculate partial derivatives for the input parameters by defining the weight matrix Q. Again, let K be a block diagonal matrix in which blocks $_K$'s are partial derivatives of the error $_e$ with respect to the errors m_e that affect the measuring. Each block $K_1,\ldots,K_N$ includes a partial derivative of one measurement and a partial derivative of the sensor tool correction. In the diagonal block, one K_N represents one block and contains these partial derivatives. The error derivative of e with respect to the measuring errors of the measured values m_e becomes:

$$\frac{\partial e_i}{\partial m_e} = \frac{\partial e_i}{\partial p_i^m}\frac{\partial p_i^m}{\partial m_e} \tag{106}$$

An error function for computation of a weight matrix can be obtained by localizing the patched surface, i.e. $e = p_{ref} - p_m$ and the measured point with respect to the measured values for pose of the robot ($m_{e,s} = \lfloor x \quad y \quad z \quad \phi_x \quad \phi_y \quad \phi_z \quad r \rfloor$, where the first six are pose parameter values and r is the sensor distance value) and the uncertainty of the points ($m_{e,p} = \lceil p_x \quad p_y \quad p_z \rceil$).

The error derivative for spline surface uncertainties can be taken into consideration based on point uncertainties in the spline path using the equation 105.

$$\frac{\partial e_i}{\partial m_{e,p}} = \left(V_{robot}^{plane}\right)^T V_{robot}^{tcp} \tag{107}$$

$V_{robot}^{plane}\ V_{robot}^{tcp}$ are rotation matrices with respect to the coordinate transformations.

In addition to the surface uncertainties, the combined error derivative for the robot and sensor uncertainties is, using the equation 106:

$$\frac{\partial e_i}{\partial m_{e,s}} = \left(V_{robot}^{plane}\right)^T V_{robot}^{tcp} \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y & \\ 0 & 1 & 0 & -p_z & 0 & p_x & V_{tcp}^{sensor} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ 0 & 0 & 1 & p_y & -p_x & 0 & \end{bmatrix} \quad (108)$$

where

V_{tcp}^{sensor} is rotation matrix from tcp to sensor frame

p_x, p_y and p_z are points in the TCP coordinate frame.

In the case of work object localization, the estimated parameters $m_{e,w-o}$ are $m_{e,w-o} = \lfloor x \quad y \quad z \quad \phi_x \quad \phi_y \quad \phi_z \rfloor$. The form of the error derivative is similar to that of equation (96), and the error derivative can be written as

$$\frac{\partial e_i}{\partial m_{e,wo}} = \left(V_{robot}^{plane}\right)^T V_{robot}^{tcp} \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y & \\ 0 & 1 & 0 & -p_z & 0 & p_x & V_{tcp}^{sensor} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ 0 & 0 & 1 & p_y & -p_x & 0 & \end{bmatrix} \quad (109)$$

where

V_{robot}^{plane} V_{robot}^{tcp} V_{tcp}^{sensor} are rotation matrices with respect to the coordinate transformations,

p_x, p_y and p_z are points in the TCP frame.

To model the noise in the Bayesian system, R is set as a block diagonal matrix in which $R_1, \ldots, R_N$ include the covariance matrices of the different spatial

uncertainty sources in the measuring system. R_1 includes the variances of the robot pose parameters and the variances of the measured point coordinates together with the sensor covariance matrix. In the case of work object localization, R_1 also includes the error covariance matrix of sensor calibration. The whole noise matrix R is formed by setting the R_1 matrices in the diagonal block of the matrix R as many times as there are measurements in the measurement data. For a different point, there are different uncertainties in robot uncertainty due to the robot pose uncertainty calculation.

Covariance matrices Q for the noise of the error K_m can be calculated using equation (93), presented in the previous chapter, i.e. multiplication of the noise matrix by the weight matrix.

5.3.6 Combining estimates and uncertainties from different measurements

Fusion using the Fisher Method

The Fisher method is used for fusing the measurements and their respective uncertainties in the present work. The requirement for sensor fusion arises from the flexibility and computational limits of the current system. When using a laser rangefinder with a laser stripe, one measurement produces a large number of measurement points. To use all the measurements effectively, information obtained from all the points has to be used. Because computation of the Jacobian and weight matrices for several thousands of points at one time is too laborious a task, the measurement data can be divided into several sets and estimation can be carried out in several steps. In addition, the system is easy to expand to include new sensors providing information about the environment.

The Fisher method has been used for local estimation purposes by Luo and Lin (1988), together with a Bayesian model for global estimation, see Figure 23:

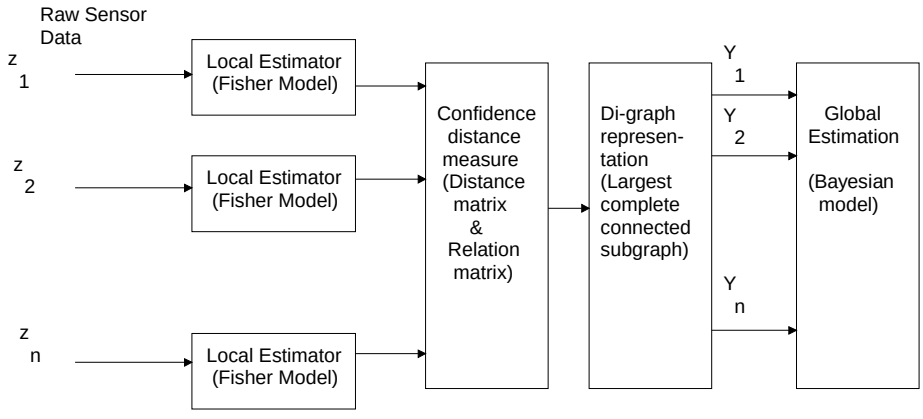


Figure 23. Functional block diagram of multi-sensor fusion, after Luo & Lin (1988).

The difference between the Fisher model and Bayesian modelling, however, is that Bayesian modelling uses initial information for weighting the estimation (the a posteriori error covariance matrix P). This gives an initial direction for the uncertainties, which are usually quite close to the initial value when considering the same system. Luo and Lin use the a posteriori error covariance obtained from Fisher estimation in Bayesian estimation, which allows more reliable results to be obtained, as Fisher estimation filters out most of the noise in the system and Bayesian estimation can use fairly reliable information.

First, the set of measurement data is divided into smaller amounts, of a size that can be computed at one time. The data for each estimation have to be selected carefully, as they have to be versatile enough to allow the parameters for coordinate transformation to be chosen. This may be a problem when estimating a full transformation with 6 DOF and using information on single points in a sensor frame. Even if the point information in the sensor frame includes location in 2D (y,z values, where y is the width of the point in the laser stripe and z the distance value), the estimate can very easily become biased. Using the parameter estimation methods presented in previous chapters, an estimate and its respective error covariance matrix can be calculated, and this procedure is repeated until all the measurements have been used for computing local estimates. After that fusion phase follows in which the previously calculated sub-results are

combined. The estimates from two measurements can be combined using the following equation:

$$m = P_{tot} P_c^{-1} m_{estimate,c} + P_{tot} P_{tot,c-1} m_{estimate,c-1} \tag{110}$$

where

P_{tot} is the combined covariance of the new measurement and previous covariance

P_c is the covariance of the current step

$P_{tot,c-1}$ is the total uncertainty from the previous iteration step

$m_{estimate,c}$ are the estimates for the parameters

$m_{estimate,c-1}$ are the estimates for the parameters from the previous iteration step.

The combined covariance P_{tot} can be calculated by combining the error covariances of the current parameters and the error covariance matrix of the total uncertainties for all the previous measurements:

$$P_{tot} = (P_c^{-1} + P_{tot,c-1}^{-1})^{-1} \tag{111}$$

When fusing state parameters and uncertainties from different sensors, one has to be very careful with the forms of spatial representation (A, B or C). Error covariances that are to be combined have to be in the same form.

A recursive algorithm for fusing measurements and uncertainties

Given equations can be written in the form of a recursive algorithm in which the different steps are explained. This can mainly be divided into two phases, estimation of parameters and their uncertainties and fusion of the estimates and

their uncertainties. The parameter estimation is similar to that presented in the previous chapter. The recursive algorithm works as follows:

1 Set the initial value for parameters m_S to be estimated and their covariance P_{c-1}.

2 Take every n:th point from each measurement pose, in such way that the matrices in the calculation will not become too small or too large.

3 Calculate the correction Δm for the estimated parameters.

4 Update the estimated parameters m_S using the correction Δm .

5 If the corrections Δm_S are close enough to zero, go to the next phase, otherwise go to step 3.

6 The estimates for the parameters are now in the vector m_S. Calculate the covariance matrix P_c.

7 Reiterate steps 1–6 until all the data have been processed. There should then be n values for the estimates m_S and n values for the respective covariances P_c.

8 Calculate the total covariance P_{tot} using equation (109) and the respective estimates. Continue calculation until all the estimates and covariances have been processed.

9 The resulting P_{tot} and estimates m are the error covariance matrix and estimates for the estimated parameters.

The final estimate m and respective covariance P_{tot} can be upgraded by using them as the previous estimate $m_{estimate-1}$ and covariance $P_{tot,c-1}$ and fusing them with new data. When combining new data with the estimates, one nevertheless has to be careful not to include very biased data, especially with low-level uncertainties attached, because this will have quite a large effect on the final estimate.

5.4 Planning of the measurements

This paragraph presents a new synthesis method for sensing planning based on minimization of an a posteriori error covariance matrix and eigenvalues in it. This latter consists of a multiplication of the Jacobian matrix by the weight matrix, and minimization means here a manipulation of the terms in the Jacobian and weight matrices to achieve a low level of spatial uncertainties. The minimization of error covariance matrix and effect of signal-to-noise ratio is illustrated in simulations, where the location of the measurement points in the surface of the workobject effects to the pose uncertainties. The Jacobian and weight matrices may belong to any phase of the calibration of the robot system, and planning is carried out for each phase separately. The planning algorithm presented here is based on methods for modelling spatial uncertainties put forward in this thesis, and relies on interpretation of the error covariance matrix.

5.4.1 Error covariance matrix and SNR

A novel method for sensing planning is presented here based on a posteriori uncertainties in the estimated parameters. The methods for estimating the spatial uncertainties presented in this thesis form the groundwork for using the error covariance matrix as a criterion for sensing planning and as a basis for evaluating and modelling spatial uncertainties in a robot system. The existing methods for using a posteriori error covariance matrices as evaluation criteria are suitable for many cases, but they have following shortcomings: Borghi & Caglioti (1998) used the method only in a 2D case and based it on the calculation of range distances between a robot and a surface. This kind of criteria formation is not possible in hand-eye calibration, for instance, because the solution space is more complicated. Nilsson & Nygårds (1996) also used geometric distances and the respective uncertainties as criteria for sensing planning. These methods were designed for mobile robots and are more suitable for purposes in which there are usually 3 degrees of freedom and the minimization criteria can be composed based on simple geometric features of the environment.

The planning is focused on selecting reference features from the measurement pose. The poses itself is assumed to be given, and optimization considers the

features that will be measured. The goal is to minimize the uncertainties in an estimated pose and the selection of measurements that especially affect the rotation uncertainties is considered. As will be illustrated later, the estimation of translation parameters is not so sensitive to quality of measurements as the estimation of the orientation parameters of the pose. Here the minimization and maximization of matrices means selecting of values for parameters in matrix rows and columns. The goal of the selection is to achieve low level of uncertainties in terms of eigenvalues of the error covariance matrix. To simplify the evaluation of uncertainty criterias, multiplication and sum of all eigenvalues is used which is illustrated in simulations further in this chapter.

Computation of the covariance

The method presented here is based on minimizing the a posteriori error covariance matrix of the estimated parameters when using Bayesian-form modelling of the spatial uncertainties. This means that the sensor has to be placed in such poses that the noise in the estimated parameters will be minimized. In addition to noise, there are a lot of unmodelled error sources which cannot be adjusted. When using Bayesian-form estimation with weight matrices, the a posteriori error covariance matrix is used as a minimization criterion for planning the set of samples.

The following error sources have to be considered in the minimization of the spatial uncertainties in the system, depending on the task: noise in the joints, noise in the sensor, hand-eye calibration of the sensor, work object localization, surface uncertainty and point uncertainty. When generating the set of samples, i.e. when calculating the matrix P, these error sources are taken into consideration in matrices J and Q equation (110).

The error covariance matrix P of the estimated parameters can be calculated as follows:

$$P = (J^T Q^{-1} J)^{-1} \tag{112}$$

where

J is the Jacobian matrix for estimating the model parameters

Q is the weight matrix of the measurements.

The error covariance matrix P consists of the product of the Jacobian matrix J and the weight matrix Q. According to equation (110), the minimization leads to a situation in which J has to be maximized and Q has to be minimized. These matrices share the same parameters, however, so that minimization of P is not so straightforward. It may be carried out in two ways: by measuring optimal points for localization, i.e. each row in the Jacobian matrix will give a large amount of information and there are only a few rows, or else the number of rows in the Jacobian matrix can be increased but each row will give only a little information.

The control parameters of the sensing planning system presented here are parameters of the Jacobian matrix J and weight matrix Q. These may include (x,y,z) point values of the calibration plate frame in hand-eye calibration, the range of measurement in the sensor frame, point values on the work object surface or the orientation of the TCP of the robot, for example. In the simple examples of planning measurements for line estimation and work object localization later in this section, matrices for control parameters are illustrated.

Signal-to-noise ratio

Each measurement includes a measurement point p with noise δp. This point is measured in the sensor frame and may be transformed into the world or work object frame using a known geometrical transformation with spatial uncertainty. The noise is assumed here to be constant in the parameter space, and the goal is to achieve a high signal-to-noise ratio.

When estimating the pose parameters of the line or the work object, assuming that the noise is constant in the working space, the aim is to obtain a large signal. The signal is here determined as a distance between the parameters of two sequential lines in Jacobian matrix in the case of maximizing J and two sequential lines of partial derivative measurement matrix K when minimizing weight matrix Q. Noise in the signal is noise of the system, i.e. matrix R. SNR

is not used here as a direct cost function for selecting of control points, instead, as a ground for maximizing the Jacobian and minimizing the weight matrix when the noise of the system is known. The simulations for dependence between distance of points and uncertainties are illustrated in later in this chapter.

Computation of the criteria

The principle of the sensing planning method is illustrated as simple example. The line is descibed in coordinate frame in Figure 24 below:

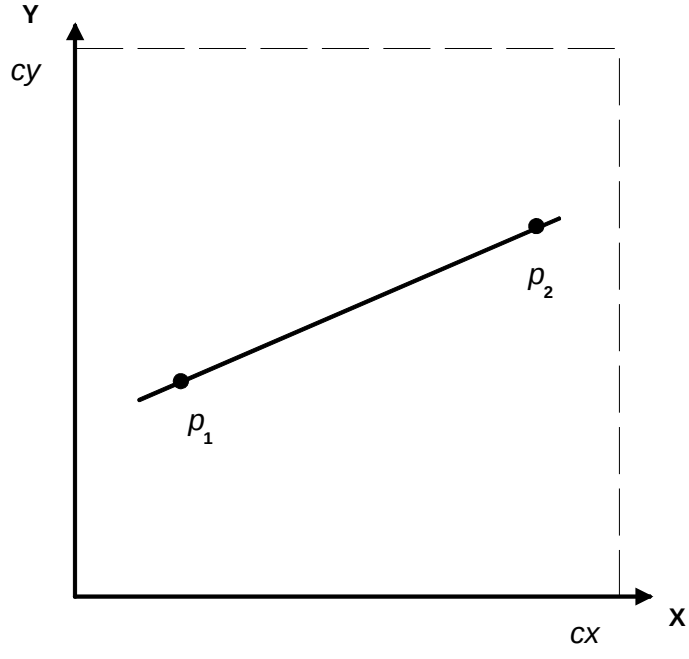


Figure 24. Estimation of the location of a line in a coordinate frame.

Estimation of a line in (XY) space using two points p_1 and p_2 is illustrated in Figure 24. The solution space is constrained using two parameters in X and Y direction, c_x and c_y such that $0 < p_x < c_x$ and $0 < p_y < c_y$. The objective is to localize a line such that the pose uncertainties will be minimized. The pose of

a line is described using the orientation between x-axis and a line and shortest distance from the origin, i.e.

$$m_{line} = \begin{bmatrix} \alpha & d \end{bmatrix} \tag{113}$$

where

α is a angle between x-axis and a line

d is a shortest distance between a line and an origin.

The error function is a shortest distance of a point from the line,

$$e_{line} = \begin{bmatrix} -x\sin\alpha + y\cos\alpha - d \end{bmatrix} \tag{114}$$

where

x, y are the coordinates of a measurement point

When planning the set of measurements, the parameters to be estimated have to be considered first. The partial derivatives of the error function with respect to the estimated parameters comprise the parameters that affect the spatial uncertainties of the estimated parameters. The error function is the same as that used in parameter estimation, e.g. estimation of the pose of the work object. Estimation of rotation parameters requires at least a pair of points, which leads to a situation in which the distance between these points is maximized in order to achieve a high SNR and this is formulated as a max problem. Based on these parameters, the set of samples will be generated by selecting the maximum values of the composed parameters, taking into consideration the constraints of the parameter space. Thus the equation for generating a set of samples based on a Jacobian matrix can be written as follows:

$$\Psi_J = \max\left\{\frac{\partial e}{\partial m}\right\}_{\min=c1}^{\max=c2} \tag{115}$$

where

e is the error function

m are the estimated parameters

c_2 is the upper constraint value of the parameter space

c_1 is the lower constraint value of the parameter space.

The constraint values for equation (113), c_1 and c_2, are usually set by the dimensions of the parameter space, as in example of localizing a line, c_x, c_y. In the actual robot system this means the dimensions of the working range of the robot, the measurement tool and the work object, e.g. the lengths of the sides of the work object.

Maximizing the Jacobian matrix J in a case of localizing of a line means maximizing distance of sequential point pairs of Jacobian matrix, as follows:

$$J = \begin{bmatrix} -x_1 \cos\alpha - y_1 \sin\alpha & -1 \\ -x_2 \cos\alpha - y_2 \sin\alpha & -1 \\ \vdots & \vdots \\ -x_{n-1} \cos\alpha - y_{n-1} \sin\alpha & -1 \\ -x_n \cos\alpha - y_n \sin\alpha & -1 \end{bmatrix} \tag{116}$$

where

n is the number of points.

It means the control parameters x and y have to be manipulated to maximize the distance between the measurement points, i.e. sequential lines of Jacobian matrix. Signs of control parameters does not need to be considered because covariance matrix P includes multiplication of Jacobian matrix $J^T J$ and the result is always positive. The number of solution spaces depends on the amount of parameters in Jacobian row. In the example above, there are two parameters and it will give two different point pairs: $(x \approx 0, y \approx c)$ and $(x \approx c, y \approx 0)$.

In general case, the Jacobian matrix $\partial e/\partial m$ determines the control parameters and may include position and orientation parameters from the world frame or work object frame as well as the sensor range or other internal or external parameters of the sensor. The composed position parameters from the partial derivative $\partial e/\partial m$ are maximized by selecting their largest possible values. The selection of rotation maxima and minima means selecting large and small angles for each axis. That is the case when the Jacobian matrix includes rotation matrices, e.g. hand-eye calibration. Orientation parameters are planned directly from the rotation matrices, not by computing partial derivatives from the Jacobian matrix. The orientation matrices that affect to the uncertainties are composed from Jacobian matrix and weight matrix. By maximizing the composed parameters the highest SNR can be obtained for the measurements. Each dimension must be considered when selecting the points, and both its minimum and its maximum must be set. If there are several dimensions, all of them have to be taken into consideration.

The idea is based on the proposal of Lowe (1985) that a small rotation around different axes can be projected as a small translation in a perpendicular axis. Using this principle, the set of samples is generated by maximizing the distances between the points.

Effect of the weight matrix

The weight matrix Q also has to be studied when considering the optimal positions and orientation. The weight matrix consists of partial derivatives of the error function with respect to the measurements K and their noise matrix, as follows:

$$Q = KRK^T \tag{117}$$

where

K is a matrix consisting of partial derivatives of the error function with respect to the input parameters

R is a matrix consisting of the noise affecting the input parameters.

If the noise matrix R is assumed to be constant, then the goal will be the minimize the weight matrix K, which can be written in partial derivative form:

$$K = \frac{\partial e}{\partial m_{input}} \tag{118}$$

where

e is the error function

m_{input} is the vector containing the measurement parameters.

As in the case of computation of the criteria, the error function depends on the type of surface form and the measurement parameters of the robot system. The error function is the same in equations (113) and (117), however. The partial derivatives for the weight matrix can be written:

$$\Psi_K = \min\left\{\frac{\partial e}{\partial m_{input}}\right\}_{\min=c1}^{\max=c2} \tag{119}$$

Optimizing both equations (113) and (117) is not so simple, due to the interdependence of the parameters between the equations. Of the two, equation (113) has the larger effect on the estimation process, due to the double terms in the Jacobian matrix J compared with the weight matrix Q in equation (110). The effect of equation (117) becomes noticeable when the noise of the input parameters is not linear.

Minimization of weight matrix Q is similar with maximizing the Jacobian matrix J. In the case of weight matrix, also the point pairs of sequential rows are selected and distance between them are minimized. The weight matrix includes multiplication of matrices K such that sign does not have to be considered (KK^T).

The effect of equation (117) on work object localization can be seen when there the error covariance of the sensor and the covariance of the pose of the robot are included. The uncertainty attached to each measurement is weighted by reference to the sensor hand-eye calibration uncertainty. The orientation of the

sensor should be chosen based on the form and orientation of the uncertainty ellipsoid of the measuring point in the calibrated sensor frame.

When forming the total measurement matrix Ψ, the information obtained from both the Jacobian measurement matrix Ψ_J and the weight measurement matrix Ψ_K has to be considered. In most cases minimization and maximization focus on the same parameters in the work space, e.g. points or angles of the measuring tool. In that case, planning is quite straightforward and dimensions of these parameters can be maximized and minimized. In the case of hand-eye calibration, for example, the measurement poses are set such that the range values of laser rangefinder are minimum and maximum values and the orientation of the TCP of the robot has both large and small angles. In the case of work object localization, measurement poses are selected such that they are some distance away from the origin of the work object in each direction. The parameters have first to be composed, however, and after that minimized and maximized. If these dimensions share the same size class, the uncertainty ellipsoid should be close to a sphere, but when there are large differences in the dimensions, the uncertainties will be much larger in some directions than in others and the uncertainty ellipsoid will be wide but flat in form.

There are more sources of uncertainty in the localization of the work object, than in hand-eye calibration. The total criteria can be found by first transforming the error covariance matrices into one coordinate frame and then combining them. There are always one or more error covariance matrices which are assumed to be constant and one which is to be minimized, e.g.:

$$P_{total} = P_{TCP} + P_{hand-eye} + (J_{wo} Q^{-1} J_{wo}^T)^{-1} \tag{120}$$

where

P_{total} is the total uncertainty in the system

P_{TCP} is the error covariance matrix for the TCP of the robot

$P_{hand-eye}$ is the error covariance matrix for the hand-eye calibration

$(J_{wo} Q^{-1} J_{wo}^T)^{-1}$ is the a posteriori error covariance of the pose of the work object.

Otherwise, if the error covariance matrices are not assumed to be constant, each error covariance will have to be mininized and divided into separate terms, cf. equation (110). When expanding further on this principle, each pose has its own uncertainty for joints and sensor uncertainty. By minimizing the uncertainties of each pose of the robot, including the sensor measurement uncertainty and the uncertainty arising from hand-eye calibration, total uncertainty minimization can be achieved. The direct computation of global covariance is not considered, however.

5.4.2 Verification of the planning criterion

The selection of measurement points in sensing planning i.e. maximizing the distances between the points is shown here to illustrate the effect of given criterias for spatial uncertainties of the localized cube. The result of the sensing planning is verified by distributing the measurement points around the three sides of the workobject and analyzing the spatial uncertainties of a corner point of the localized cube. Throughout this simulation, the criteria of selection of the measurement points maximum distance from each other will be shown. The criterias used for evaluating the uncertainties were product of eigenvalues of error covariance matrix, sum of eigenvalues of error covariance matrix and volume of the uncertainty ellipsoid. In all the three different cases, two pair of points from each surface was measured. The placing of the points to the surfaces was done similarly in each side: all the possible location variations of measurement points was calculated. The parameter that uncertainties were related to was a distance between measurement points on each surface. The criterion for selecting the pair of points is two distances that give the largest total distance. These two largest distances was summed together and total distance is a sum of distances between two point pairs in each surface. Uncertainties to the system are taken from the actual system illustrated in the chapter 5.

Figure 26 illustrates the uncertainties of a point in the calibrated cube frame when the measurement points are distributed around the three sides of the cube. In all the Figures 26–28 the most highest uncertainties are cut off to improve the illustration of the curves.

From the Figure 25 can be seen that when the distance between the points on the surface of each side increases, the uncertainties of pose of the cube decreases. The trend of the curve seems to be that uncertainties are decreased tramatically first and after that, decrease is more stable.

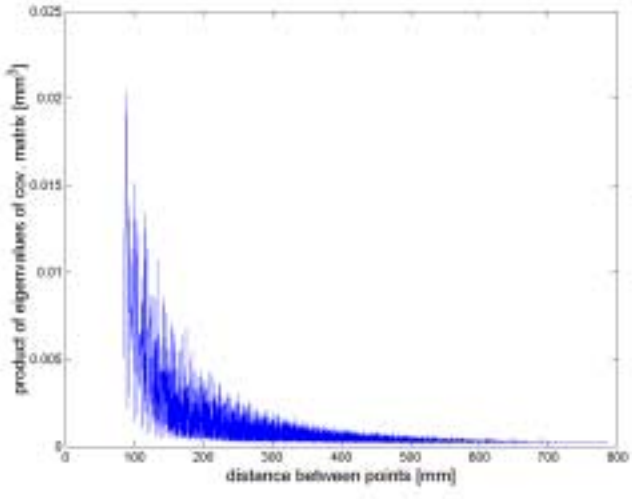


Figure 25. Product of eigenvalues of error covariance matrix vs. distance between the measurement points.

In the Figure 26, sum of the eigenvalues of the error covariance matrix with respect to distance between the points. As in the case of product of eigenvalues, the trend is that uncertainties are decreasing, here not such fast.

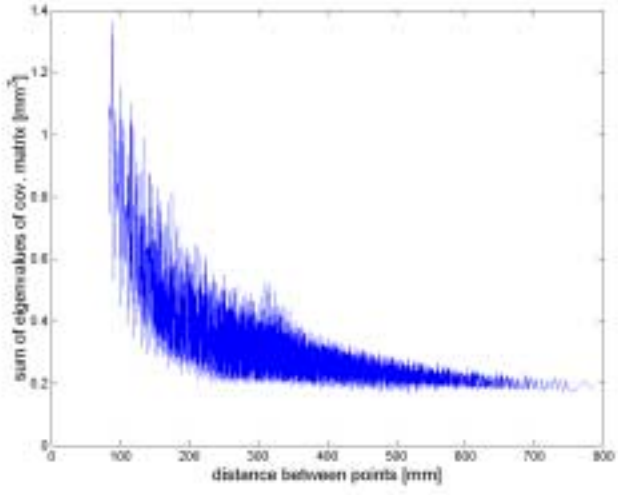


Figure 26. Sum of eigenvalues of error covariance matrix vs. distance between the measurement points.

In the last case, volume of the uncertainty ellipsoid is illustrated for different distances between the points, Figure 27. Curve is close to product of eigenvalues, i.e. the curve is decreasing rather fast.

All the three different cases illustrated the same behaviour between uncertainties of the localized cube and distance between the measurement points. Here one criteria of uncertainties in simulations has been selected to simplify the interpretation of results, i.e. product on sum of eigenvalues, not single eigenvalues. This kind of methods are suitable because usually the goal is to minimize the whole uncertainties of the system, not only one direction.

The principle of using distances between measurement points can be generalized into other forms of set of point. The set of points may form a feature which distances will be considered. Here an example of localizing of a cube using three points forming a tringle in each surface is illustrated. The goal is to decrease uncertainties and the distance of points is calculated as a circle of triangle formed by the three measurement points. The uncertainties are illustrated in a same way as in case of two point pairs, product and sum of eigenvalues and the volume of the uncertainty ellipsoid.

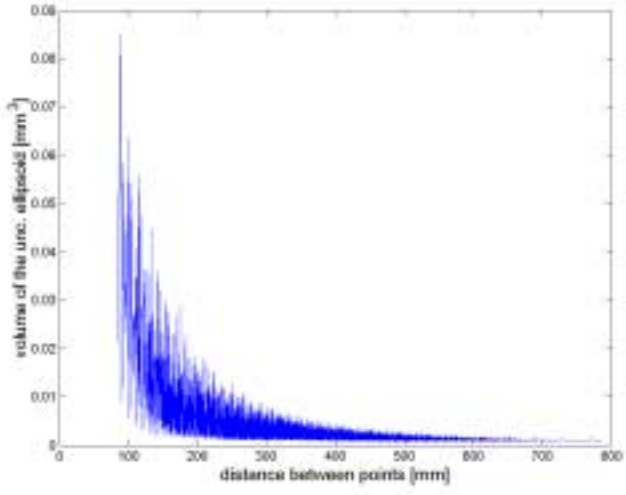


Figure 27. Volume of the uncertainty vs. distance between the measurement points.

The form of the curve is similar with the case of using two pair of points for localizing the cube, Figure 28. The uncertainties are decreasing first quite fast, but after that the uncertainty of the measurement points becomes such high that the pose uncertainties are decreasing rather slowly. The peaks in the curve means that the distance between two points is rather long but the third point is close one of these two points and located unfavourable. The curve in Figure 29 shows that uncertainties may become larger even when distance is increasing if location of one point is much worse than the other two.

Simulation results from sum of eigenvalues and volume of uncertainty ellipsoid, Figures 29 and 30 shows that increase of distance between measurement points decreases spatial uncertainties. Using of maximum distance of measurement points as a criterion for minimizing error covariance matrix can be applied to the different forms of set samples. However, if the number of points increase, the complexity of computing distances becomes higher and one have to be careful when calculating the distances.

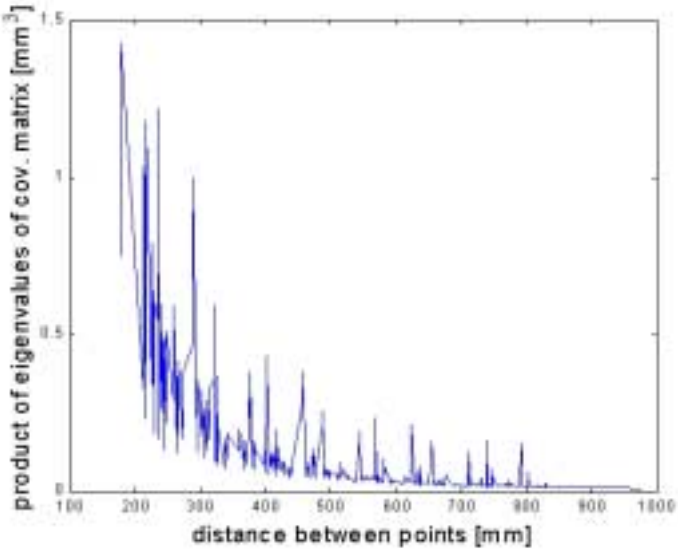


Figure 28. Product of eigenvalues of error covariance matrix vs. distance between the measurement points, case three points in each side.

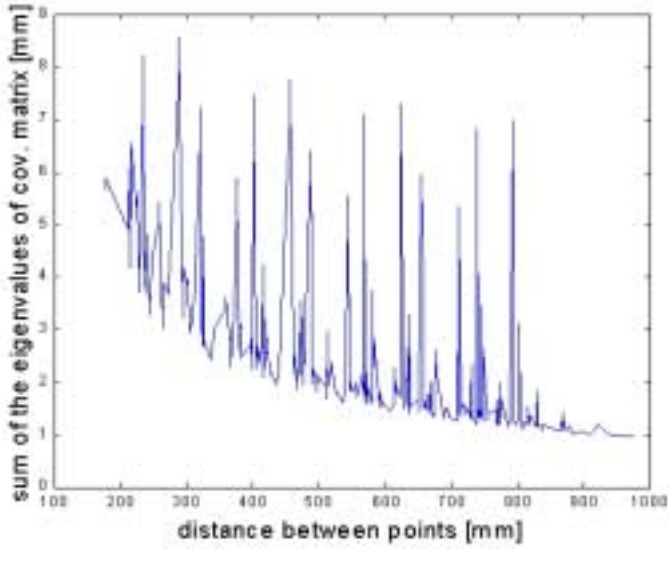


Figure 29. Sum of eigenvalues of error covariance matrix vs. distance between the measurement points, case three points in each side.

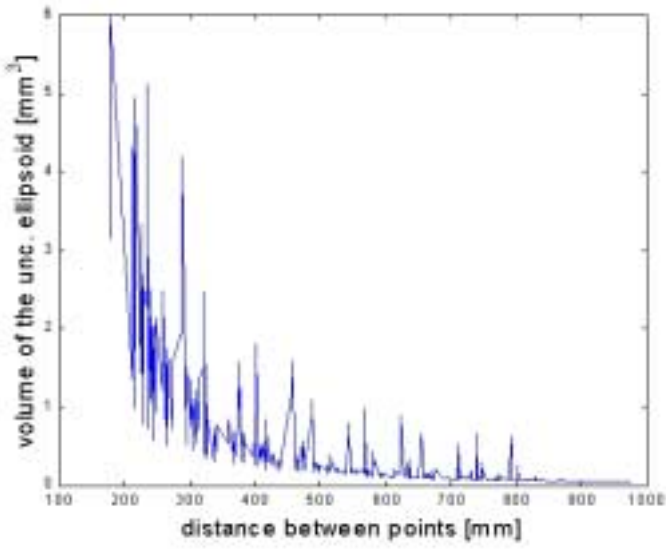


Figure 30. Volume of the uncertainty vs. distance between the measurement points, case three points in each side.

Steps in planning the measurements

The sensing planning procedure can be divided into the following eight steps. Here the measurement target, i.e. hand-eye calibration or work object localization, is referred to as an object.

1. Select the object and parameters to be estimated

2. Determine the boundary values for the system with regard to the measurement parameters

3. Generate the measurement matrix Ψ_J from the Jacobian matrix J

4. Generate the measurement matrix Ψ_K from the weight matrix Q

5. Combine the measurement matrices Ψ_J and Ψ_K into the total measurement matrix Ψ

6. Check that the measuring device is able to measure the selected points

7. Estimate the spatial uncertainties of the estimated parameters

8. If the required level of spatial uncertainties is achieved, stop, otherwise go to step 3 and add further measurements.

The above sensing planning algorithm is connected with a whole parameter estimation model, so that after the measurements have been planned the spatial uncertainties can be analyzed. Depending on the requirements, the achieved level of spatial uncertainties is either satisfied or not. If a more accurate level is required, more measurements should be planned and performed. When running the actual robot system, all the measurements should be run off-line, and also motion planning and verification of the accuracy of the required level. In that way the production system can run efficiently without extra interruptions.

The problem involved in planning the reference features illustrated in this chapter is that planning is based on the assumption that the feature contain very few or no spatial uncertainties. This means that the requirement for successful measurement is that the deviation between the nominal location of the pose of the work object, for example, and its actual location should be smaller than in the selection of details for the reference features. If the deviation is larger, there is a possibility that the robot will measure false features or even ones lying beyond the work object.

The proposed method is illustrated in few examples in this thesis. In the next paragraph, planning the set of measurements is carried out for workobject localization. In the chapter 6, a planning for localization of a cube is presented and localization of the cube in 3D world is demonstrated.

5.4.3 Planning the measurements for work object localization

We will consider here equations for planning the measurements required to localize a cubic work object. Of the six plane surfaces constituting the cube,

three are measured for localization of the pose. The sensing planning starts out from the parameters to be estimated, i.e. the pose of the work object.

The pose of the work object is determined using three translation values and three orientation values:

$$m_{cube} = \begin{bmatrix} x & y & z & \phi_x & \phi_y & \phi_z \end{bmatrix} \tag{121}$$

The measurement points are planned for each of the three plane surfaces, in order to minimize uncertainties in all six pose parameters. Each surface is determined using translation from the origin to the surface and two orientation values of the surface. The surfaces are termed here the x-plane, y-plane and z-plane, referring to normal directions in the world coordinate frame. Error functions for localizing the cube in the world frame can then be written for the sides as follows: the error function for the measurement point on the x-plane surface is $e_x = x$, that for the measurement point on the y-plane surface $e_y = y$, and that for the measurement point on the z-plane surface $e_z = z$. These error functions are used to compose the parameters for sensing planning, equations (121), (124) and (127).

The Jacobian matrix for estimating the pose parameters of the work object for the measurements $1 \ldots n$ for three different surfaces can be written as follows:

$$\frac{\partial e}{\partial m_{cube}} = \begin{bmatrix} 1 & 0 & 0 & 0 & -z_{x,1} & y_{x,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & -z_{x,n} & y_{x,n} \\ 0 & 1 & 0 & -z_{y,1} & 0 & x_{y,1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & -z_{y,n} & 0 & x_{y,n} \\ 0 & 0 & 1 & -y_{z,1} & x_{z,1} & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & -y_{z,n} & x_{z,n} & 0 \end{bmatrix} \tag{122}$$

where

$(x_1 \ldots x_n, y_1 \ldots y_n, z_1 \ldots z_n)$ are the measurements of the parameters (x,y,z) between $1 \ldots n$ for x, y and z surfaces

n is the number of measurements.

The goal of the planning is to find measurements points (x,y,z) for each pair of rows in equation (120) such that they will give an optimum or close to optimum amount of information for parameter estimation.

To find the parameters to be maximized in each pair of rows in Jacobian matrix, the partial derivative of the error function with respect to the parameters to be estimated for the plane surface in the direction x can be written as follows:

$$\frac{\partial e_x}{\partial m_{cube,x}} = \begin{bmatrix} \frac{\partial e}{\partial x} & \frac{\partial e}{\partial \phi_y} & \frac{\partial e}{\partial \phi_z} \end{bmatrix} \tag{123}$$

The whole pose includes 6 pose parameters, but each surface (here the x-plane) gives information only on three parameters, and also information on these parameters for the Jacobian matrix, i.e. one translation and two orientation parameters. The partial derivative of the error function with respect to the estimated pose parameter x is 1. The control parameters are then obtained from partial derivatives of the error function with respect to the orientation parameters, equation (121),when the constraints of the model are $0 \ldots n$ and $n+1 \ldots n+n$, as follows:

$$J_{cube,x} = \begin{bmatrix} 1 & 0 & 0 & 0 & -z_1 & y_1 \\ 1 & 0 & 0 & 0 & -z_2 & y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & -z_{n-1} & y_{n-1} \\ 1 & 0 & 0 & 0 & -z_n & y_n \end{bmatrix} \tag{124}$$

$$\Psi_{i=0}^{i=n} = \max\{(y_1, z_1), (y_2, z_2) \ldots (y_{n-1}, z_{n-1}), (y_n, z_n)\}_{cube,x} \tag{125}$$

The result is a set of points generated by minimizing and maximizing the measurement points on the reference surface of the cube in directions y and z.

This actually means that there are four locations on the x-plane surface of the work object where the points are measured: close to the coordinate origin, where y = 0 and z = 0, at the end of the y-axis, where y = max, z = 0, at the end of the z-axis, where y = 0, z = max, and at the opposite corner from the origin, where y = max and z = max.

The localization of the y-plane partial derivative of the error function with respect to the parameters defining the y-plane can be written:

$$\frac{\partial e_y}{\partial m_{cube,y}} = \begin{bmatrix} \frac{\partial e}{\partial y} & \frac{\partial e}{\partial \phi_x} & \frac{\partial e}{\partial \phi_z} \end{bmatrix} \tag{126}$$

and the measurement parameters to be minimized and maximized are (control parameters):

$$J_{cube,y} = \begin{bmatrix} 0 & 1 & 0 & -z_1 & 0 & x_1 \\ 0 & 1 & 0 & -z_2 & 0 & x_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 0 & -z_{n-1} & 0 & x_{n-1} \\ 0 & 1 & 0 & -z_n & 0 & x_n \end{bmatrix} \tag{127}$$

$$\Psi_{i=0}^{i=n} = \max\{(x_1, z_1), (x_2, z_2) \ldots (x_{n-1}, z_{n-1}), (x_n, z_n)\}_{cube,y} \tag{128}$$

and correspondingly for the plane surface in the direction z :

$$\frac{\partial e_z}{\partial m_{cube,z}} = \begin{bmatrix} \frac{\partial e}{\partial z} & \frac{\partial e}{\partial \phi_x} & \frac{\partial e}{\partial \phi_y} \end{bmatrix} \tag{129}$$

and the measurement parameters to be minimized and maximized (control parameters):

$$J_{cube,z} = \begin{bmatrix} 0 & 0 & 1 & -y_1 & x_1 & 0 \\ 0 & 0 & 1 & -y_2 & x_2 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 1 & -y_{n-1} & x_{n-1} & 0 \\ 0 & 0 & 1 & -y_n & x_n & 0 \end{bmatrix} \tag{130}$$

$$\Psi_{i=0}^{i=n} = \max\{(x_1, y_1), (x_2, y_2) \ldots (x_{n-1}, y_{n-1}), (x_n, y_n)\}_{cube,z} \tag{131}$$

The minimization and maximization equations (123), (126) and (129) all contain the same parameters, so that the result of planning is that measurements will be minimized and maximized in these directions and the uncertainties in the pose of the work object will be minimized.

The weight matrix

The criteria to be considered are the weight matrix of the error covariance matrix and computation of the parameters which affect the uncertainties in the estimated parameters, i.e. matrix Ψ_K. This means that the weight matrix K has to be minimized, which is done by minimizing its parameters in a pair of rows. When locating the work object, as in the case of calculating the matrix Ψ_J, each surface has to be considered separately, because weights for the surfaces are different. The input to the system is a measurement point p:

$$p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix} \tag{132}$$

The weight matrix is composed by calculating the partial derivative of an error function $e_x = x$ with respect to the input parameters. For the plane in the direction x, the weight matrix can be computed as follows:

$$\frac{\partial e_x}{\partial p} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \tag{133}$$

The weight matrices for the plane surfaces in the directions y and z can be written as follows, whereupon the error functions are $e_y = y$ and $e_z = z$:

$$\frac{\partial e_y}{\partial p} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \tag{134}$$

$$\frac{\partial e_z}{\partial p} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \tag{135}$$

As illustrated above, the weight matrices are constant in all directions. It can be written for $1 \ldots n$ measurements for all in one matrix:

$$K_{cube} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 1_n & 0 & 0 \\ 0 & 1_n & 0 \\ 0 & 0 & 1_n \end{bmatrix} \tag{136}$$

The above example means that only the translation parameters of the measurement affect the uncertainties in the estimated parameters, and the measurement poses can be planned based on information from the Jacobian matrix Ψ_J. This is because only the uncertainty attached to the measurement points in the work object coordinate frame was considered.

The output of the sensing planning algorithm is a total measurement matrix Ψ which is a combination of measurement matrix Ψ_J and measurement matrix Ψ_K. The total measurement matrix is a set of points selected based on given criteria, i.e. located at reference features on the surface of the work object, here close to the origin and corners of the cube.

The result of the localization of the work object is illustrated in the Figure 31. In the figure, actual location is solid line and reference location is described with dashed line. The set of samples are selected from four location in each side, drawn in the reference object. As illustrated in the figure if the deviation between reference model and actual location is large, the planning have to carry

out such that reference features are not out of the actual surface. The planning is here carried out for each surface separately.

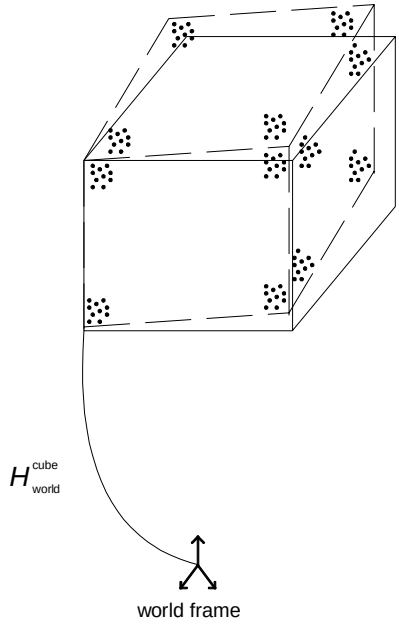


Figure 31. Result of the sensing planning for work object localization.

The total number of features composed using sensing planning is 12. Depending on the requirements of the spatial uncertainties of the location of the work object, the measurement points are constantly distributed around these features.

5.5 Discussion a comprehensive framework for the modelling, estimation and planning of parameters in robot workcells

The geometry of a robot has been modelled here using the Denavit-Hartenberg notation, which is based on joints and links which define the transformation from the base frame of the robot to the TCP of the robot. When modelling the robot, the geometric model is assumed here to include all the models of the

control system, as these are not produced separately. The calibration of the robot system consists of calibrating static parameters, i.e. the geometry of the system.

A method for simultaneous estimation of the hand-eye transformation and the location of a calibration object, here a plane, is also presented in this chapter for a laser -range sensor. The method is flexible and easy to use, due to the freely selected motions and simple calibration object.

Localization of the pose of the work object depends on the surface form of the object. Its surfaces can be divided into parametric and patched surfaces, where the parametric surface forms considered here are planar, cylinderical and spherical and the patched surfaces considered are B-spline surfaces. The localization of parametric surfaces is more straightforward than that of work objects with patched surfaces. On the other hand, patched surfaces provide a much larger variety of surface forms. When using parametric surfaces, the feature selection that follows data acquisition plays an important role, although this task is not considered here. The localization of work objects with parametric surfaces gives a more accurate result, however, due to the form of the surface model. Problems are often encountered in estimating the orientation parameters of objects with patched surfaces, because flat surfaces cause large orientation errors even with a low level of noise. Localization of the work object with a triangular facet surface is similar to the localization method using patched surfaces presented here, i.e. calculations of the shortest distance from a measured point to the closest facet on the surface. The difference between a patched and a facet surface is that the facet model is discrete but the patched model is continuous.

The parameter estimation models presented in this chapter provide tools for estimating the spatial uncertainties of the estimated transformation parameters. These methods are also fairly easy to apply in different contexts. A Gaussian form of noise modelling has been studied in this thesis, the main intention being to investigate whether the assumption of Gaussian-form noise gives a reliable approach to modelling the spatial uncertainties in a robot system. As the model is sensitive to noise, it is important to know whether it will work realiably when the noise level is changing. Answers to these questions have been sought in experimental tests.

Three means are employed here to model noise based on its location in the transformation, and are considered when estimating the spatial uncertainties of a coordinate transformation in a robot workcell. Covariance propagation gives the transformation inclusive of noise that follows the nominal transformation, while Monte Carlo simulation adds the noise to the transformations. Different forms of uncertainty have to be transformed to the same form before they can be combined. Equations for the computation of spatial uncertainties are composed in this thesis for hand-eye calibration, work object localization and surface model parameters. In each case uncertainties are estimated for the parameters to be estimated. To interpret the resulting covariances, it is better to consider uncertainties in a specific feature, e.g. a point. In that way it is possible to avoid several types of units and mixing of results. No method for considering the spatial uncertainties of a surface model when localizing a work object with a patched surface has been reported in the literature. Equations are given for this here and tests are described in the experimental chapter. Modelling of the spatial uncertainties in the surface is one of the crucial points, especially if the surface is rough or irregular, because it has a major effect on the localization uncertainties attached to the work object (Sallinen & Heikkilä 2001b). In the case of surface uncertainty included in the localization uncertainties at a typical noise level, the work object location estimate was 10 times larger than when localizing the work object without surface model uncertainties.

The combining of several measurements with respect to uncertainties was discussed in this chapter, and a Fisher-form sensor fusion method was described which can be used computationally for this purpose. The main motivation for the use of sensor fusion in the estimations of coordinate transformations presented here is to utilize all the measurement information obtained from a sensor. Another important advantage is the improvement in accuracy when updating the estimate afterwards. This means that the accuracy of the estimate can be improved by adding the new measurements to the estimate without calculating all the data again.

The intention of this chapter in the field of sensing planning has been to illustrate a novel method for sensing planning which relies on modelling of the spatial uncertainties using covariances, as presented in this thesis, and trying to minimize the error covariance matrix. When the behaviour of the spatial uncertainties in parameter estimation is known, the error covariance matrix can

be used as a planning criterion and the results can be interpreted either as values or geometrically. The selection of set of points is based on minimizing the error covariance matrix and it leads to a situation where distances between measurement points are maximized. The reliability of the criterial is illustrated with simulations where pose uncertainties compared with distance between the measurement points. There is not such a close connection between uncertainty analysis and sensing planning in the literature as is proposed in this thesis, and as a result the new method of sensing planning is very flexible and easy to implement in conjunction with other planning operations, e.g. hand-eye calibration or estimation of the model parameters for a surface.

6. Experimental tests

This chapter presents the results of simulations and of the experimental tests required for verifying the properties of the proposed covariance propagation methods, which are of a kind that has not been reported before. The intention was to test the proposed methods at different phases and to analyse carefully the spatial uncertainties affecting the estimated parameters.

The algorithms adopted for parameter estimation and covariance propagation were tested with Monte Carlo simulations and actual tests to verify the confidence of the presented methods when used in different phases of a robot system as described in this thesis. The tests included covariance propagation of the joint–based model of the robot, hand-eye calibration, work object localization and sensing planning. In the case of hand-eye calibration and work object localization, large-scale repeated tests were carried out to verify the results. Parameter estimation and uncertainty analysis were performed in conjunction with covariance propagation and the Monte Carlo simulation in the case of the surface models, while the principle of the pose estimation methods and their sensitivity to noise were first illustrated in a simple cube pose estimation problem.

The dimensions and poses of both calibrated objects such as a sensor in the hand, the calibration plane and the localized work object and geometrical illustrations including an uncertainty ellipsoid were not verified using any external measuring tool, e.g. a machine for measuring coordinates. The performing of such an operation would be complicated and expensive. Instead the required reliability was demonstrated by comparing the results of covariance propagation with a Monte Carlo simulation and actual tests.

The experimental robot system was equipped with a hand-eye laser rangefinder based on the conventional triangulation principle (CCD/laser)(MEL 1998). There were no problems with reflection, due to the use of a matte surface on the targets and the high quality of the sensor. The robot used in the experimests was an ABB IRB1400 manipulator arm with six rotational joints and a S4 controller. The tag points for the robot motions (paths) were generated off-line using the commercial off-line programming tool ENVISION (Deneb 2000). The range sensor values and the actual values of the robot joint encoders were stored to

verify the positions actually achieved. Reference data on the surfaces and robot motion data were acquired from the work object CAD model and robot off-line programming tools, respectively. In addition, both hand-eye calibration and localization of the work object were supported by the noise models for the range sensor and the robot joint sensors. The measurement models assume rigid links for the robot, which is a fair assumption when measurements are made at a reasonably slow speed and similar joint configurations are used, so that the deflections are stable in each set of samples.

6.1 Uncertainties in simple examples

6.1.1 Estimating the model parameters

The principle of the estimation method presented in this thesis is illustrated in the example of estimating the pose of a cube in a world coordinate frame. The state vector defining the pose includes all six pose parameters, i.e. three translation parameters and three orientation parameters. The input values for the estimator are a set of points which are distibuted along the axis of the cube and a priori information about the location of each side of the cube in the cube frame and in the world frame. The location of a cube is described as a homogeneous transformation H_{world}^{cube} in the world frame, i.e. each point $p_i = \begin{bmatrix} x & y & z \end{bmatrix}$ can be expressed in the world frame using the following transformation:

$$p_{world} = V_{world}^{cube} p_i + T_{world}^{cube} \tag{137}$$

where

V_{world}^{cube} is the rotation matrix of the H_{world}^{cube},

T_{world}^{cube} is the translation vector of the H_{world}^{cube}.

The pose parameters of the cube can be written in vector form:

$$m_{cube} = \begin{bmatrix} x & y & z & \phi_x & \phi_y & \phi_z \end{bmatrix} \tag{138}$$

This is a state vector which is needed to calculate an estimate for the homogeneous transformation. The set of points is fitted to the model of a cube on each side using an error function as the fitting criteria. The error function is the shortest distance of a measured point from each surface, cf. the example of sensing planning in section 5.4.3:

$$e_x = x \tag{139}$$

$$e_y = y \tag{140}$$

$$e_z = z \tag{141}$$

Because the error function is a non-linear function related to the state parameters, these latter are estimated iteratively by linearizing the error function around the nominal values, $m_i = 0$. Each iteration step produces correction values for the estimated parameters, and the remaining parameters to be estimated are updated using these corrections. For the cube transformation, linearization gives one point measurement for each side of the cube. The result is a partial derivative from the error function with respect to the estimated parameters, i.e. the state vector can be written:

$$\frac{\partial e}{\partial m_{cube}} = \frac{\partial e}{\partial p_{cube}} \frac{\partial p_{cube}}{\partial m_{cube}} \tag{142}$$

The result of a partial derivative of the error function with respect to each pose parameter is as follows:

$$\frac{\partial e}{\partial m_{cube}} = \begin{bmatrix} \frac{\partial e}{\partial x} & \frac{\partial e}{\partial y} & \frac{\partial e}{\partial z} & \frac{\partial e}{\partial \phi_x} & \frac{\partial e}{\partial \phi_y} & \frac{\partial e}{\partial \phi_z} \end{bmatrix} \tag{143}$$

and the partial derivatives are calculated from

$$\frac{\partial e}{\partial m_{cube}} = \begin{bmatrix} 1 & 0 & 0 & 0 & p_z & -p_y \\ 0 & 1 & 0 & -p_z & 0 & p_x \\ 0 & 0 & 1 & p_y & -p_x & 0 \end{bmatrix} \tag{144}$$

where

p_x, p_y, p_z are points in the cube frame.

This equation is used as a Jacobian matrix for parameter estimation and is solved iteratively, with equation (141) giving one line with six columns for the Jacobian matrix when multiplied by $\partial e / \partial p_{cube}$, which is a 1x3 vector. The estimated parameters are corrected iteratively to find the transformation which minimizes the distances between the measured points and the surfaces of the object. Using the Bayesian-form estimation method, the increments for updating the estimated parameter can be calculated using equation (17), as given in section 5.2.1.

The noises of the measured points are collected into one noise matrix, marked as R_i, and are set as diagonal terms. The noise is assumed to be gaussian distributed. One 3D point in the noise matrix forms a 3x3 noise matrix in which the diagonals are variances of the actual noises in the directions x, y and z:

$$R_i = \begin{bmatrix} \sigma_x^{\,2} & 0 & 0 \\ 0 & \sigma_y^{\,2} & 0 \\ 0 & 0 & \sigma_z^{\,2} \end{bmatrix} \tag{145}$$

The whole noise matrix is formed by setting the 3x3 noise matrices of a point noise into the total noise matrix R in the form of diagonal blocks:

$$R = \begin{bmatrix} R_1 & & \cdots & & 0 \\ & R_2 & & & \\ \vdots & & \ddots & & \vdots \\ & & & R_{n-1} & \\ \bar{0} & & \cdots & & R_n \end{bmatrix} \tag{146}$$

The number of blocks is equal to the number of measurements in the system.

The other part of the error covariance matrix is the weight matrix K. Sources for this simple example come from the measured points:

$$p_i = \begin{bmatrix} x & y & z \end{bmatrix}^T \tag{147}$$

The error function depends non-linearly on the measurements and it is linearized with respect to the parameters to be measured. The error is assumed to be very small, and for the partial derivative we can write:

$$K = \frac{\partial e_i}{\partial p_i} \tag{148}$$

The partial derivative for localizing the cube then becomes

$$\frac{\partial e_i}{\partial p_i} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{149}$$

The matrix in equation (146) is 3x3, which means that it weights the measurements from three directions. Thus the weight matrix for computing small corrections to the estimated parameters is calculated in section 3.3 in the form of equation (72), by multiplying the weight matrix by the noise matrix on both sides.

6.1.2 Uncertainties in the estimation of the pose of a cube

The spatial uncertainties of a point in a localized cube are studied in this section. The idea is to study the behaviour of the estimation as the set of samples changes from poor to versatile. The observations are made by comparing the covariance propagation method and Monte Carlo simulation.

The three sets of samples are illustrated in the figure below, which shows three sets of points along all the coordinate axes: 1) points covering the length of the cube, range 0…100 units, 2) points spread along the coordinate axes, representing part of the side of the cube, range 0…30 units, and 3) points close to the origin along the axis, range 0…10 units. To analyze the uncertainties in the pose of the cube, the uncertainties of a point $p = [100 \quad 100 \quad 100]$ mm in its coordinate frame were calculated. The feature point represents the corner of the cube and is on the opposite side from the coordinate origin. When estimating the point with translation parameters alone, the mixing of scales in the error covariance matrix must be avoided.

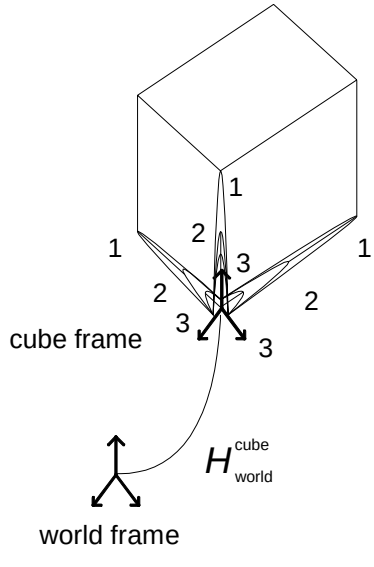


Figure 32. Cube with three sets of points.

Case 1: Point range 100 units, number of points 90.

The first case will consider points spread along the total length of the sides of the cube, in the range 0…100. The results of the covariance propagation are presented in Table 1 and the Monte Carlo simulation in Table 2. The uncertainty ellipsoids for noise level 0.1 are illustrated in Figure 33.

The uncertainty ellipsoids are rotated, i.e. they are eigenvectors, in both covariance propagation and Monte Carlo simulation, as can be seen in the results in Tables 1 and 2 and Figure 31. The form of the uncertainty ellipsoid is wide but flat in both propagation and Monte Carlo simulation (i.e. the dimensions of the two largest eigenvalues are high compared with that of the third). The dimensions of the two largest eigenvalues in covariance propagation and Monte Carlo simulation are also close to each other, in spite of the fact that there is about a 45 degree orientation difference between the propagation and Monte Carlo simulation. In these cases most of the rotation is around eigenvector 3, which has the smallest eigenvalue, and the resulting ellipsoids are quite close to each other. The uncertainties are scaled linearly as the noise at the points changes. The covariance propagation gives a nominal solution for the uncertainty ellipsoid, i.e. the eigenvalues are of the same size, whereas the Monte Carlo simulation is closer to a practical approach (actual system), in which the eigenvalues are rarely the same, due to the effect of noise.

Table 1. Eigenvalues and respective eigenvectors for the error covariance matrix, case 1, covariance propagation.

Noise at the points [unit]	λ_1 [unit]	λ_2 [unit]	λ_3 [unit]	Eigenvectors
0.1	0.062	0.062	0.018	0.730 -0.367 0.577 -0.682 -0.448 0.577 -0.047 0.815 0.577
1	0.618	0.618	0.183	0.773 -0.367 0.577 -0.682 -0.448 0.577 -0.047 0.815 0.577
10	6.184	6.184	1.826	0.730 -0.367 0.577 -0.682 -0.448 0.577 -0.047 0.815 0.577

Table 2. Eigenvalues and respective eigenvectors for the error covariance matrix, case 1, Monte Carlo simulation.

Noise at the points [unit]	λ_1 [unit]	λ_2 [unit]	λ_3 [unit]	Eigenvectors
0.1	0.047	0.059	0.0138	0.852 -0.374 -0.368 0.030 0.735 -0.677 -0.523 -0.566 -0.638
1	0.488	0.599	0.139	0.807 0.461 -0.370 0.102 -0.725 -0.681 -0.582 0.512 -0.632
10	4.792	6.410	1.371	0.881 -0.286 -0.377 -0.059 0.724 -0.687 -0.470 -0.627 -0.622

Case 2: Point range 30 units, number of points 90

In the second case, the points are spread over the range 0…30 units along the axis of the cube. The results of the covariance propagation are presented in Table 3 and those of the Monte Carlo simulation in Table 4. The uncertainty ellipsoids are again rotated, as can be seen from the eigenvectors, especially when the level of noise is increasing. The eigenvalues for the two methods are not so close to each other as in case 1, the uncertainties still rise linearly with a linear increase in noise level.

Table 3. Eigenvalues and respective eigenvectors for the error covariance matrix, case 2, covariance propagation.

Noise at the points [unit]	λ_1 [unit]	λ_2 [unit]	λ_3 [unit]	Eigenvectors
0.1	0.248	0.248	0.018	0.730 -0.367 0.577 -0.682 -0.448 0.577 -0.047 0.815 0.577
1	2.477	2.477	0.183	0.773 -0.367 0.577 -0.682 -0.448 0.577 -0.047 0.815 0.577
10	24.773	24.773	1.826	0.730 -0.367 0.577 -0.682 -0.448 0.577 -0.047 0.815 0.577

Table 4. Eigenvalues and respective eigenvectors for the error covariance matrix, case 2, Monte Carlo simulation.

Noise at the points [unit]	λ_1 [unit]	λ_2 [unit]	λ_3 [unit]	Eigenvectors
0.1	0.198	0.232	0.015	0.743 0.468 -0.478 0.039 -0.744 -0.667 -0.668 0.477 -0.571
1	1.891	2.268	0.147	0.823 -0.3050 -0.480 -0.113 0.740 -0.663 -0.558 -0.599 -0.574
10	24.310	29.770	122.156	0.973 0.220 -0.077 0.220 -0.763 0.607 -0.075 0.608 0.791

The uncertainty ellipsoids are rotated in both propagation and Monte Carlo simulation, as can be seen from the eigenvectors and in Figure 33.

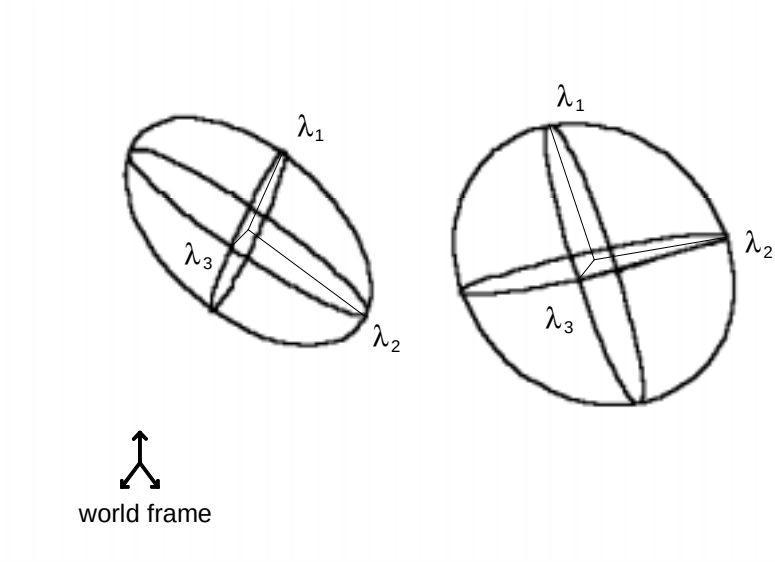


Figure 33. Uncertainty ellipsoids (eigenvectors and respective eigenvalues). Covariance propagation (right) and Monte Carlo simulation (left).

Case 3: Point range 10 units, number of points 90

In the third case the points are close to the origin of the coordinate frame, range 0…10 units. The resulting uncertainties for propagation are presented in Table 5 and those for Monte Carlo simulation in Table 6.

The uncertainty ellipsoids are again rotated in this third case, as can be seen from the eigenvectors in Table 6. The eigenvalues of the Monte Carlo simulation are slightly smaller than when using covariance propagation, but the deviation between covariance propagation and Monte Carlo simulation is fairly large. The difference is quite large at noise levels of 0.1 and 1, and very large at a noise level of 10.

The results suggest that the difference between covariance propagation and Monte Carlo simulations seems to become smaller as the set of sample points becomes more versatile and larger when the set of samples is poor. In addition, the level of uncertainty improves significantly when the set of samples becomes more versatile. The failure of covariance propagation can be seen first when the noise level increases to 10 times that at noise level 1 in case 3. This means that the calculation converges to local minima. The difference between covariance propagation and Monte Carlo is greatest in case 3, while as the versatility of the set of samples reduces the uncertainties, the difference between covariance propagation and Monte Carlo simulation is lowest in case 1.

In all the cases with noise levels of 0.1 and 1 the uncertainty ellipsoids have two large dimensions and one much smaller one, but their orientations differ between covariance propagation and Monte Carlo simulation. In any case, the largest uncertainties point in the same direction, and the results can be considered reliable for estimating uncertainties in the robot system, for instance. When considering the eigenvectors and respective eigenvalues, it is clear that new information about the uncertainties in the system has been obtained.

Table 5. Eigenvalues and respective eigenvectors for the error covariance matrix, case 3, covariance propagation.

Noise at the points [unit]	λ_1 [unit]	λ_2 [unit]	λ_3 [unit]	Eigenvectors
0.1	0.784	0.018	0.784	0.730 0.577 -0.367 -0.682 0.577 -0.448 -0.047 0.577 0.815
1	7.844	0.183	7.844	0.773 0.577 -0.367 -0.682 0.577 -0.448 -0.047 0.577 0.815
10	78.436	1.826	78.436	0.730 0.577 -0.367 -0.682 0.577 -0.448 -0.047 0.577 0.815

Table 6. Eigenvalues and respective eigenvectors for the error covariance matrix, case 3, Monte Carlo simulation.

Noise at the points [unit]	λ_1 [unit]	λ_2 [unit]	λ_3 [unit]	Eigenvectors
0.1	0.599	0.015	0.739	0.821 -0.495 -0.284 -0.145 -0.662 0.735 -0.552 -0.562 -0.616
1	5.927	0.309	7.382	0.840 -0.496 -0.219 -0.204 -0.664 0.719 -0.502 -0.559 -0.660
10	101.508	99.614	98.122	-0.840 0.438 -0.319 -0.134 -0.738 -0.661 0.526 0.513 -0.679

6.2 Uncertainties in robot geometry

The geometry of the robot is modelled using the Denavit-Hartenberg model (Paul 1983) and the joint-based model is transformed into pose form, including translation and orientation values of the TCP of the robot (6 DOF). The transformation also includes the spatial uncertainties of the particular pose as presented in section 5.2.3 and assuming the noise is gaussian distributed. The differences between covariance propagation, experimental tests and Monte Carlo

simulation were tested, and the results can be seen in Table 7 below. The results are eigenvalues for the error covariance matrix.

Table 7. Validity of the TCP error propagation model.

	λ_x [mm]	λ_y [mm]	λ_z [mm]	$\lambda_{\phi x}$ [deg]	$\lambda_{\phi y}$ [deg]	$\lambda_{\phi z}$ [deg]
Covariance propagation	0.0180	0.0102	0.0086	0.0021	0.0027	0.0010
Experimental test	0.0180	0.0098	0.0079	0.0031	0.0013	0.0009
Monte Carlo Simulation	0.0181	0.0104	0.0086	0.0021	0.0027	0.0010

The eigenvalues in Table 7 are quite close to each other, because the level of noise is fairly low and pure propagation does not need estimation, so that linearization gives the largest effects for the difference between the three methods. The eigenvalues are calculated from an error covariance matrix with 6 DOF. Covariance propagation is close to Monte Carlo simulation, and the eigenvalues for the experimental tests are somewhat smaller even if 6 DOF covariance matrix is analyzed.

6.3 Uncertainties in hand-eye calibration

In the first estimation phase, hand-eye calibration was carried out by measuring the points from a steel calibration plate located in front of the robot. All the measurement motions were generated off-line by randomly selecting positions and orientations around the calibration plate. When estimating the spatial uncertainties, the reference point $p = \lfloor 0 \quad 0 \quad 200 \rfloor$ mm was estimated in the coordinate frame of the tool. The set of points consisted of 28 randomly chosen points on the surface of the calibration plane. The number of actual tests carried out was 800, i.e. each 28-point sequence was run 800 times and covariance matrices calculated from 800 tool-to-TCP transformations. Different tests are collected to Table 8.

Table 8. Specifications of different methods in hand-eye calibration.

	Number of points	Number of repeats
Covariance Propagation	28	1
Experimental test	28	800
Monte Carlo Simulation	28	2000

The noise matrices for hand-eye calibration was calculated as follows: the joint noises were measured for each pose separately and propagated through the links as a TCP pose error covariance matrix, cf. section 5.2.3 These matrices were calculated dependent on pose. The noise in the sensor range measurement was also measured for each pose separately.

The results of hand-eye calibration obtained by three methods are given in Table 9. The eigenvalues are close together and only small differences exist. The form of the uncertainty ellipsoid is wide but flat in all cases, and there are eigenvalues for two markedly larger dimensions perpendicular to each other and one eigenvector for an eigenvalue perpendicular to these two. The flatness applies in the measuring direction, which indicates that the uncertainties in orientation parameters have a greater effect on the measurement point than do the translation parameters, and that the orientation parameters are more difficult to estimate accurately. Estimation of orientation parameters is also a problem in several cases reported in the literature (Gunnarsson & Prinz 1987, Dornaika & Horaud 1998, Heikkilä et al. 1993).

Table 9. Point uncertainty in the calibrated sensor frame.

Method	λ_1 [mm]	λ_2 [mm]	λ_3 [mm]	Eigenvectors
Covariance Propagation	1.286	1.505	0.598	0.389 0.855 0.344 0.914 -0.405 -0.028 -0.116 -0.325 0.939
Experimental test	1.183	1.659	0.504	0.211 -0.836 0.507 0.931 0.330 0.156 -0.298 0.439 0.848
Monte Carlo Simulation	0.961	0.991	0.319	0.261 0.895 0.363 0.957 -0.288 0.022 -0.124 -0.342 0.931

To test the effect of noise on system behaviour, the propagated covariances were calculated and Monte Carlo simulations were run for cases in which the input noises were 0.001, 0.01, 0.1, 1, 10, 100 and 1000 times the original values for the robot system. The behaviour of the algorithm was studied in two ways using this test: 1) how much the resulting uncertainty is reduced as the level of input noise decreases, and 2) how the correspondence between the covariance propagation and Monte Carlo simulation changes as the noise level changes. The difference is due to system linearization, i.e. the reliability of the covariance propagation weakens as the noise level increases, and the final result can be an unreliable estimate.

Figures 34 and 35 illustrate the (x,y,z) point uncertainty (eigenvalues in the direction of the eigenvectors of the error covariance matrix). The covariance propagation algorithm is depicted in Figure 34 and the Monte Carlo simulation in Figure 35. In both cases the uncertainty level of the measured point changes in a linear manner depending on the input noise. According to the figures, when the uncertainty level decreases by a factor of 10 the uncertainty of a point in the calibrated sensor frame decreases simultaneously by a factor of 10. The same behaviour happened when the noise level was scaled to 10 times larger than the nominal value, and the same trend continued when scaling the noise level to 0.01, 0.001, 100 and 1000 times. The form of the uncertainty ellipsoid can be seen from Figures 34 and 35. The parameters x and y have larger values and z is slightly smaller, i.e. the form is wide but flat. The difference between these parameters also remains the same when the noise level changes.

The second issue was the behaviour of the algorithm (i.e. how the propagation estimate follows the simulation result) when the noise level changes. To illustrate this behaviour, the difference between the covariance propagation and Monte Carlo simulation was calculated for each level of noise. The differences, calculated in percentages, are depicted in Figure 36.

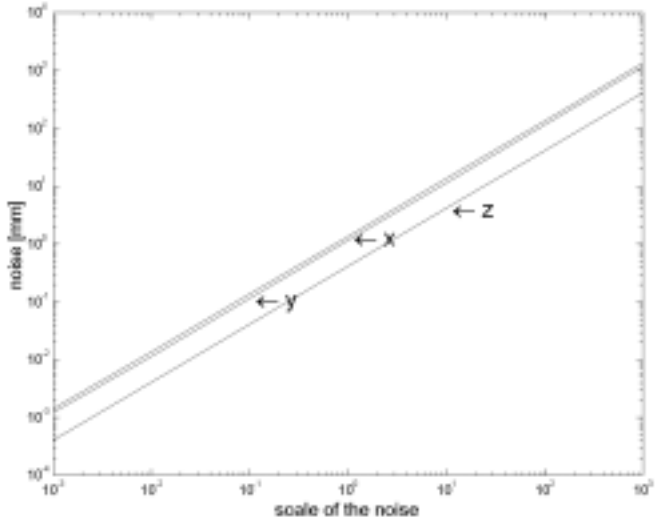


Figure 34. Point uncertainty (x,y,z) in the estimated sensor frame, calculated by the covariance propagation method.

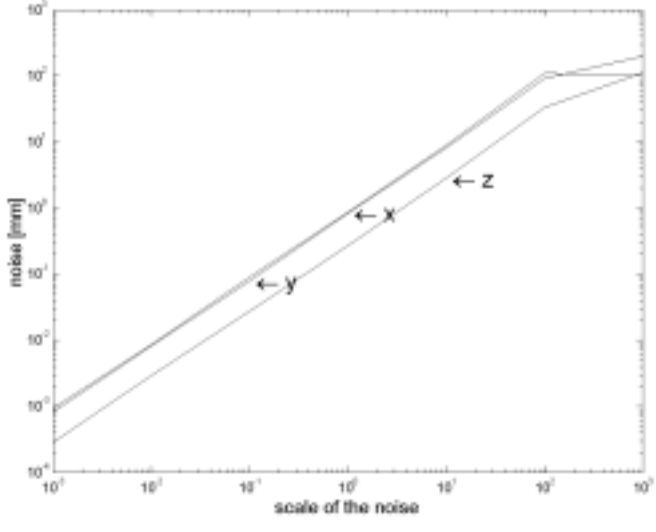


Figure 35. Point uncertainty (x,y,z) in the estimated sensor frame, calculated using Monte Carlo simulation.

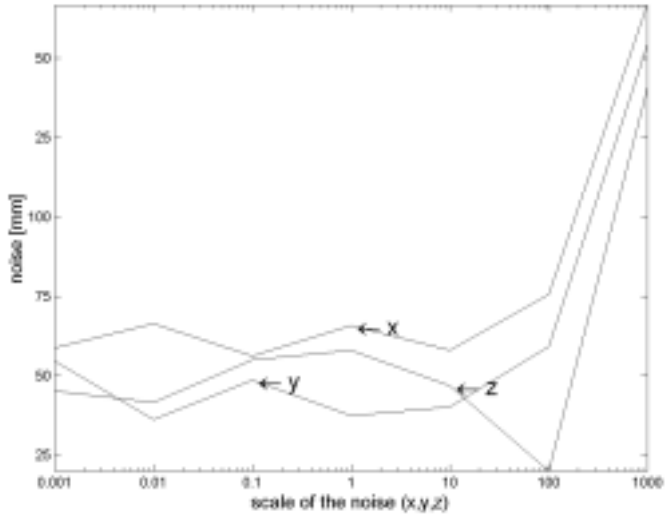


Figure 36. Difference between the propagation algorithm and Monte Carlo simulation with respect to the measurement point in the calibrated sensor frame.

According to Figure 36, the difference between the propagation algorithm and the Monte Carlo simulation remains similar even if the noise level changes. The difference rises when the noise level increases 100 times compared with the original value, but below this noise level the difference between the propagation and Monte Carlo simulation is about 35%. As shown in Table 8, the results in terms of the uncertainty level for the actual system lie between the propagation and simulation results.

6.4 Uncertainties in work object localization

A motivation for careful analysis of localization of the work object is affect of propagated uncertainty of hand-eye calibration. An other point for study is that while hand-eye calibration is carried out relatively rarely in flexible manufacturing cell, in the single series production, work object has to be localized for every piece. Therefore uncertainties and their analysis plays an important role in work object localization.

Covariance propagation for work object localization was tested with actual measurements. A machined cube was localized and measured on three sides to estimate its full pose with six degrees of freedom. The maximum uncertainties in the form of eigenvalues in the direction of the eigenvectors for the error covariance matrix of the work object pose is presented in Table 11.

Table 10. Specifications of different methods in work object localization.

	Number of points	Number of repeats
Covariance Propagation	26	1
Experimental test	26	300
Monte Carlo Simulation	26	1000

The uncertainties in the work object pose were studied by considering the a posteriori error covariance matrix of a point in the corner of the cube. This corner was located on the opposite side of the cube from the origin and the dimensions of the point were $p = [100 \quad 150 \quad 100]$ mm. The actual work object localization tests were carried out 300 times and the Monte Carlo simulation 1000 times. The result regarding the pose of the work object was $m_{cube} = [853.537 \quad -396.717 \quad 667.983 \quad 0.958 \quad -0.581 \quad -0.0378]$ [mm/deg] in the robot frame.

Table 11. Point uncertainty in the calibrated work object frame.

Method	λ_1 [mm]	λ_2 [mm]	λ_3 [mm]	Eigenvector
Covariance Propagation	1.916	0.350	1.253	0.828 0.401 -0.394 -0.089 0.786 0.612 -0.554 0.472 -0.686
Exprimental test	0.890	0.257	1.231	-0.621 0.496 0.607 0.670 0.738 0.083 -0.407 0.458 -0.790
Monte Carlo Simulation	6.372	1.488	4.166	0.949 0.296 -0.109 -0.305 0.770 -0.561 -0.082 0.565 0.821

There are two larger eigenvalues λ_1 and λ_3 in Table 11 which seem to be much higher than the third λ_2 (about three times larger). This same form of the uncertainty ellipsoid can also be seen in the case of hand-eye calibration, see Table 9. Uncertainties in orientation values have the greatest effect on the total uncertainties in the pose of the work object, the implication for the planning phase being that the measurements should be planned so as to minimize the uncertainties in orientation parameters.

The results suggest that the relative level of noise in the system seems to be rather high. Another important fact is the versatility of the set of samples, which in these tests was a randomly chosen set of points. If the points were selected inorder to minimize the level of uncertainties, the correspondence between the propagation algorithm and the Monte Carlo simulation could be improved.

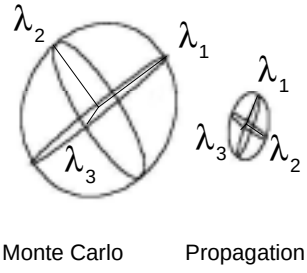


Figure 37. Uncertainty ellipsoids and eigenvectors for the Monte Carlo simulation (left) and covariance propagation methods (right).

The uncertainty ellipsoids in the case of covariance propagation and the Monte Carlo simulation are illustrated in Figure 37, where λ_1, λ_2 and λ_3 are the eigenvalues in the direction of the eigenvectors, the latter being represented in the figure by lines. The two ellipsoids are similar in form but that for the Monte Carlo simulation is much larger in size. In both cases the eigenvalues for the two largest eigenvectors are closer to each other and the third is smaller. Geometrically, this means that the ellipsoids are wide but flat. In the case of work object localization it means that the orientation uncertainties of the coordinate frame of the work object affect the uncertainties most. It is good to keep this in mind when planning the measurements, in order to know what task

has to be to carried out in the robot system, i.e. the number of measurements has to be increased in the critical directions.

As in hand-eye calibration, the behaviour of parameter estimation was similarly studied here as the noise level changed. The input noise was scaled between 0.001, 0.01, 0.1, 1, 10 and 100 times the actual level of noise and the feature considered was a point in the corner of the work object.

The difference between the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$ also remains small in both covariance propagation, Figure 38, and Monte Carlo simulation, Figure 39, behaviour that indicates that the uncertainty ellipsoid has not been rotated. If the orientation had changed, the eigenvalues would have changed and the result would have been a saw-like function for each eigenvalue (Sallinen et al. 2000). The uncertainties arising from the hand-eye calibration still affect the parameter estimation in work object localization, however, as can be seen when comparing the eigenvalues for hand-eye calibration, Figure 39, with those for work object localization, Figure 40, the latter curves being less linear.

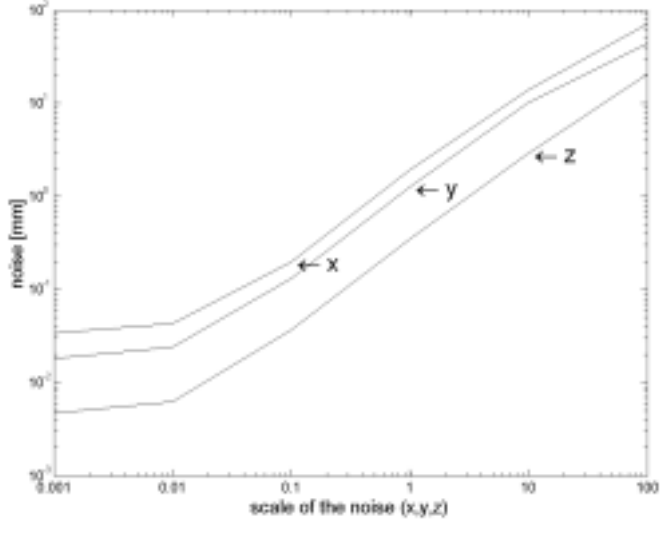


Figure 38. Uncertainties at the given point (directions x,y,z), covariance method.

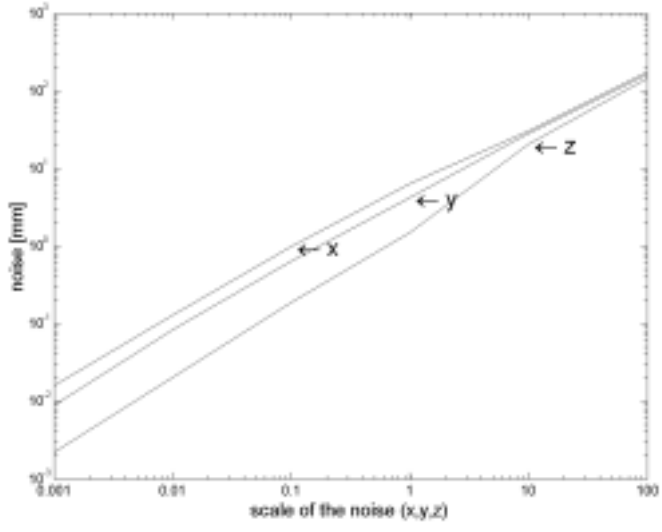


Figure 39. Uncertainties at the given point (directions x,y,z), Monte Carlo simulation.

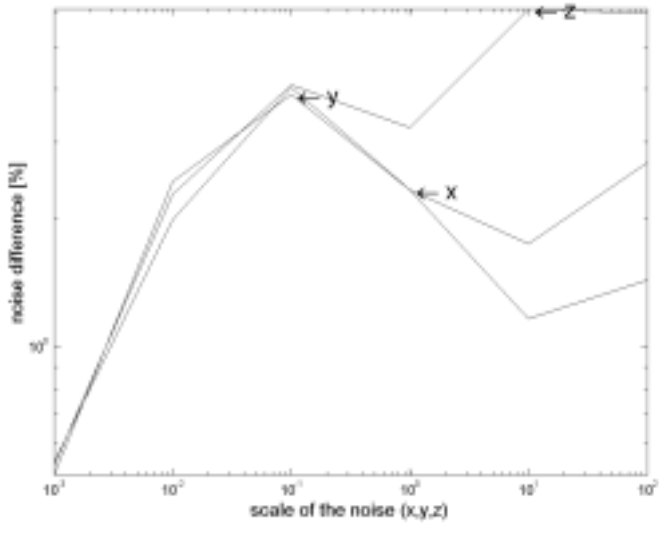


Figure 40. Difference between covariance propagation and Monte Carlo simulation (directions x,y,z).

The fact that the curves representing the eigenvalues for the point on the work object are not linear can be attributed to the system linearization and the fairly high level of noise. The difference between the covariance propagation method and Monte Carlo simulation remains at the same level, however, indicating that the system is working properly. The difference between covariance propagation and Monte Carlo simulation can be seen in Figure 40, where at is seen to contrast with that found in then case of hand-eye calibration. The total level of noise is propagated to the higher level and the difference is not linear, the difference on each axis is about the same. The interpretation for this behaviour lies in the fact that the orientation of the uncertainty ellipsoid remains the same but its size and volume is change.

The following three implications can be extracted from the test results illustrated above: First, when examining the eigenvectors of different methods in Tables 8 and 9, it can be seen that the orientation difference between the methods is somewhat exiguous. The eigenvectors maintain approximately the same orientation even when the noise level changes, because the curves are linear in Figures 35 and 39. In other words, the orientation of eigenvectors in the error covariance matrix (and uncertainty ellipsoid) does not rotate when the noise level changes. Second, the results of the actual tests are close to the propagation results, indicating that the distribution of noise in the actual system is close to a Gaussian model. This can also be seen from the measurements of the sensor and joint noise. The covariance propagation method assumes that all noise is Gaussian-distributed. Third, uncertainty propagation seems to be reliable when combining the uncertanties coming from hand-eye calibration with those affecting work object localization. The noise level increases in that case, but propagation seems to work robustly, because the difference between covariance propagation and Monte Carlo estimation is so small.

From the system designer’s point of view, limits can be set for the system requirements. In the actual system, the mean value for noise in the joints of the robot was 0.0015 degree when considering all the poses and respective joints. The other noise source, noise from the sensor, was 0.0305 mm. According to the simulations, the noise level can be raised to ten times this figure and the propagation algorithm still gives relatively reliable estimates. This means that the robot can have an average noise of up to 0.015 degree in each joint and the measuring noise in a sensor may be up to 0.3 mm. This is not so straightforward,

however. The noise from different joints has different effects on the compound noise, the noise in the first three joints having a much larger effect on the TCP noise than that in the next three, and noise in the joints is dependent on the poses, as illustrated by Sallinen & Heikkilä (2001a). Moreover, the versatility of the set of samples is even more important than the noise level of the system, as demonstrated in the sensing planning tests in section 5.6.

It can be stated that the reliability of the estimation of parameters and respective uncertainties depends on the quality of the set of samples in two ways: via the versatility of the set of points and via the noise level of the set of points. Thus substantial attention has to be paid to these two aspects when designing hand-eye calibration and work object localization. The covariance propagation method seems to be useful for estimating spatial uncertainties in the robot system even though it is approximate and not very accurate due to non-linearities in the model and the actual system.

It should be emphasized that simulation and actual tests are the best ways to verify the sensitivity of the covariance propagation method. When considering the different noise levels, it is evident that evaluation methods such as condition number (Motta & McMaster 1997) or noise amplification index (Nahvi & Hollenbach 1996) do not give such a reliable result and are focused on evaluating other things and not spatial uncertainties in the estimated parameters.

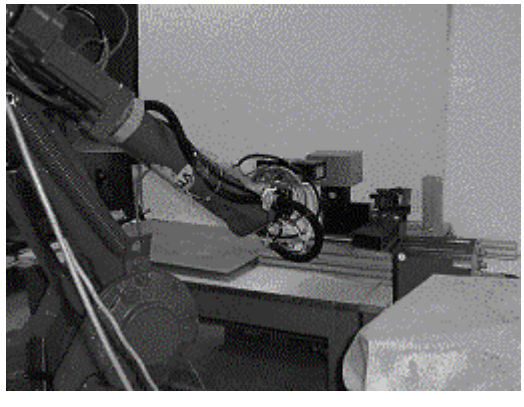


Figure 41. The actual robot system measuring the cube.

The robot is shown in Figure 41 measuring a cubic work object located in its working area, i.e. the noise values of the joints could be expected to be close to the mean value for the Gaussian noise (i.e. possible noise bias was avoided or minimized).

6.5 Uncertainties in work object surface models

Estimation of surface model parameters and spatial uncertainties was tested using the covariance propagation method for different surface forms including parametric and patched surfaces. Spatial uncertainties were calculated for different numbers of points to observe the behaviour of the estimation. For patched surface forms, both covariance propagation and Monte Carlo simulation methods were used and the differences compared. All the sets of points were randomly generated.

6.5.1 Parametric surfaces

Model parameters for parametric surfaces were estimated on the Reverse-Engineering principle, which means that noise was imported into to the system from points measured using the actual robot system. Each measurement point had uncertainties of [0.01 0.013 0.015] mm in the directions x,y,z respectively, and the parameters of each surface were estimated using different numbers of points between 3 and 15. The results, presented in Figures 42–46, are first std. deviations of the diagonal terms of the error covariance matrix. The tests were run in simulations, not with actual workobjects with different surface forms.

Estimation of parameters for a planar surface

Model parameters for a plane surface were estimated using a different number of points, in order to test the effect of the estimated parameters on the spatial uncertainties. The plane surface was defined using one translation parameter and two orientation parameters, which means that the minimum number of points required for estimation is three. The tests started from three measurement points and lasted up to 14 points. The results are illustrated in Figures 42 and 43. The

uncertainties were analyzed by calculating the diagonal values in the error covariance matrix. The values for the estimated parameters are given in Table 10 using 10 points in the estimation.

Table 12. Estimated parameters of the plane surface.

z [deg]	ϕ_x [deg]	ϕ_y [deg]
100.7403	5.0000	5.0000

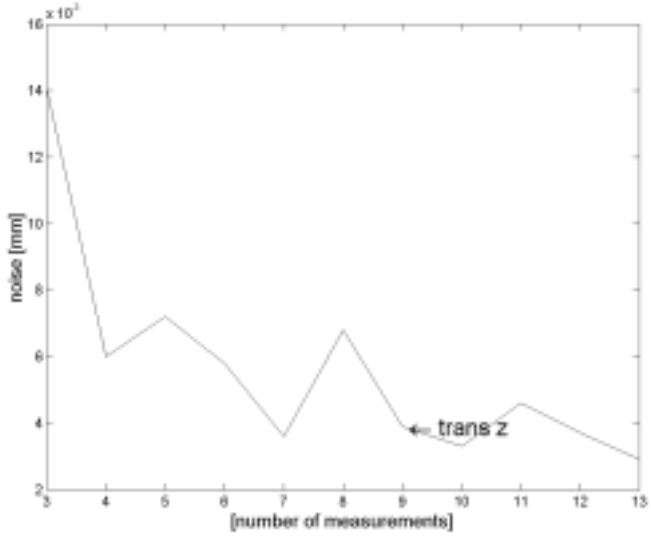


Figure 42. Uncertainty of the plane translation value z with respect to the number of measured points.

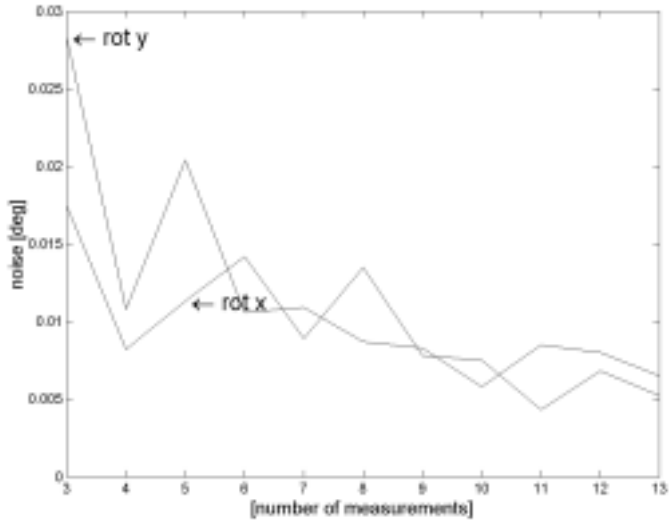


Figure 43. Uncertainty of the orientation values Ox and Oy with respect to the number of measured points.

A typical number of points used for estimating the plane model parameters is 10, and the level of uncertainty achieved is 0.004 mm in translation values and 0.01 deg in orientation values in the plane frame. According to the curves, a fourth point would reduce the uncertainties considerably, while subsequent points would not have such a large effect on the uncertainties. A satisfactory level of uncertainties was achieved with the robot system used in this thesis using 10 points.

Estimation of parameters for a cylinderical surface

The surface model for the cylinder was defined using five parameters, including two translation and two orientation parameters and the radius of the cylinder, and the uncertainties were analyzed using 6 to 15 points. The results in terms of diagonal values in the error covariance matrix are illustrated in Figures 45 and 46. The estimated parameters are given in Table 11, where 10 points were used for parameter estimation.

Table 13. Estimated cylinder parameters.

x [mm]	y [mm]	ϕ_x [deg]	ϕ_y [deg]	d [mm]
40.0374	-600.0373	5.0000	5.0000	75.0000

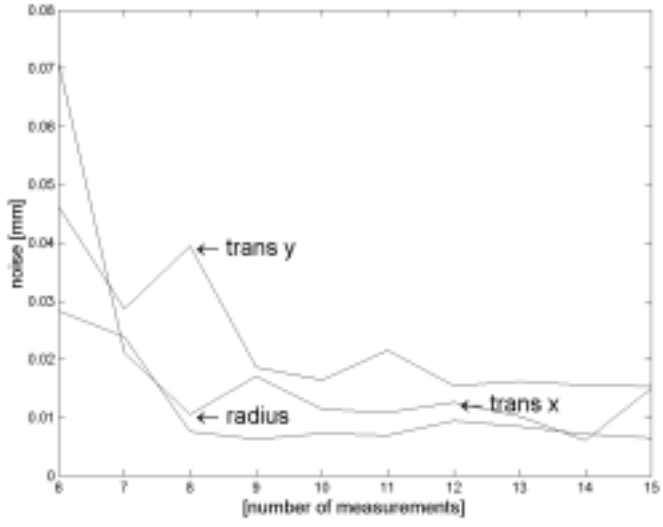


Figure 44. Uncertainty in cylinder translation values x,y and the radius of the cylinder with respect to the number of measured points.

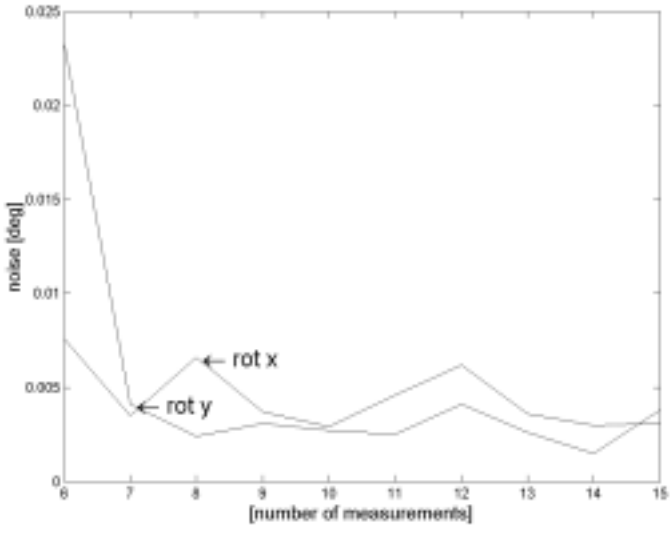


Figure 45. Uncertainty in cylinder orientation values Ox and Oy with respect to the number of measured points.

The same effect as in estimation of plane model parameters can be seen here for cylinder model parameters, in that the uncertainties decreased radically after the first points and more steadily after that. 10 points gave a fairly good level of uncertainties in the robot system used here, 0.02 mm in translation values and 0.005 deg in orientation values when estimating the cylinder pose parameters, even though the number of parameters to be estimated was more than in the case of a plane surface.

Estimation of parameters for a spherical surface

Model parameters for the surface of a sphere were estimated using 4…12 points measured on the surface. The following results were obtained for the estimated parameters, the centre point and the radius of the sphere, Table 14:

Table 14. Estimated sphere pose parameters.

x [mm]	y [mm]	z [mm]	d [mm]
20.000	200.0000	50.0000	75.0000

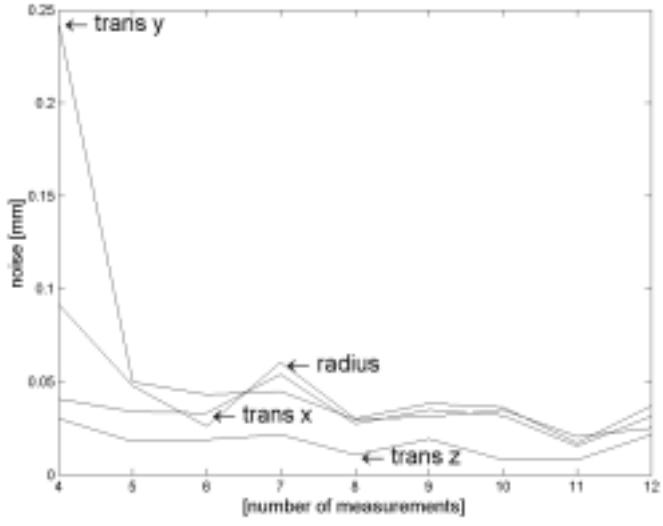


Figure 46. Uncertainty of the sphere translation values x, y, z and radius with respect to the number of measured points.

Again the same effect can be seen in the uncertainties in the spherical model parameters, the first point after the minimum number of points having the largest effect on the spatial uncertainties of the estimated parameters. According to Figure 46, a satisfactory level of uncertainties can be achieved using 6 measurement points (0.05 mm in translation parameters).

In all the surface models the uncertainty values change between each coordinate axis. This is due to rotation of the uncertainty ellipsoid, in which the form of the ellipsoid remains almost the same but the axes change in their positions. If the surfaces are modelled and estimated using measurements which contain random

variation i.e. noise, the tests suggest that 10 points should be used for parameter estimation. The points are randomly selected around the surfaces in the tests, however, and if the number of points is to be reduced, their quality should be considered. The estimation of surfaces and uncertainties in model parameters by means of uncertainties in calibration and localization would complete the model of spatial uncertainties in a robot system.

6.5.2 Patched surfaces

The effect of spatial uncertainties on patched surfaces was tested in a slightly different way from the case of parametric surfaces because the generation of patched surfaces by the method presented here does not need any estimation. The effect of uncertainties was tested by localizing the object with a B-spline surface patch and analysing the uncertainties at a point in the localized work object pose.

The localized work object was a ship's propeller, and the point, $p = [100 \quad 100 \quad 100]$ mm, was located on its surface. Eigenvalues and eigenvectors were calculated for the point, and the results were as presented in Table 15. To verify the covariance propagation method, Monte Carlo simulation was run 2000 times and the analysis was carried out for the same point. The location of the propeller in the robot coordinate frame was estimated to be $m_{propeller} = [97.459 \quad 926.948 \quad 472.691 \quad 1.158 \quad -0.291 \quad -0.191]$ [mm/deg].

Table 15. Point uncertainty in the estimated work object frame.

Data Set	λ_1 [mm]	λ_2 [mm]	λ_3 [mm]	Eigenvectors
Covariance Propagation	1.466	0.757	0.326	-0.171 0.912 0.372 0.761 -0.118 0.636 -0.625 -0.392 0.675
Monte Carlo Simulation	2.140	1.160	0.326	0.018 0.567 0.824 -0.499 -0.709 0.498 0.867 -0.420 0.270

The eigenvalues in Table 15 are close to each other, involving only small deviations, and as pointed out in previous chapters, the uncertainty ellipsoids are similar in form and orientation. The two main axes of the uncertainty ellipsoid point approximately in the same directions in both covariance propagation and Monte Carlo simulation. The two largest eigenvalues in the case of Monte Carlo simulation seem to be larger than in covariance propagation, but the smallest is the same. Covariance propagation is slightly optimistic, and the same trend can also be seen in other cases of work object localization.

An increase in the number of points used in estimation will reduce the spatial uncertainties attached to the work object pose. To illustrate this, a large number of points (2304) were used to estimate the pose of the work object. Using the sensor fusion method decribed in section 5.3.6 and including the spatial uncertainties of the surface patch, the spatial uncertainties of a point in the estimated work object frame were calculated. The results are given in Table 16.

Table 16. Point uncertainty in the estimated work object frame, determined by the covariance propagation method with a large number of points.

Data Set	λ_1 [mm]	λ_2 [mm]	λ_3 [mm]	Eigenvectors
Sensor Fusion	0.076	0.056	0.015	0.317 0.926 -0.204 -0.675 0.372 0.638 0.666 -0.064 0.743
No Fusion	1.466	0.757	0.326	-0.171 0.912 0.372 0.761 -0.118 0.636 -0.625 -0.392 0.675

When increasing the number of points (up to 100 times relative to the case with no sensor fusion), the uncertainties decreased by a factor of 30…100. This can be used as a rough evaluation of how the uncertainties behave when the number of measurements is increased. It is obvious, however, that there are much more sophisticated methods for reducing the uncertainties than increasing the number of points, e.g. sensing planning.

Measurement of the actual ship propeller is illustrated in Figure 47. There were no problems with reflection, even if a laser range sensor was used, but the surface of the propeller is fairly flat, which means that localization is quite

difficult and it is recommended that measurements should be obtained from positions with high gradients.

As in hand-eye calibration and work object localization, the effect of noise on the parameter estimation was analysed for the localization of a work object with a B-spline surface. The level of system uncertainties was scaled to 0.1, 0.01, 0,001 or 10 times the original situation, including uncertainties attached to the robot and to hand-eye calibration of the sensor. The main interest in the tests lay in studying how much the patched surface affects the work object localization uncertainties. Again the uncertainties at a point on the surface of the propeller were studied and Monte Carlo simulations were run 500 times to verify the results. The results are illustrated in Figures 48 and 49.

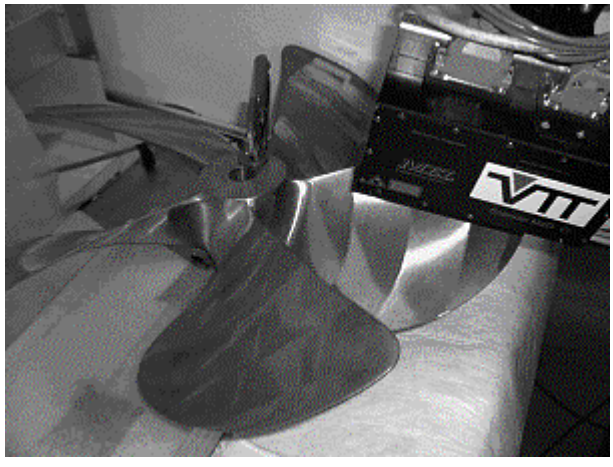


Figure 47. A robot measuring the actual propeller.

The curves in the Figures indicate that localization of a work object with a patched surface does have an effect on the spatial uncertainties, and the eigenvalues are altered and the eigenvectors show much greater rotation than when localizing work objects with parametric surfaces. When the noise level is increased to 10 times that of the reference system, the uncertainties increase markedly. Localization using noises 100 times larger than in the nominal system was tried, but it failed. There are no large differences between covariance propagation and Monte Carlo simulation, but the Monte Carlo curves are slightly smoother.

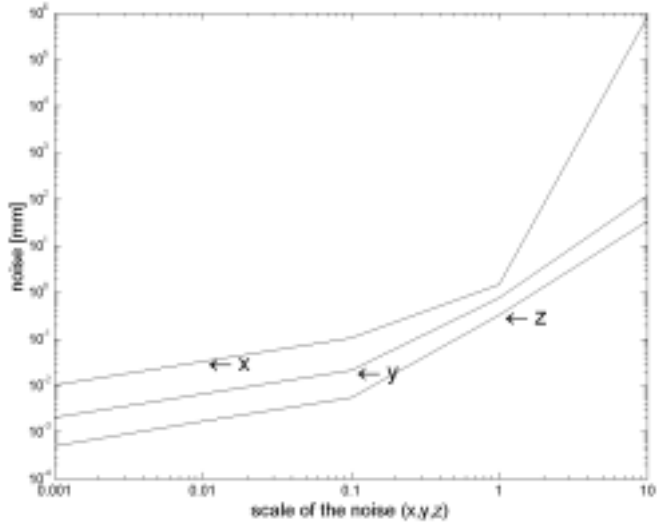


Figure 48. Change in noise level for a point in the work obejct frame, propagation method.

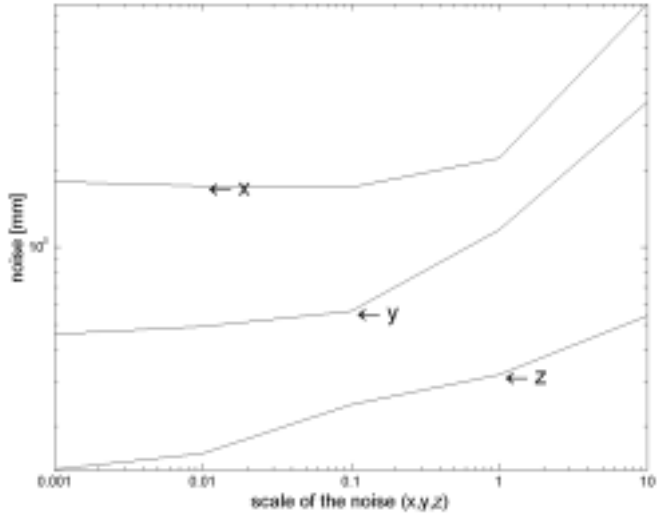


Figure 49. Change in noise level for a point in the work object frame, Monte Carlo simulation.

One of the reasons for the differences between the localization results for a parametric surface and a patched surface lies in the identification of a reference point on the surface. While a point can be found quite straightforwardly on a parametric surface and it is mathematically accurate, it must be located iteratively in the case of a patched surface, which means that its location error will depend on the accuracy of the iteration. In addition, the accuracy clearly increases with the uncertainty of the surface points. It is important in such cases to select points that give information that reduces the uncertainties.

In the test the set of points was selected randomly in order to be distributed as widely as possible around the ship's propeller. The localization measurements was then carried out on three of the propeller's five blades so as to improve the estimation.

6.6 Sensing planning

The sensing planning algorithm was tested with simulations and actual tests. To illustrate the principle of the planning method, a simple example of planning the optimal measurement points for locating a square object is given first, after which the results of locating of a cube in the actual robot frame are presented.

Several methods were used to evaluate the goodness of a set of samples, including calculation of the error covariance matrix by the covariance propagation method, condition number and noise amplification index. The error covariance matrix is studied by calculating the eigenvalues in the directions of the eigenvectors and the volume of the uncertainty ellipsoid, as in the previous chapters. To obtain reliable results, the error covariance matrix is propagated in such a way that it includes parameters with the same units, usually translation information on a point in the estimated coordinate frame.

6.6.1 Planning measurement of the localization of a square

The sensing planning method is illustrated here with a simple example of the estimation of the pose of a square. The steps required to execute the algorithm are as follows: first, computation of the parameters to be considered, second,

minimization and maximization of the points in the direction of the computed parameters (control points), third, generation of the measurements, and fourth, running of the points. After generating the points, the set of samples is compared with a different set and evaluated on several criteria. Only two sides of the square are considered in this example.

The estimated pose parameters of a square in 2D are $m_{square} = \begin{bmatrix} x & y & \phi_z \end{bmatrix}$, i.e. its position on the axis (x,y) and its orientation around the axis z of the origin. The error function is a shortest distance from a measured point to the surface of the square, $e_x = \begin{bmatrix} 0 & 1 \end{bmatrix} p$ for the surface in x direction and $e_y = \begin{bmatrix} 1 & 0 \end{bmatrix} p$ for the surface in y direction, where p is the measurement point. The dimensions for the measurements in both directions to be planned, x and y, will then be, i.e. Ψ_J:

$$\frac{\partial e}{\partial m_{square}} = \begin{bmatrix} \frac{\partial e_x}{\partial x} & \frac{\partial e_x}{\partial y} & \frac{\partial e_x}{\partial \phi_z} & \frac{\partial e_y}{\partial x} & \frac{\partial e_y}{\partial y} & \frac{\partial e_y}{\partial \phi_z} \end{bmatrix} \tag{150}$$

which is equal to

$$\frac{\partial e}{\partial m_{square}} = \begin{bmatrix} 0 & 1 & -x & 1 & 0 & y \end{bmatrix} \tag{151}$$

It can be seen from equation (148) that the dimensions x and y of points on the surface of the work object have the largest effect on the spatial uncertainties in the pose of the square (columns 3 and 6). Let the lengths of the sides of the square be 100 units and number of points on each side 100. For localization of the square, the minimum and maximum values of the points in the directions x and y on both sides are selected, as follows.

$$\Psi_{i=n}^{i=0} = \min, \max\{x\}, \quad \Psi_{i=n+n}^{i=n+1} = \min, \max\{y\} \tag{152}$$

where

n is the number of points on each side of the square.

The result of the planning is that points for localizing the square are selected at four locations in it. On the side x the points are selected from close to the origin (X', Y') and along the x axis in the distant corner While on the side y they are selected from close to the origin and along the y axis in the distant corner, see Figure 51.

The effect of the weight matrix on the set of samples is the same as in the example presented in section 5.4.1, where only the translation values of the set of points affected the uncertainties and the effect of the weight matrix do not have to be taken into consideration when generating the set of samples.

Four set of samples are illustrated in Figure 50: planned, poor, pattern and random. The point considered is located on the opposite side of the square from the coordinate origin.

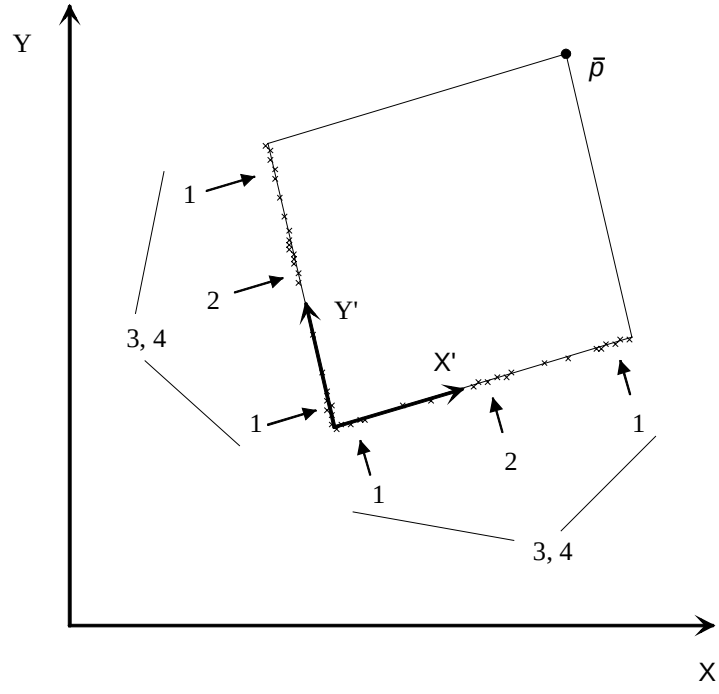


Figure 50. A square and four sets of samples: 1) planned measurements, 2) a poor set of samples, 3) a pattern, and 4) a random set of samples around the sides of the square.

The uncertainties in the square are evaluated by analyzing the spatial uncertainties of the point p on the opposite side of the square from the coordinate origin, to obtain the largest effect of noise on the transformation from the world frame to the square frame. Uncertainties in the position and orientation of the square are propagated to this point. The error covariance matrix for the point can be calculated using Equation (107).

The eigenvalues in the direction of the eigenvectors of the error covariance matrix and the volumes of the uncertainty ellipsoids for the point p in the square frame are presented in Table 17.

If the assumption regarding the selection of the best set of points is correct, a geometrically opposite set of samples will be selected to represent a poor set. These points are distributed in the middle of both sides of the square. The third set of samples is selected constantly around the sides of the square to form a pattern. The pattern set of samples means that measurement positions and orientations are selected regularly around the measurement space. Sample sets of this kind are usually used in calibrating camera systems, for example (Tsai & Lenz 1989). The fourth set of samples consists of randomly selected points around both sides of the square.

Table 17. Eigenvalues, eigenvectors and volumes of the uncertainty ellipsoid for the various set of samples.

Set of samples	λ_1	λ_2	Eigenvector	Volume of the uncertainty ellipsoid [unit^3]
Planned	0.460	0.316	0.707 0.707 -0.707 0.707	0.457
Opposite to planned	5.528	0.316	0.707 0.707 -0.707 0.707	5.492
Pattern	0.680	0.316	-0.707 -0.707 0.707 -0.707	0.675
Random	0.3162	0.817	-0.835 0.551 -0.551 -0.835	0.812

The results in Table 17 indicate that the planning method presented here will give the best set of samples and lowest level of uncertainties for the estimated

parameters. The second best choice is the patterned set and the third best the randomly distributed points. The set of points opposite the planned ones is clearly the worst case. When selecting the set of samples, a pattern is a fairly safe choice, but when the number of points is limited, there will be some problems in selecting suitable ones. A random approach is quite risky, of course, as it may give a good result or a very bad one.

The uncertainty ellipsoids for the planned, patterned and random cases in Table 17 are presented in Figure 51, which shows that there are difference in the lengths of the axes and the orientation of the ellipsoid. The same behaviour will also be observed in the 3D case.

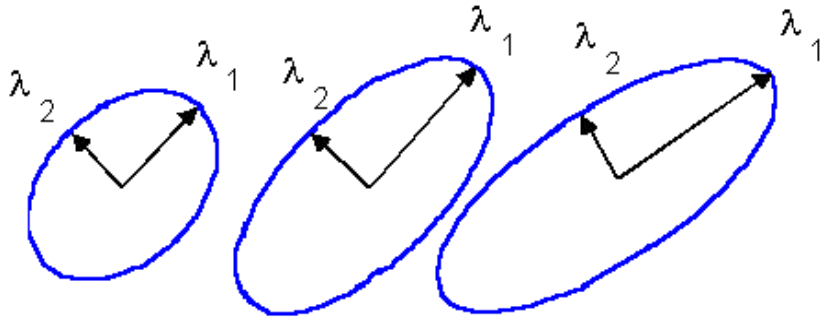


Figure 51. Uncertainty ellipsoids. Left: planned, middle: pattern and right: random.

When considering the orientation of the eigenvectors, all the sets of samples except for the random one give very familiar results. This is because almost the same information is provided for each parameter, whereas in the random case all the parameters have different amounts of information.

6.6.2 Calculation of the pose of a cube

The sensing planning algorithm was tested with actual tests carried out in a different environment from the previous ones and with different robot and sensor systems. The robot system used was a PUMA 650 industrial robot with 6 DOF, and all the measurement poses were generated off-line. The sensor system was a

CCD stereo camera system, developed by Tomita et al. (1998) which enabled 3D points to be measured from a single view, and the point information was used in the estimation. The internal parameters of the camera system were calibrated before the tests and assumed to be constant in the measuring space.

The sensing planning algorithm was evaluated by comparing it with two sets of measurement points, leaving out the one that was opposite to that planned. This meant that the first set used in the test was composed using the planning method, the second was a pattern where the points were consistently distributed around the work object and the third was the random, as used in almost all of the sets of samples in this thesis. The noise level of the system was measured and its values were used in the algorithms. Contrary to hand-eye calibration and work object localization, noise is assumed here to be constant in each pose. The number of points was 8 on each side of the cube, and the uncertainties were analyzed for a point $p = \lfloor 100 \quad 400 \quad 100 \rfloor$ mm in the estimated work object frame. The results are eigenvalues and eigenvectors of the error covariance matrix. The results are presented in Table 18.

Table 18. Results of estimation of a cube, point uncertainty at one corner.

Set of samples	Eigenvalue λ_1 [mm]	Eigenvalue λ_2 [mm]	Eigenvalue λ_3 [mm]	Eigenvectors
Planned	0.0381	0.0439	0.0339	0.9651 0.1782 0.1919 0.1577 -0.9805 0.1173 -0.2090 0.0830 0.9744
Pattern	0.0493	0.0408	0.0453	-0.9369 0.3494 0.0133 0.2193 0.6168 -0.7560 0.2723 0.7054 0.6545
Random	0.0510	0.0402	0.0722	-0.7292 0.6107 -0.3086 0.6607 0.7457 -0.0856 -0.1778 0.2663 0.9473

The results in Table 18 illustrate the effect of the sensing planning algorithm in a 3D case. The eigenvalues are smallest and the eigenvectors point almost in the directions of the work object frame. In the case of the patterned distribution they are rotated slightly, and in the random set of samples the uncertainty ellipsoid has been rotated a lot. The uncertainty ellipsoid is also different in form in the planned, random and patterned cases, being flat in the random case, almost a

sphere in the planned case, and between these two in the pattern case. One reason for this is that the planned set of samples gave an equal amount of information on each parameter to be estimated and an equal noise level. As seen in Table 18, the selection of the points has an effect on the form of the uncertainty ellipsoid, and thus the user has an opportunity to affect the form of the uncertainty ellipsoid at some level. If the points are selected using the sensing planning method presented here and the dimensions of the work object are equal, the uncertainty ellipsoid will take the form of a sphere and the number of points and uncertainties will scale the a posteriori uncertainties of the estimated parameters. This is a very important property when running a robot system where the requirements regarding accuracy and uncertainties are high. If the uncertainties can somehow be controlled, then the measurement sequence can be planned such that the requirements will be fullfilled.

The results of sensing operations was also analysed using various evaluation criteria, including condition number, noise amplification index (Nahvi & Hollenbach 1996) and the volume of the uncertainty ellipsoid. The condition number and noise amplification index were calculated from the identification Jacobian matrix and the volume of the uncertainty ellipsoid from the error covariance matrix of the point at the corner of the work object. The results are illustrated in Table 19.

Table 19. Condition number, noise amplification index and volume of the uncertainty ellipsoid for different set of samples

Set of samples	Condition number	Noise amp. index	Vol. of uncert. ellipsoid [mm^3]
Planned	565	$0.470*10^6$	$2.3760*10^{-4}$
Pattern	813	$0.594*10^6$	$3.8159*10^{-4}$
Random	1721	$1.424*10^6$	$6.1957*10^{-4}$

As indicated in the results in Table 19, the planned set of samples gives the best results with all the evaluation methods. The pattern set is close to the planned one, but the random set clearly lags behind. The results in Table 19 show that the carefully planned set of samples gives roughly three times better results than a random set.

6.7 Other experimental tests

The methods presented in this thesis for estimating of model parameters and covariance propagation have also been applied to a number of other systems, together with the tests presented above. A short overview of the systems is given here.

The whole robotized surface inspection system was built using an ABB IRB1400 industrial robot with a Cyber Optics laser rangefinder. The sensor gives a point laser measurement and the resolution is 0.001 mm (Cyber Optics 1989). The tasks carried out have been hand-eye calibration, work object localization (the work object was a graphite mould) and inspection of the surface of the mould. The research included estimation of spatial uncertainties in the different phases of the system (Heikkilä et al. 1999).

The work object localization was implemented in an optical sensor attached to the TCP of the robot (Sallinen et al. 2000). The sensor was a triangulation angular scan LED array based on a range imaging sensor. A method for calibrating two robots in a one workcell was presented in the same study, followed by localization of the work object using these two robots. The results showed that additional measurements reduce the pose uncertainties of even if the noise level is quite high.

A Bayesian–form method was used for estimating the spatial uncertainties of camera location in the robot workcell (Heikkilä et al. 2000), the principle of the calibration being to move a robot arm with a simple calibration object attached to it into several positions and estimate the transformation from the robot base frame to the constant camera frame by measuring the calibration object. The spatial uncertainties of the transformation were studied by considering the eigenvalues of the a posteriori error covariance matrix.

A work object localization system based on a touch sensor was developed for an industrial application. The sand mould was localized before machining by means of six measurements made on its front surface. The measuring system was built in a demonstration robot cell at the Metso Lokomo Steels foundry, Tampere, Finland. The measuring system includes an easy-to-use graphical interface to facilitate use.

6.8 Discussion of the experimental tests

The parameter estimation methods are explained in this chapter through experimental tests. One of main topics was illustration of the methods for modelling spatial uncertainties, first by means of simple examples and then with the idea expanded to full 6 DOF pose estimation. Another topic was the behaviour of parameter estimation when the noise level is changing, which was assessed by carefully considering the a posteriori error covariance matrices. Finally the sensing planning method presented in this thesis was tested and evaluated.

The different phases of calibration of the robot system are presented, including hand-eye calibration, work object localization and estimation of the surface model parameters. The model parameters and their respective uncertainties were estimated using covariance propagation, Monte Carlo simulation and actual tests. Special attention was paid to the reliability of the estimator by carrying out extensive repeat tests for hand-eye calibration and work object localization in which joint distribution information on different poses was collected and used to model the uncertainties in the robot geometry. A further purpose of this was also to find out whether the noise in the joints is biased or whether a Gaussian assumption is close enough to represent the actual noise distribution.

The tests indicated that higher reliability can be achieved when the system is modelled in more detail than when using a general pose noise model. This effect is illustrated by the difference between the covariance propagation, Monte Carlo and actual tests when models were improved with regard to noise. The test began by modelling the transformation from the base of the robot to the TCP of the robot using constant noises in the pose parameters of the TCP (Heikkilä et al. 1999). Next, the transformation from the base of the robot to its TCP was modelled based on constant noise in each joint (Sallinen et al. 2000), and in the final solution the model was constructed so that each pose involved separate noise. The results, presented by Sallinen & Heikkilä (2001a) and Ramsli (1991), indicate that the noise varies greatly between poses and that this augments the motivation to build a detailed model to represent this noise.

A careful analysis of a posteriori error covariances after each parameter estimation has been presented in this chapter. Computation of a posteriori error

covariances has been suggested in the literature, e.g. by Sanderson (1999), but not such a careful analysis of the results of the estimation. In addition, the analysis of the behaviour of the spatial uncertainties when the noise level is changing and the comparison with Monte Carlo simulation are further contributions made by this chapter.

The proposed sensing planning method was tested in this chapter. First, a simple example of localizing a square in a coordinate frame using different set of samples including planned, opposite to planned, pattern and random was illustrated. Lenghts and directions of eigenvectors as well as volume of the uncertainty ellipsoid was considered as criterias for evaluating the results. The actual tests was carried out by localizing a cube in the world frame and three different set of samples was used for localizing. In both square and cube localization cases presented planning method gave the results with smallest uncertainties and uncertainty ellipsoids were least rotated.

Some additional experiments on the methods presented here are reported briefly at the end of the chapter. The straightforward way of implementing a new application is one of the main advantages of the method.

Selection of the criteria depends on the particular application and the performance requirements. The well-known criterion of the condition number is effective in cases where the uncertainties have to be equal in all directions, but that is not usually the ultimate goal. In some applications there are directions that are critical for uncertainty and directions not so critical, and here the uncertainty ellipsoid provides more infromation than the condition number or observability index. The disadvantage of the uncertainty ellipsoid is interpretation of its rotation and eigenvalues, which is more complicated with the simple numerical values of the condition number, for instance.

7. Discussion

The goal of this thesis is to present methods for estimating model parameters with respect to spatial uncertainties and for evaluating the spatial uncertainties in a robot system. The transformations in a robot system consist of spatial relations between the coordinate frames and of their respective uncertainties. The motivation for the work is to have a method for combining the uncertainties arising from different sources and for estimating the total uncertainty in the system. The model is augmented with criteria for evaluating the uncertainties and using them for planning sensing actions. The methods presented in this thesis are evaluated with actual tests.

Methods for estimating model parameters can be divided into deterministic and stochastic methods as well as linear and non-linear types. One traditional, rather straightforward method for solving the estimation problem is to use linear models. These are often very successful, but they are not the best solution in the presence of noise. Instead, non-linear methods provide a more powerful solution for parameter estimation, and it is these that are used in the present thesis. Deterministic approach do not consider noise in the system whereas stochastic system models the noise in given distribution.

Basically the same parameter estimation methods can be used for hand-eye calibration, work object localization and estimating the parameters of surface models. The major differences between the calibrations lie in the numbers of coordinate transformations and parameters estimated. In hand-eye calibration, both the TCP to sensor origin and robot to calibration object transformations have to be estimated, while in other forms of calibration one transformation is usually estimated. The number of pose parameters estimated varies from three in the case of a plane surface to six in the case of localization of a work object with three surfaces. In addition, variations in work object localization can be found depending on the surface forms of the object to be localized. These may be divided into parametric and patched forms. Modelling the different surface forms has been reported in the literature (Charlebois et al. 1997, Li et al. 1998, Sanderson 1999) but they suffer from the lack of modelling the spatial uncertainties of the estimated parameters, especially in the case of patched surfaces.

The stochastic parameter estimation method presented here is based on iterative Newton-Rhapson estimation. The method has been found to be applicable in the presence of noise and the iterative solution usually converges to a global minimum after four to eight steps. There is a possibility, however, that the iterative search will converge to a local minimum, especially in cases where a set of measurement data does not contain enough information for estimating the parameters or the level of noise is relatively high. This may be the case when localizing a ship's propeller using measurements of the surface of a single blade. Another example is 6DOF hand-eye calibration of a sensor with insufficient rotations around the z-axis of the sensor frame. Situations of this kind can be managed by carefully planning the set of samples and avoiding a high level of noise relative to the number of measurement points.

The modelling of spatial uncertainties can be divided into three classes: stochastic, deterministic and simulation methods. The method proposed in this thesis belongs to the stochastic class and is Bayesian in form. The advantage of the method is that it is straightforward to implement in many applications where the noise is assumed to have a Gaussian distribution. Another stochastic method which uses assumption of Gaussian distribution of noise is Kalman filter (Maybeck 1979), which is actually close to the method presented here. Probability functions provide a method for modelling uncertainties using probability bars or a grid for describing them. Tolerance propagation (Taylor & Rajan 1988) and grid-based modelling are examples of deterministic methods for modelling spatial uncertainties (Burgard et al. 1998). The most common simulation-based method is Monte Carlo localization (Moreno et al. 2002), where the uncertainties are presented as probability density samples. The methods presented for modelling spatial uncertainties can be applied to all phases of estimating coordinate transformation in a robot cell.

Each method for modelling spatial uncertainties has its own advantages. A Bayesian-form method was selected here and methods for evaluating the uncertainties were developed. Basically, the uncertainties are evaluated on the basis of an a posteriori error covariance matrix. These uncertainties are illustrated geometrically using an uncertainty ellipsoid, which can be studied by considering the orientation and lengths of its axes. The method is sensitive to noise, and in order to prove its suitability for use in an actual robot system, several simulation and actual tests were run to verify the limits of the noise level

in the system. When these limits can be determined, the system is assumed to work reliably within them.

One goal of the thesis was to study the suitability of a posteriori error covariance as a criterion for sensing planning. The evaluation criteria of the covariance propagation were tested with different numbers of points and different levels of noise. The sensing planning method presented here uses an a posteriori error covariance matrix as an evaluation criterion and the tests show that the result is better than methods compared with the proposed method. The planning criterion was in the case of localization of a cube, the distance between sequential measurement points. The planned set of samples was compared other sets generated by different methods, and the results indicated that the method presented here planned the set of samples that gave the smallest level of uncertainties for the parameters to be estimated. The method is rather simple and straightforward to implement in many estimation processes.

The covariance propagation methods presented here were tested experimentally and compared with Monte Carlo simulation and actual tests. Attention was paid in the extensive actual tests to verifying the distribution of the set of samples. The results illustrate that the method presented here is sensitive to noise but copes within the noise level of the actual robot system. If the noise level increases 10 times relative to the noise in the actual system, the covariance propagation may fail, but the performance of the estimator can be extended to higher levels of noise by increasing the number of measurement points and the quality of the set of samples.

One motive for the work was to achieve a flexible robot-based manufacturing workcell. A lot of work can be done off-line before moving the actual robot. This work involves planning the measurements, simulating them and computing the spatial uncertainties. The final task is analysis of the results and a decision to increase the number of measurements or begin executing the actual motions. The system uses initial CAD information obtained from the workcell, including the robot model and work object models, but is able to adjust itself to changing conditions which are not whole programmed beforehand. This adjustment is based on sensor observations on the environment and can be carried out on-line. The results of this thesis represent a system which is operating in the way described here.

The modelling of spatial uncertainties can be applied to many other situations, e.g. evaluation of the accuracy of a 3D coodinate measuring machine or milling machine. Another field would be mobile machines, which is quite a famous field of robotics, so that the modelling of spatial uncertainties in these applications is not new (Dissanayake et al. 2001, Sukkarieh & Durrant-Whyte 2001).

8. Summary

Modelling and verification of spatial relationships in a robot workcell is one of the fundamental tasks in intelligent robotics. Almost every application has certain requirements for accuracy and repeatability that have to be verified in advance somehow. The requirements are exceptionally high in the production of single items or short series, when it is not economically feasible to make several test pieces or a heavy jigs for a single product. In addition to accuracy, flexibility plays an important role in intelligent robotics. The use of sensors to observe the robot's environment and computing of the spatial relations of the workcell can constitute a key task in fullfilling these requirements.

This thesis presents methods for estimating model parameters for hand-eye calibration, work object localization and surface model parameters and for simultaneous estimation of the spatial uncertainties in the estimated parameters. It is based on a Bayesian-form stochastic method and a priori information on the estimated parameters. An a posteriori error covariance matrix is used as a criterion for evaluating spatial uncertainties.

A sensing planning method based on the use of an a posteriori error covariance matrix as an evaluation criterion is presented here. The goal of the planning is to generate a set of measurement points that will give an optimal or close to optimal solution for the parameters to be estimated when the reference location of the pose is known in advance. The quality of the set of samples plays an important role in cases where the amount of measurement data is limited. Typical cases are measurement information obtained from a tactile sensor or a point laser rangefinder.

Special attention is paid to analysing spatial uncertainties and their behaviour under different conditions. The geometrical interpretation of the error covariance matrix is an uncertainty ellipsoid, which is illustrated using the directions of eigenvectors and lengths of eigenvalues. A careful analysis of these properties is presented in connection with the experimental tests. The differences between covariance propagation, Monte Carlo simulation and the actual tests are calculated and evaluated, and the behaviour of covariance propagation and Monte Carlo simulation when the noise level of the system changes is analysed.

One basis for a flexible robot cell is CAD information about the robot and the surrounding objects. Measurement points for the robot can be planned first using the sensing planning algorithms presented here. After that, motions can be programmed off-line to verify the configurations of the robot when running the path and to check for collisions. The methods presented in thesis can be carried out off-line or on-line. As a designer's tool, they assist in selecting appropriate components when constructing the robot cell. In on-line applications, the sensing planning algorithm generates a set of samples to be used for localizing a new work object. The intention is to embed the new features and algorithms into the larger software tools so that most of the features will be transparent to the user.

The equations composed for parameter estimation and the modelling of spatial uncertainties were tested in simulations and actual tests. The best methods for verifying the reliability of algorithms of this kind are careful simulations. Here Monte Carlo simulation was used and the results were also compared with those of actual large repeat tests. No case in which such close attention is paid to analysing the a posteriori error covariances has been reported before in the literature.

Future work on this research theme will be concentrated on algorithms for sensing planning at all phases of the calibration of a robot cell and the integrating of planning into an embedded on-line solution. A theme extending further into the future concerns modelling of the uncertainties in a robot working space. So far, uncertainty has been measured in the actual workcell, e.g. in various joint configurations, but the level of noise should somehow be estimated without actual measurements in each pose and without loss of accuracy.

References

ABB Robotics 2003a, Product Guide, IRB 7600, www.abb.com/robots.

ABB Robotics 2003b, Product manual IRB 1400. *ABB Flexible Automation M94A Rev 1. ABB Robotics Products AB*, S-721 68 Västerås, Swaden 3HAB 0009-43, 1993.

Abrams, S., Allen, P. & Tarabanis, K. 1999. Computing Camera Viewpoints in an Active Robot Work Cell. *Int. Journal of Robotics Research*, Vol. 18, No. 3, pp. 267–285.

Ahn, S.J., Effenberger, I., Rauh, W., Cho, H.S. & Westkämper, E. 2002. Automatic Segmentation and Model Identification in Unordered 3D-point Cloud. Proceedings of Optomechatronic Systems III, Proc. of SPIE 4902. Pp. 723–733.

Andreff, N., Horaud, R. & Espiau, B. 2001. Robot Hand-Eye Calibration Using Structure-from-Motion. *The Int. Journal of Robotics Research*, Vol. 20, No. 3, pp. 228–248.

Angeles, J., Soucy, G. & Ferrie, F. 2000. The Online Solution of the Hand-Eye problem. *IEEE Transactions on Robotics and Automation,* Vol. 16, No. 6, pp. 720–731.

Bernhardt, R. & Albright, S.L. 1993. *Robot Calibration.* Chapman & Hall. ISBN 0-412-491-40-0, 311 p.

Besl, P. & McKay, N. 1992. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 14, No. 2, pp. 239–256.

Boley, D., Steinmetz, E. & Sutherland, K. 1996. Robot Localization from Landmarks using Recursive Total Least Squares. Proceedings of the IEEE Int. Conference on Robotics and Automation. Minneapolis, Minnesota, pp. 1381–1386.

Borghi, G. & Caglioti, V. 1998. Minimum Uncertainty Explorations in the Self-Localization of Mobile Robots. *IEEE Transaction on Robotics and Automation*. Vol. 14, No. 6, pp. 902–911.

Borm, J-H. & Menq, C-H. 1991. Determination of Optimal Measurement Configurations for Robot Calibration Based on Observability Measure. *Int. Journal of Robotics Research*, Vol. 10, No. 1, pp. 51–63.

Burgard, W., Derr, A., Fox, D. & Cremers, A. 1998. Intergrating Global Position Estimation and Position Tracking for Mobile Robots: The Dynamic Markov Localization Approach. Proc. Of IEEE Int. Conference on Intelligent Robots and Systems. Victoria, Canada, pp. 730–735.

Caglioti, V. 1994. Uncertainty Minimization in the Localization of Polyhedral Objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 5, pp. 524–530.

Cameron, A. & Durrant-Whyte, H. 1990. A Bayesian Approach to Optimal Sensor Placement. *Int. Journal of Robotics Research*, Vol. 9, No. 5, pp. 70–88.

Charlebois, M., Gupta, K. & Payandeh, S. 1997. Shape Description of General, Curved Surfaces Using Tactile Sensing and Surface Normal Information. Proc. Of IEEE Int. Conference on Robotics and Automation. Albuquerque, New Mexico, pp. 2819–2824.

Chen, Y.H., Ng, C. T. & Wang, Y.Z. 1999. Generation of an STL File from 3D Measurment Data with User-Controlled Data Reduction. *Int. Journal of Advanced Manufacturing Technology*, Vol. 15, pp. 127–131.

Choi, B. K. 1991. *Surface Modeling for CAD/CAM*. Elsevier Science Publishers B.V. 329 p.

Cyber Optics, Sensor specifications brochure, May 1989.

Davison, A.J. & Kita, N., 2001. Sequential localization and map-building for real-time computer vision and robotics. *Int. Jour. Of Robotics and Autonomous Systems*, Vol. 36, pp. 171–183.

Dellaert, F., Fox, D., Burgard, W. & Thrun, S. 1999. Monte Carlo Localization for Mobile Robots. Proc. Of the IEEE Int. Conference on Robotics and Automation, Detroit, Michigan, pp. 1322–1328.

Deneb 2000. www.deneb.com/ Debeb, The Digital Manufacturing Company, 2000.

Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F. & Csorba, M. 2001. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, Vol. 17, No. 3, pp. 229–241.

Dornaika, F. & Horaud, R. 1998. Simultaneous Robot-World and Hand-Eye Calibration. *IEEE Transactions on Robotics and Automation*. Vol. 4, No. 4, pp. 617–622.

Drapikowski, P. & Nowakowski, T. 2002. 3D Object Modelling in Mobile Robot Environment Using B-spline Surfaces. Proc. Of the First Int. Symposium on 3D Data Processing Visualization and Transmission.

Durrant-Whyte, H. 1988. Uncertain Geometry in Robotics. IEEE Int. Conference in Robotics and Automation, Scottsdale, Arizona, pp. 646–652.

Eccles, J., Herd, J.T. & Duffy, N.D. 1989. Sensor Fusion for a Robotic Assembly Cell. Proc. Of 8th Int. Conf. of Robot Vision & Sensory Control, pp. 87–96.

Faugeras, O. & Hebert, M. 1986. The Repesentation, Recognition and Locating of 3-D Objects. *Int. Journal of Robotics Research*, Vol. 5, No. 3, pp. 27–52.

Gou, J., Chu, Y. & Li, Z., 1998. On the Symmetric Localization Problem. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 4, pp. 533–540.

Gu, X., Marefat, M. & Ciarallo, F. 1999. A Robust Approach for Sensor Placement in Automated Vision Dimensional Inspection. Proc. Of IEEE Int. Conf. On Robotics and Automation, Detroit, Michigan, pp. 2602–2607.

Gunnarsson, K. & Prinz F. 1987. CAD Model -based Localization of Parts in Manufacturing. *IEEE Computer*, pp. 66–74.

Hager, G. 1990. *Task-Directed Sensor Fusion – A Computational Approach*. Kluwer Academic Publishers, Norwell, MA, USA, 254 p.

Heikkilä, T., Järviluoma, M. & Voutilainen, O. 1993. An Object Locating Method with Uncertainties Applied to an Ultrasonic Multi-Sensor System. *Journal of Robotics and Mechatronics* 5(2), pp. 134–140.

Heikkilä, T., Matsushita, T. & Sato, T. 1989. Robot – Sensor Co-operation Planning for Visual Guidance. Proceedings of the 6th Scandinavian Conference on Image Analysis. Oulu, Finland, pp. 860–867.

Heikkilä, T., Rannanjärvi, L., Rintala, M. & Salonen, P. 1999a. A Holonic Shot Blasting System. Proc. Of Scandinavian Symposium of Robotics, pp. 193–200.

Heikkilä, T., Sallinen, M. & Järviluoma, M., 1999b. A Robot based surface inspection system with estimation of spatial uncertainties. Proceedings of SPIE - The International Society for Optical Engineering, Intelligent Robots and Computer Vison XVIII: Algorithms, Techniques, and Active Vision. Vol. 3837, pp. 242–253.

Heikkilä, T., Sallinen, M., Matsushita, T. & Tomita, F. 2000. Flexible Hand-Eye Calibration for Multi-camera Systems. Proc of IEEE/RSJ Int. Conf. On Intelligent Robots and Systems. Takamatsu, Japan, pp. 2292–2297.

Henderson, T. C., Allen, P., Cox, I., Mitiche, A., Durrant-Whyte, H. & Snyder, W. 1987. Workshop on Multisensor Integration in Manufacturing Automation. 31p.

Horaud, R. & Dornaika, F. 1995. Hand-eye Calibration. *Int. Journal of Robotics Research*. Vol. 14, No. 3, pp. 195–210.

Huang, Z. & Cohen, F. 1993. Affine-invariant Moments and B-splines for Object Recognition from Image Curves. Proceedings of SPIE – The

International Society for Optical Engineering, Applications of Artificial Intelligence. Machine Vision and Robotics. Vol. 1964, pp. 2–12.

Hutchinson, S. & Kak, A. 1989. Planning Sensing Strategies in Robot Work Cell with Multi-Sensor Capabilities. *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 6, pp. 765–783.

Ikeuchi, K. & Robert, J-C. 1991. Modeling Sensor Detectability with the VANTAGE Geometric/Sensor Modeler. *IEEE Transactions on Robotics and Automation*, Vol. 7, No. 6, pp. 771–784.

Joshi, R. & Sanderson, A. C. 1999. Minimal Representation Multisensor Fusion Using Differential Evolution. IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 29, No. 1, pp. 63–76.

Järviluoma, M. 1991. Object Model Correction with Iterative Estimation. Report C6, University of Oulu, Systems Engineering Laboratory, 41 p. ISBN 951-42-3104-X.

Järviluoma, M., & Heikkilä, T. 1997. Modelling the Positioning Accuracy of Sensor Based Robots. SPIE Vol. 3208. 0277-786X/97, pp. 540–550.

Järviluoma, M. & Heikkilä, T. 1994. On the representation and propagation of uncertainty in position and orientation relations. Proceedings of the SPRANN '94. IMACS International Symposium on Signal Processing, Robotics and Neural Networks, pp. 49–54.

Kalman, R. E. 1960. A New Approach to Linear Filtering and Prediction Problems. Journal of Basic Engineering. March 1960, pp. 35–45.

Karimäki, V. 1993. Monte Carlo menetelmät, opintomoniste. Report series HU – SEFT I 1993-01, Research institute for high energy physics, University of Helsinki. [In Finnish], ISSN 0788-3587, 93 p.

Kemmotsu, K. & Kanade, T. 1995. Uncertainty in Object Pose Determination with Three Light-Stripe Range Measurements. *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 5, pp. 741–747.

Kitamura, Y., Sato, H. & Tamura, H. 1990. An Expert System for Industrial Machine Vision. Proc. Of 10th Int. Conf. on Pattern Recognition. Atlantic City, USA. pp. 771–774.

Klein, C.A. & Blaho, B.A. 1987. Dexterity measures for the design and control of kinematically redundant manipulators. *Int. Jour. Of Robotics Research*, Vol. 6, No. 2, pp. 72–83.

Knopf, G. & Abouhossein, A. 2000. Adaptive reconstruction of anatomical surfaces from human body measurements. IEEE Int. Conference on System, Man & Cybernetics, Nashville, USA, pp. 1181–1185.

Kreyszig, E. 1993. *Advanced Engineering Mathematics.* New York, John Wiley & Sons, 1271 p. ISBN 0-471-59989-1.

KUKA Robotics 2003. Product Manual, KUKA KR 500, L420PA/1, www.kine.fi.

Li, Z., Gou, J. & Chu, Y. 1998. Geometric Algorithms for workpiece Localization. *IEEE Transactions on Robotics and Automation*, Vol. 14, No. 6, pp. 864–878.

Liu, F. & Hasegawa, T. 2001. Reconstruction of Curved Surfaces Using Active Tactile Sensing and Surface Normal Information. Proc. Of IEEE Int. Conf. on Robotics and Automation. Seoul, Korea, pp. 4029–4034.

Lowe, D. 1985. Perceptual organisation and Visual Recognition. Kluwer Academic, Hingham, U.K., 162 p.

Luo, R.C. & Kay, M. 1989. Multisensor Integration and Fusion in Intelligent Systems. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 19, No. 5, pp. 901–931.

Luo, R.C. & Lin, M-H. 1988. Robot multi-sensor fusion and integration: Optimum estimation of fused sensor data, Proc. Of IEEE Int. Confrence of Robotics and Automation, pp. 1076–1081.

Mantripragada, R. & Whitney, D. 1999. Modeling and Controlling Variation Propagation in Mechanical Assemblies Using State Transition Models. *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 1, pp. 124–140.

Matsuyama, T. 1988. Expert Systems for Image Processing – Knowledge-Based Composition of Image Analysis Process. Proc. Of 9th Int. Conf. on Pattern Recognition, Rome, Italy, pp. 125–133.

Maver, J. & Bajcsy, R. 1993. Occlusions as a Guide for Planning the Next View. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 15, No. 5, pp. 417–433.

Maybeck, P. 1979. Stochastic Models, Estimation and Control, Volume 1. Academic press, 423 p.

MEL 1998. LASER-Scanner zur Konturerfassung, Linien-Scanner M2D. MEL Mikroelektronik GmbH, Bahnhofstr. 28a, D-85386 Eching/Germany.

Menq, C-H., Yau, H-T. & Lai, G-Y. 1992. Automated Precision Measurement of Surface Profile in CAD-Directed Inspection. *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 2, pp. 268–278.

Mooring, B.W. Roth, Z.S. & Driels, M.S. 1991. Fundamentals of Manipulator Calibration. John Wiley & Sons, inc. 324 p.

Moreno, L., Armingol, J., Garrido, S., Escalera, A. & Salichs, M. 2002. A Genetic Algorithm for Mobile Robot Localization Using Ultrasound Sensors. *Journal of Intelligent and Robotic Systems*, Vol. 34, pp. 135–154.

Motta, J.M. & McMaster, R.S. 1997. Improving Robot Calibration Results Using Modeling Optimization. IEEE Catalog Number 97TH8280.

Nahvi, A. & Hollenbach, J. 1996. The Noise Amplification Index for Optimal Pose Selection in Robot Calibration. Proc. Of IEEE Int. Conf. On Robotics and Automation, pp. 647–654.

Nahvi, A., Hollenbach, J. & Hayward, V. 1994. Calibration of a Paraller Robot Using Multiple Kinematic Closed Loops. Proc. Of IEEE Int. Conf. On Robotics and Automation. San Diego, California, pp. 407–412.

Nakamura, Y. & Xu, Y. 1989. Geometrical Fusion Method for Multi-Sensor Robotic Systems. IEEE Int. Conf. on Robotics and Automation. Vol. 2, pp. 668–673.

Nilsson, B., Nygårds, J. & Wernersson, Å. 1996. On 2D Posture Estimation with Large Initial Orientation Uncertainty and Range Dependant Measurement Noise. ICARV'96, Signapore, Dec 1996.

Nygårds, J. & Wernersson, Å. 1998. On Covariances for Fusing Laser Rangers and Vision with Sensors Onboard a Moving Robot. Proceedings of the 1998 IEEE/RSJ Int. Conference on Intelligent Robots and Systems. Victoria, B.C. Canada, pp. 1053–1059.

Nilsson, B. & Nygårds, J. 1996. Sensor Motion Plannig with Active Uncertainty Reduction: Gripping and Doching Tasks. IEEE Int. Conf. On Robotics and Automation, Minnesota.

Olson, C. 1999. Subpixel Localization and Uncertainty Estimation Using Occupancy Grids. Proc. Of IEEE Int. Conf. on Robotics & Automation. Detroit, Michican, pp. 1987–1992.

Paul, R.P. 1983. Robot Manipulators. Mathematics, Programming and Control. MIT Press, 279 p. ISBN 0-262-16082-X.

Ramsli, E. 1991. Probability Distribution of Repeatability of Industrial Robots. *The International Journal of Robotics Research*, Vol. 10, No. 3, pp. 276–283.

Sakane, S., Sato, T. & Kakikura, M. 1987. Model-based Planning of Visual Sensor Using Hand-Eye Action Simulator HEAVEN. Proceedings of the 3rd International Conference on Advanced Robotics. Versailles, pp. 163–174.

Sakane, S., Sato, T., Okoshi, H. & Kakikura, M. 1993. Distributed Sensing System with 3D Model-Based Agents. Proc. Of the IEEE/RSJ Int. Conf. On Intelligent Robots and Systems, Yokohama, Japan.

Sallinen, M. 2001. Robotized Surface Inspection: Range Data Acquisition with Estimation of Spatial Uncertainties. University of Oulu, Systems Engineering Laboratory, Report C25, 74 p. ISBN 951-42-6568-8.

Sallinen, M., Heikkilä, T. & Järviluoma M. 1999. Robotized surface inspection with carefull design of measurements. Proceedings Scandinavian Symposium on Robotics. Robotic Society in Finland, pp. 133–141.

Sallinen, M. & Heikkilä, T. 2000a. A Simple Hand-Eye Calibration for A 3D Laser Range Sensor. IFIP/IEEE Int. Conf. On Information Technology for Balanced Automation Systems, Berlin, Germany, pp. 421–430.

Sallinen, M. & Heikkilä, T. 2000b. Flexible Localization for CAD-Based Robotics. Proc. of SPIE – The International Society for Optical Engineering, Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision. SPIE – The International Society for Optical Engineering. Bellingham, Washington. Vol. 4197, pp. 130–139.

Sallinen, M. & Heikkilä, T. 2001a. Pose-dependent uncertainty modelling of the hand-eye calibration. IEEE Int. Conf. On Int. Eng. Systems, Tampere, Finland, pp. 313–318.

Sallinen, M. & Heikkilä, T. 2001b. Estimation of the surface model parameters and analysis of spatial uncertainties. IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI2001). Baden-Baden, Germany, pp. 209–214.

Sallinen, M. & Heikkilä, T. & Rintala, M. 2000. Sensor fusion with spatial uncertainties in a Holonic multirobot workcell. IEEE International Conference on System, Man & Cybernetics, pp. 1428-1437.

Sallinen, M. & Heikkilä, T. 2002. Effect of Noise Level for Estimating the Spatial Uncertainties of Workobject Localization. Proceedings of SPIE Optomechatronic Systems III. Stuttgart, Germany, Vol. 4902 (2002), pp. 334–343.

Sallinen, M., Heikkilä, T. & Sirviö, M. 2001. A Robotic Deburring System of Foundry Castings Based on Flexible Workobject Localization. Proc. of SPIE Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision. Vol. 4572 (2001), pp. 476–487.

Sanderson, A. 1999. Assemblability Based on Maximum Likelihood Configuration of Tolerances. *IEEE Transactions on Robotics and Automation*, Vol. 15, No. 3, pp. 568–572.

Schutz, C. & Hugli, H. 1995. Towards the Recognition of 3D Free-form Objects. Intelligent Robots and Computer Vision XIV: Algorithms, Techniques, Active Vision, and Materials Handling, pp. 476–484.

Sheng, W., Xi, N., Song, M., Chen, Y. & Rankin, J. 2000. Automated CAD-Guided Automobile Part Dimensional Inspection. Proc. Of IEEE Int. Conf. On Robotics and Automation, San Fransisco, CA, pp. 1157–1162.

Sheng, W., Xi, N., Song, M. & Chen, Y. 2001. Graph-based Surface Merging in CAD-guided Dimensional Inspection of Autonomotive Parts. Proc. Of. IEEE Int. Conference on Robotics and Automation, Seoul, Korea, pp. 3127–3132.

Shirai, Y. 1987. Three-Dimensional Computer Vision. Springler-Verlag. 296 p. ISBN 3-540-15119-2.

Shiu, Y. & Ahmad, S. 1989. Calibration of Wrist-Mounted Robotic Sensors by Solving Homogenous Transform Equations of the Form AX=XB. *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 1, pp. 16–29.

Smith, R.C. & Cheeseman, P. 1986. On the representation and Estimation of Spatial Uncertainty. *The International Journal of Robotic Research*, Vol. 5, No. 4, pp. 56–68.

Su, S-F. & Lee, C.S. 1992. Manipulation and Propagation of Uncertainty and Verification of Applicability of Actions in Assembly Tasks. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 6, pp. 1376–1389.

Sukkarieh, S. & Durrant-Whyte, H. 2001. Towards the development of Simultaneus Localization and Map Building for and Unmanned Air Vehicle. 3rd International Conference on Field and Service Robotics, pp. 193–200.

Takahashi, H. & Tomita, F. 1988. Self Calibration of Stereo Cameras. *Proc. Of ICCV*. pp. 123–128.

Tarabanis, K.A., Allen, P. & Tsai, R. 1995a. A Survey of Sensor Planning in Computer Vision. *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 1.

Tarabanis, K.A., Tsai, R.Y. & Allen, P.K. 1995b. The MVP Sensor Planning System for Robotic Vision Tasks. *IEEE Transactions on Robotics and Automation* Vol. 11, No. 1, pp. 72–85.

Tarabanis, K., Tsai, R. & Kaul, A. 1996. Computing Occlusion-Free Viewpoints. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 3, pp. 279–292.

Taylor, R. H. & Rajan, V. T. 1988. The Efficient Computation of Uncertainty Spaces for Sensor-Based Robot Planning. Proceedings of the 1988 IEEE International Workshop on Intelligent Robots and Systems (IROS '88), pp. 231–236.

Tomita, F., Yoshimi, S., Ueshiba, T., Kawai, Y., Sumi, Y., Matsushita, T., Ichimura, N., Sugimoto, R. & Ishiyama, Y. 1998. R&D of Versatile 3D Vision System VVV. Proc. IEEE Int. Conf. on System, Man and Cybernetics, pp. 4510–4516.

Tsai, R. & Lenz, R. K. 1989. A New Technique for Fully Autonomous and Efficient Robotics Hand/Eye Calibration. *IEEE Transctions on Robotics and Automation*. Vol 5, pp. 345–358.

Wang, C-C. 1992. Extrinsic Calibration of a Vision Sensor Mounted on a Robot. *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 2, pp. 161–175.

When, H. & Belanger, P. 1997. Ultrasound-Based Robot Position Estimation. *IEEE Transactions on Robotics and Automation*, Vol. 13, No. 5, pp. 682–692.

Wei, G.-Q., Arbter, K. & Hirzinger, G. 1998. Active Self-Calibration of Robotic Eyes and Hand-Eye Relationships with Model Identification. *IEEE Transactions of Robotics and Automation,* Vol. 14, pp. 158–166.

Zhang, Y. & Ji Q. 2001. Camera Calibration with Genetic Algorithms. Proc. Of IEEE Int. Conf. on. Robotics and Automation. Seoul, Korea, pp. 2177–2182.

Zhuang, H. 1998. Hand/Eye Calibration for Electronic Assembly Robots. *IEEE Transactions on Robotics and Automation,* Vol. 14, No. 4, pp. 612–616.

Zhuang, H., Kuanchih, W. & Roth, Z. 1994a. Optimal Selection of Measurement Configurations for Robot Calibration using Simulated Annealing. Proc. Of IEEE Int. Conference On Robotics and Automation. San Diego, California, pp. 393–398.

Zhuang, H. & Roth, Z.S. 1996. Camera Aided Robot Calibration, CRC Press, 363 p. ISBN 0-8493-9407-4.

Zhuang, H., Roth, Z.S. & Sudhakar, R, 1994b. Simultaneous Robot/World and Tool/Flange Calibration by Solving Homogenous Transformation Equations of the Form AX=YB. *IEEE Transactions on Robotics and Automation*, Vol. 10, No. 4, pp. 549–554.

Zhuang, H., Wang, K. & Roth, Z. 1995. Simultaneous Calibration of a Robot and a Hand-Mounted Camera. *IEEE Transactions on Robotics and Automation*, Vol. 11, No. 5, pp. 649–660.

Published by

Series title, number and report code of publication

VTT Publications 509
VTT–PUBS–509

Author(s)
Sallinen, Mikko

Title

Modelling and estimation of spatial relationships in sensor-based robot workcells

Abstract

Requirements for verifying spatial relations in robot workcell in terms of accuracy and repeatability are increasing. To satisfy the requirements of overall geometric performance and flexibility in a robot system, a sensor-based, intelligent robot can be used. One of the goals of this thesis was to develop a flexible, CAD-based robot system. Modern applications and cost-effective production require off-line programming, and the difference between off-line programming systems and actual robot workcells has to be illustrated somehow in order to verify the gap between simulation models and actual robot systems. A method for modelling spatial uncertainties in a robot system is presented here, based on Bayesian-form estimation of model parameters and of the spatial uncertainties in the resulting parameters. The calibration of the robot workcell consists of several phases: hand-eye calibration, localization of the work object and estimation of model parameters for the work object surface. After localizing the work object, a finalization task can be carried out, e.g. inspection, manufacturing or assembly. A synthesis method of sensing planning that uses the same form of modelling spatial uncertainties is also presented. The deviation between covariance propagation models and actual systems is reduced by using detailed noise models of the robot system, including measured noise, at different phases in the calibration.

Keywords
intelligent robots, parameter estimation, spatial uncertainties, pose estimation, sensing planning

Activity unit
VTT Electronics, Kaitoväylä 1, PL 1100, 90571 Oulu

ISBN	Project number
951–38–6247–X (soft back ed.) 951–38–6248–8 (URL:http://www.vtt.fi/inf/pdf/)	

Date	Language	Pages	Price
October 2003	English	218 p.	E

Name of project	Commissioned by
	Technology Development Centre of Finland Tekes, Technical Research Centre of Finland VTT, Finnish Work Environment Fund TSR, Tauno Tönning Foundation, Foundation for Technology in Finland TES

Series title and ISSN	Sold by
VTT Publications 1235–0621 (soft back ed.) 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)	VTT Information Service P.O.Box 2000, FIN–02044 VTT, Finland Phone internat. +358 9 456 4404 Fax +358 9 456 4374

Requirements for verifying spatial relations in robot workcell in terms of accuracy and repeatability are increasing. To satisfy the requirements of overall geometric performance and flexibility in a robot system, a sensor-based, intelligent robot can be used. Modern applications and cost-effective production require off-line programming, and the difference between off-line programming systems and actual robot workcells has to be illustrated somehow in order to verify the gap between simulation models and actual robot systems. A method for modelling spatial uncertainties in a robot system is presented here, based on Bayesian-form estimation of model parameters and of the spatial uncertainties in the resulting parameters. The calibration of the robot workcell consists of several phases: hand-eye calibration, localization of the work object and estimation of model parameters for the work object surface. A synthesis method of sensing planning that uses the same form of modelling spatial uncertainties is also presented.

Tätä julkaisua myy
VTT TIETOPALVELU
PL 2000
02044 VTT
Puh. (09) 456 4404
Faksi (09) 456 4374

Denna publikation säljs av
VTT INFORMATIONSTJÄNST
PB 2000
02044 VTT
Tel. (09) 456 4404
Fax (09) 456 4374

This publication is available from
VTT INFORMATION SERVICE
P.O.Box 2000
FIN–02044 VTT, Finland
Phone internat. +358 9 456 4404
Fax +358 9 456 4374

ISBN 951–38–6247–X (soft back ed.)
ISSN 1235–0621 (soft back ed.)

ISBN 951–38–6248–8 (URL: http://www.vtt.fi/inf/pdf/)
ISSN 1455–0849 (URL: http://www.vtt.fi/inf/pdf/)

VTT PUBLICATIONS 509 Modelling and estimation of spatial relationships in sensor-based robot workcells Mikko Sallinen