Photogrammetric Calibration of Mobile Robot Kinematics

by

Louis P. McNamee

A thesis submitted to the
School of Graduate Studies and Research
in partial fulfillment of the Requirements for the Degree of

Master of Applied Science
in Electrical Engineering

Ottawa-Carleton Institute for Electrical and Computer Engineering

School of Information Technology and Engineering
(Electrical and Computer Engineering)

June, 2003

Abstract

Numerous simulation and control applications of mobile robotics require accurate kinematic models. A kinematic model relates the position and orientation of a robot to control inputs. This thesis proposes a non-intrusive methodology to calibrate kinematic models for wheeled mobile robots. Model calibration requires accurate measurement of kinematic state and robust estimation techniques to account for process and measurement uncertainty. A photogrammetric camera system is used to measure the kinematic trajectory of wheeled mobile robot. A fully projective formulation of Lowe's pose recovery algorithm is used to estimate robot pose from monocular images. A derivative free form of the extended Kalman filter is applied to the time series pose data to estimate robot model parameters. Experimental results are presented for a differential wheeled mobile robot. Calibration by photogrammetry is shown to be viable for typical mobile robot applications.

Acknowledgment

I thank my supervisor Dr. Emil Petriu for his guidance and support throughout the conduct of the research presented herein. I thank Dr. Lowell Winger for his guidance with respect to image processing algorithms. I thank my wife Ainley for her endless support and patience.

Contents

1. Introduction 1

1.1 Thesis Motivation 1

1.2 Research Questions 2

1.3 Thesis Organization 2

1.4 Thesis Contributions 4

2. Robot Calibration 5

2.1 Robot Calibration Problem 5

2.2 Related Work in Robot Calibration 5

2.2.1 Manipulator Calibration 5

2.2.2 Mobile Robot Calibration 8

2.2.3 Summary of Related Work 11

2.3 Proposed Mobile Robot Calibration Methodology 11

3. Wheeled Mobile Robot Kinematic Models 15

3.1 Kinematic Model Development 15

3.1.1 Kinematic Equations of Motion by Muir and Newman 16

3.1.2 ICR Based Kinematic Models 20

3.2 Unicycle WMR Kinematic Model 22

3.3 Differential WMR Kinematic Model 24

3.4 Ackerman Steered WMR Model 27

3.5 Augmented State Vector 31

4. Computer Vision and Photogrammetry 34

4.1 Photogrammetry 35

4.2 Imaging Process Model 37

4.3 Camera Calibration 41

4.3.1 Direct Linear Transformation 42

4.3.2 Tsai's Camera Model 43

1.1.1 Modified Tsai Camera Model 49

4.4 Image acquisition and feature detection 53

4.5 Pose Estimation 61

5. Model Parameter Estimation 67

5.1 Linear Kalman Filter 68

5.2 Extended Kalman Filter (EKF) 75

5.2.1 EKF Comments 80

5.3 Second Order Divided Difference Filter 81

5.3.1 DD2 Filter Algorithm 91

6. Mobile Robot Calibration Results 93

6.1 Experimental Apparatus 93

6.1.1 Calibration Computer 94

6.1.2 Calibration Software 94

6.1.3 Camera 96

6.1.4 *Robot Under Calibration* *96*
6.2 *Robot Calibration Procedure* *98*
6.2.1 *Stage 1: Preparation for Calibration* *99*
6.2.2 *Stage 3: Camera Calibration* *100*
6.2.3 *Stage 3: Robot Calibration Experiment* *101*
6.2.4 *Stage 4: Calibration Processing* *103*
6.3 *Differential Wheeled Mobile Robot Calibration Experiments* *106*
6.4 *Parameter Estimation* *108*

7. Discussion 112

7.1 *Observable, Controllable and Identifiable Processes* *112*
7.2 *Robot Calibration Metrics* *113*
7.3 *Calibration Error Model* *114*
7.3.1 *Kinematic Model Error* *115*
7.3.2 *Control Vector Error* *115*
7.3.3 *Image Feature Position Error* *116*
7.3.4 *Pose Estimation Error* *118*
7.3.5 *Calibration Error* *121*

8. Conclusion 123

8.1 *Summary of Results* *123*
8.2 *What accuracy or fidelity is required for a mobile robot kinematic model?* *123*
8.3 *Is photogrammetry a suitable measurement scheme for mobile robot calibration?* *124*
8.4 *How does calibration accuracy depend on the factors under user control?* *124*
8.5 *How can robot calibration accuracy be improved?* *124*
8.6 *What a priori knowledge is necessary for mobile robot calibration?* *124*
8.7 *Future Research* *125*

9. References 126

1. Introduction

This thesis considers the problem of mobile robot calibration. Robot calibration is the determination of a mathematical model that approximates the kinematics of a robot in some sense of optimality. The primary utility of mobile robot calibration is the improved control of robot motion. A secondary benefit is the ability to accurately predict robot position in the future as a function of control inputs. This secondary benefit is essential for navigation and path planning applications.

Typically, robot calibration involves identification or estimation of parameters for a predetermined robot model from some set of measurement data. This thesis investigates the application of non-linear estimators to determine robot kinematic parameters from photogrammetric data of pre-programmed robot maneuvers.

The proposed calibration method is tested on a two wheeled differential wheeled mobile robot. The effectiveness of photogrammetry as a measurement tool of kinematic state is considered. The effectiveness of calibration by the estimation of parameters is considered.

An external photogrammetric system is used to measure robot kinematic state for the following reasons:

a. Many mobile robots have limited computational capability and are often bandwidth limited by low-speed interfaces.

b. A simple external method may be applied to all WMRs regardless of sensor suite and processors.

c. External calibration allows calibration of various robot configurations.

d. External calibration requires a single target on the robot with no requirement for multiple targets throughout the robot's calibration space.

1.1 Thesis Motivation

The motivation of this thesis is the need for a non-intrusive method to determine the kinematic models of mobile robots. The process of determining a mobile robot kinematic model is termed as calibration. The ideal calibration process results in an optimal kinematic model. Simulation of the calibrated model will match actual kinematic behavior closely.

Most mobile robot developers assume a theoretical model for robot control and rely on closed loop control to correct for model errors. Models that are more accurate will reduce the reliance upon closed loop control and the requirement for very accurate sensors. Mobile robot calibration is an area with many unsolved problems. Typically, robot calibration involves identification or estimation of parameters for a predetermined robot model from some set of measurement data. Accurate mobile robot models bring significant benefit to the rapidly expanding field of mobile robotics. The primary utility of mobile robot calibration is the improved control of mobile robot motion.

1.2 Research Questions

This thesis answers the following questions:

a. What accuracy or fidelity is required for a mobile robot kinematic model?

b. Is photogrammetry a suitable measurement scheme for mobile robot calibration?

c. How does calibration accuracy depend on the factors under user control?

d. How can robot calibration accuracy be improved?

e. What a priori knowledge is necessary for mobile robot calibration?

1.3 Thesis Organization

This thesis is organized as follows:

Chapter 2 summarizes related work in robot calibration and introduces the proposed calibration methodology. Considerable work has been conducted for manipulator calibration while little research effort has occurred on mobile robot calibration.

Chapter 3 develops ideal kinematic models for several mobile robot configurations. The kinematic models are founded upon physical constraints such as the non-holonomic constraint.

Chapter 4 describes the theory and practice of photogrammetry. An algorithm is developed to automatically detect markers that will be used for robot pose measurement. The camera image generation model is derived from the perspective transformation. Tsai's camera calibration scheme is derived. Lowe's pose estimation algorithm is shown to be an elegant and simple solution to the pose estimation problem.

Chapter 5 proposes the divided difference filter as the most suitable estimator for model calibration. The divided difference filter, which is a derivative-free implementation of the extended Kalman filter, is developed to estimate robot parameters from pose data.

Chapter 6 describes the calibration apparatus. The calibration process is summarized. Calibration results are presented for kinematic experiments performed with a differential wheeled mobile robot.

Chapter 7 discusses factors affecting the effectiveness of the proposed calibration process. Requirements that the process be observable, controllable and identifiable are identified. Definitions of calibration performance metrics are proposed. An error model of the sensitivity of calibration error to measurement errors is established.

Chapter 8 draws conclusions about the success of the proposed method. Answers to the research questions in Section 1.2 are given.

Chapter 9 lists cited references.

1.4 Thesis Contributions

This thesis makes the following contributions:

1. Demonstration of photogrammetry as a non-intrusive robot calibration process.

2. Definition of robot calibration metrics.

3. Demonstration of the suitability of derivative-free estimators for calibration.

2. Robot Calibration

This chapter introduces the field of robot calibration. The need for robot calibration is shown. Related work in manipulator and mobile robot calibration is summarized. Finally, the proposed mobile robot calibration methodology is introduced.

2.1 Robot Calibration Problem

The primary utility of robot calibration is the improved control of robot motion. The need for robot calibration has existed since the first industrial manipulators were introduced in the early 1960's[1]. This is because many robotic control systems are model based. Such systems rely on prior knowledge of the robot model. A poor or inaccurate model of the robot will reduce system effectiveness by the introduction of systematic errors. Conversely, an optimal model will reduce control errors. Accurate robot model calibration will increase mobile robot effectiveness by increased control fidelity.

The problem of robot calibration is to determine a mathematical model. The model determination is optimized according to some metric, usually related to the robot application. The most common application of robot kinematic models is in control algorithms and simulations.

The robot model is generally given as a system of kinematic and dynamic equations whose parameters characterize a specific mechatronic system.

2.2 Related Work in Robot Calibration

2.2.1 Manipulator Calibration

The calibration of robotic manipulators is an established field [BA93]. Considerable work has been conducted on the calibration of robotic manipulators. *Robot Calibration* by Bernhardt and Albright [BA93] is an excellent reference for current technology used in manipulator calibration.

Everett proposed a formulation [Everett93] of robot manipulator calibration as an optimization problem. A real robot is represented as a point in a space of real robots whose basis and dimension are unknown. A robot model is represented as a point in a model space of known basis and dimension. Ideally the model space is equivalent to or is a subset of the real robot space. However, the two spaces may differ and perfect representation may not be possible.

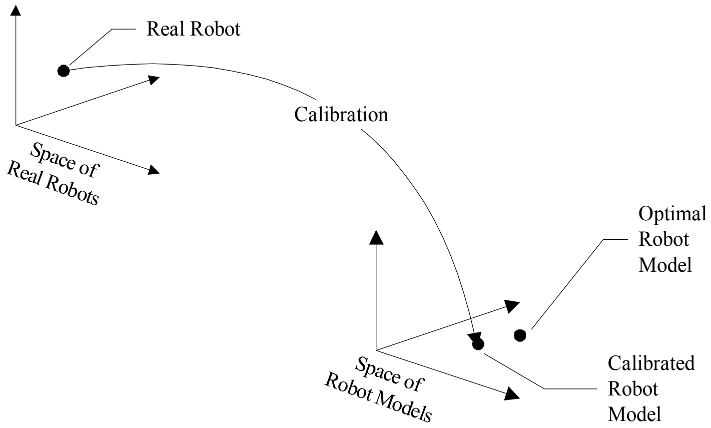


Figure 2-1 Calibration as a mapping from real robot space to model space

The calibration process attempts to determine the point in model space that optimally characterizes the real robot. The mapping from real robot to robot model is inferred through measurement. An error objective function defined on the model space is defined as some metric of the distance from the optimal model point to the calibrated model point. The quality of calibration is given by minimizing the calibration objective function.

The goal of manipulator calibration is to calculate a more accurate model of the real robot from measurement, so that the error between the programmed pose[2] and actual pose is minimized. Many industrial robots utilize calibration for off-line programming, which has been shown more cost effective than on-line programming [Stark93].

Robot calibration research is generally grouped into the following three areas:

a. Measurement systems and techniques,

[1] General Motors installed a Unimation robotic manipulator in 1962. [McK91]

[2] Pose is a commonly used term to refer to the position and orientation of an object. In three dimensions pose is given by the six-tuple {x, y, z, roll, pitch, yaw}. In two dimensions pose is given by the triple {x, y, yaw}.

b. Robot kinematic and error parameter modeling, and

c. System identification algorithms.

Measurement research involves the development of technologies and methods to measure and record robot pose with required accuracies. The industrial setting in which most manipulators are used prefers off-line programming (OLP). Off-line programming is advantageous in that it minimizes the disruption to production lines when a robot task/program is changed. In addition OLP has great economic benefit in that it allows robot work programs to be distributed across many robot systems. However, off-line programming has proven to be difficult in practice.

Most manipulator systems use joint measurement for control rather than absolute tool position. The use of relative position rather than absolute positioning imposes open loop control mechanisms on most robotic manipulator systems, which is generally less effective than closed loop control.

One solution to improve the effectiveness of open loop control is to relate tool position to joint position by a kinematic model of the manipulator [BA93], [Stark93].

Many measurement systems have been developed, with large variation in complexity, measurement speed, and required operator skills. Manipulator kinematic measurement is divided into two common methods:

a. Pose error measurement

b. Pose matching

Pose error measurement methods require a manipulator to follow commanded trajectories. Measurement tools are used measure the actual manipulator trajectory. The pose error is calculated as the difference between the measured and commanded trajectories. State of the art measurement accuracy is of the order 0.1 mm over the robot measurement workspace [BA93]. Common sensors used for pose error measurement are:

a. Theodalites

b. Laser trackers

c. Cable Potentiometers

Theodalite systems based on 1 arc-second instruments sighting at 10 meters have measurement accuracy near 0.05 mm [BA93]. The laser tracker has a similar measurement accuracy, which is close to or less than the repeatability of most commercial robots. These sensors are complicated and require skilled operators. Special tools must be attached to the robot. These systems also require regular calibration and considerable set up time, before use [BA93].

Cable potentiometers measure the amount of cable spooled from drums. Most cable potentiometer systems consist of 3 or 4 cables. Typical accuracy of such measurement systems is 0.2 mm. These systems are simple and easy to set up. However, robot movement is often restricted. Thus cable based measurement systems are not suitable for mobile robot applications.

In pose matching methods, the robot is driven to a known location, and the pose calculated by the robot controller is recorded. The difference between the known pose and that calculated by the controller leads to the pose error. The two poses are different because the controller uses the nominal robot model and measured joint angles at the encoders to calculate the pose. Pose matching requires much less sophisticated equipment than pose error measurement

Trevelyan [Trev96] uses a Perspex table, a grid and spirit level to calibrate a robot manipulator by aligning a pointer tool with a series of grid intersections with respect to the grid. The measurement accuracy of the method is approximately 0.3 mm for position and 4 mrad for orientation. Hayati [HTR88] uses a simple pointer attached to a force/torque sensor to indicate robot contact with a calibration plate. The measurement accuracy of this method is 0.1 mm with poorer rotational accuracy

2.2.2 Mobile Robot Calibration

Research conducted on mobile robot calibration is limited to a few noteworthy works. These works are described below.

2.2.2.1 UMBmark [BEF96a]

Borenstein and Feng proposed a systematic method, called the University of Michigan Benchmark test (UMBmark), to calibrate systematic odometric errors in mobile robots [BEF96a], [BF94], [BF95a], [BF95a], [BF95b], and [BF96b]. The University of Michigan Benchmark (UMBmark) test is optimized for differential drive robots, but the method can be used for other kinematic configurations. Results are presented from experiments with six mobile robots. The six robots tested were:

a. TRC LabMate,

b. Cybermotion K2A,

c. CLAPPER (4-DOF platform),

d. Remotec Andros,

e. Remotec Andros with "Basic Encoder Trailer", and

f. TRC LabMate with Smart Encoder Trailer.

UMBmark introduces a method for measuring odometry errors in mobile robots, and for expressing the errors quantitatively. Systematic errors due to kinematics are distinguished from non-systematic errors due to dynamics. An ultrasonic sensor system is used to measure robot absolute position. Odometric measurement is compared to the absolute measurements. A systematic robot trajectory is proposed to identify systematic error including corrections to wheel base and wheel radius. The calibration method is experimentally shown to reduce vehicle position errors by an order of magnitude.

2.2.2.2 Event-based Kalman Filtering Project [BLA+99] [BLN+98]

The "Event-based Kalman Filtering" project, carried out at the Technical University of Denmark (DTU), explored calibration of an Autonomous Guided Vehicle (AGV) and its sensors [BLA+99] [BLN+98]. The AGV was a differential WMR with one rear caster. The project investigated estimation of model parameters from multirate sensory data from a video camera, ultrasonic range finder, laser range finder, and wheel shaft encoders.

Detailed models of the AGV were derived, including the vehicle dynamics and kinematic transformations [Nørgaard97a]. Stochastic models of the AGV sensors were also derived to support derivation of the system state equations [Nørgaard97b]. An extended Kalman filter was used to combine the varied sensor data, estimate the AGV kinematic state, and identify model parameters.

An active calibrated camera mounted on the AGV was used to detect artificial guide marks in the robot workspace. The authors found the problem of joint state and parameter estimation for an AGV equipped with wheel encoders and a camera for guide mark detection, to be quite challenging. The extended Kalman filter was not easily applied, as it required linearizations that were quite complex to derive. The authors overcame the difficulty of linearization difficulty by deriving a derivative free Kalman filters for state estimation of nonlinear systems [Nørgaard98] [Nørgaard99]. The filters are less complex than the EKF and provide equivalent accuracy.

Experiments were conducted with a differential AGV to estimate vehicle position, orientation and kinematic parameters. The kinematic parameters of wheel radius and the distance between the driving wheels were estimated to within 1.5%.

2.2.2.3 Statistical Online Self-Calibration [RT99]

Nicolas Roy and Sebastian Thrun [RT99] proposed an online statistical method for calibrating the odometry of mobile robots. The calibration algorithm uses the robot's sensors to automatically calibrate the robot as it operates. An incremental maximum likelihood algorithm is applied to measured data during robot operation.

The approach was demonstrated with a RW1 BL2 4-wheeled synchro-drive robot in two large-scale environments: the Smithsonian National Museum of American history and the Carnegie Museum of Natural History. The robot was equipped with an array of 24 sonar sensors and a laser range finder. The Carnegie Museum experiments reduced the robot odometric error from 18 meters to 3.05 meters over 269 meter path length. The reduced odometric error represents a calibration error of about 1.1%.

2.2.3 Summary of Related Work

Each of the mobile robot calibration techniques described above make specific demands on the robot under calibration. The Event-based Kalman Filtering Project [BLA+99] [BLN+98] used an onboard camera and odometric sensors. The camera must be calibrated and multiple coded targets must be specially placed and located throughout the robot calibration space.

The UMBark method uses robot odometric sensors, a high accuracy encoder trailer and an absolute acoustic positioning system. The statistical on-line calibration proposed method requires accurate acoustic and laser positioning sensors combined with an accurate map of the operating space.

Table 2-1 summarizes the related work in mobile robot calibration. A calibration accuracy of approximately one percent is the benchmark of performance. Thus, a goal of the proposed calibration process in this thesis is to achieve better than one percent calibration accuracy.

Table 2-1 Summary of Related Work in Mobile Robot Calibration

Reference	Robot(s)	Measurement	Estimator	Accuracy
[BEF96a]	Differential WMR 4-DOF WMR Tracked	Odometry	MSE	<1%
[BLA+99]	Differential WMR	Odometry Photogrammetry	EKF DD1 DD2	1.5%
[RT99]	4-wheel synchro	Odometry	MLE	1.1%

2.3 Proposed Mobile Robot Calibration Methodology

The proposal for a calibration methodology requires specification of measurement and parameter estimation techniques. In defining the proposed calibration system, the following goals are considered:

1. Calibration should be feasible for robots with limited computation capability.

2. Calibration should be feasible regardless of the robot sensor suite.

3. Calibration should feasible for any wheeled mobile robot configuration.

4. Calibration should be simple and require little special equipment.

These considerations led to the choice of photogrammetry as a viable measurement scheme. Low cost CCD cameras are available for operation by desktop personal computers. Such cameras are easily calibrated and offer reasonable performance for costs typically less than $100.

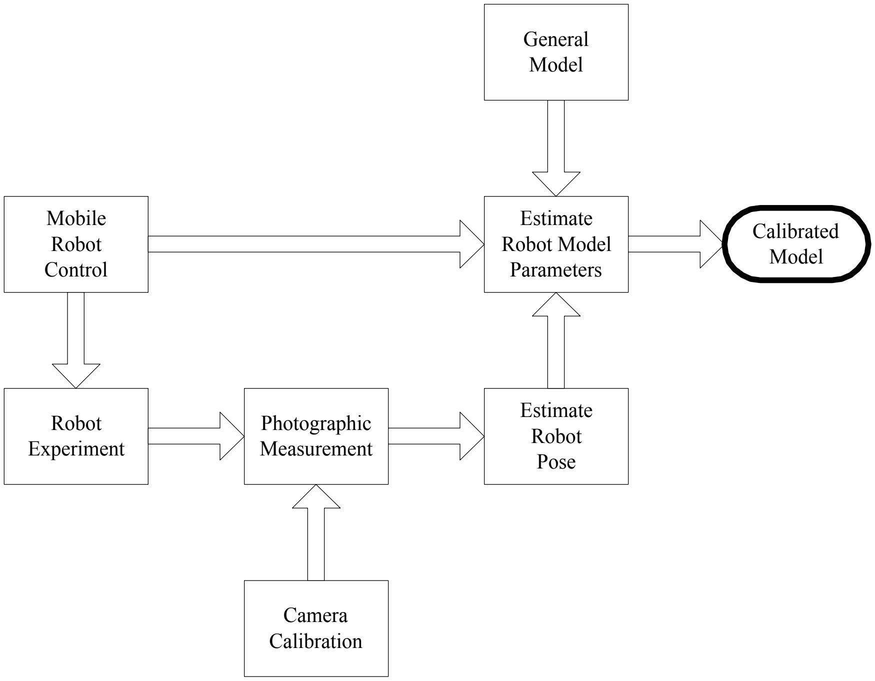


Figure 2-2 Non-intrusive Photogrammetric Mobile Robot Calibration

1. Derive the robot kinematic model and identify the parameters to be calibrated.

2. Calibrate the camera system

3. Conduct a calibration experiment with a known control vector

4. Record experiment with the calibrated camera

5. Estimate robot pose versus time

6. Estimate robot model parameters from the pose time series.

The calibration method considered is to apply a derivative-free form of the extended Kalman filter to the time varying robot kinematic data. The calibration process is illustrated in Figure 2-2.

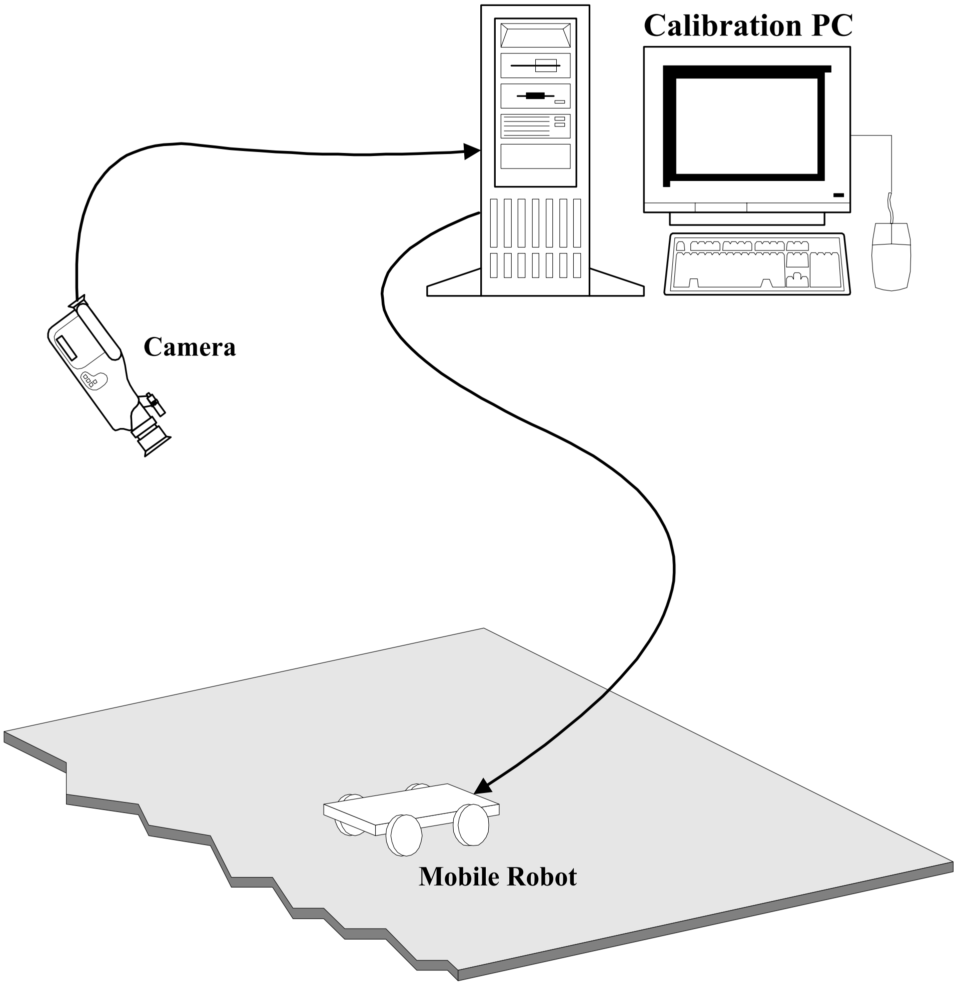


Figure 2-3 Non-Intrusive Photogrammetric Calibration Apparatus

Fundamentally, this is a system identification problem that involves selection of a model architecture and identification of model parameters based on observation. This thesis will limit discussion of the problem to system parameter identification through state estimation given an a priori model structure. The more general case of system identification will be considered in future work.

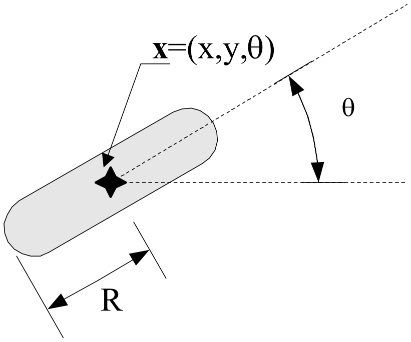
Thus the problem at hand can be stated as: given a dynamic system compute the parameters that completely characterize the system based on state estimation from measurement data.

3. Wheeled Mobile Robot Kinematic Models

This chapter presents a general formulation for determining the forward kinematics of a wheeled mobile robot (WMR). Two methods to determine the kinematic model from a robot design are presented. The first method is an extension of manipulator kinematic modeling to wheeled robots by Muir and Newman [MN86]. The second method, labeled instantaneous center of rotation (ICR) is a formalism of a common approach found throughout literature. The forward kinematic models of differential and Ackerman mobile robot configurations are developed using the ICR method. Finally, the models are extended to augmented state-space form for use with the parameter estimation algorithm described in Chapter 6.

3.1 Kinematic Model Development

A kinematic model is a system of equations that relates the change in kinematic state of a mechanical system to actuator inputs. The system of equations is derived from the geometry of the mechanical system and Newton's laws. In the case of wheeled mobile robots, the control inputs are wheel angular velocity and wheel steering angle. The kinematic state is the 6 degrees of spatial freedom given by position (x, y, z) and orientation (roll, pitch, yaw).



(3-1)

If the robot is restricted to have all wheels contact a flat level plane the z; roll and pitch components of state are constant and may be ignored. This thesis expresses a WMR kinematic model is expressed in state space form where **x** denotes kinematic state, *u* is control inputs and *f()* is a function, most often non-linear, relating state change to current state and the control inputs. In two dimensions, the vector in equation (3-2) gives the kinematic state of a robot

$$\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T \qquad (3\text{-}2)$$

(x, y) are the Cartesian position of the robot with respect to the world reference frame. And θ is robot orientation, which is the positive counter clockwise angle from x-axis.

Two general categories classify the approaches to mobile robot kinematics found in literature. The first approach, proposed by Muir and Newman [MN86], formally derives the kinematic equation of motion by modeling the mobile robot as a multiple closed link chain system. The second, and more common approach [BEF96a] [DJ00], assumes that the all points on the robot move about an instantaneous center of rotation (ICR). The position of the center of rotation relative to the robot is a function of the wheel speeds and turning angles. Each of these approaches to modeling WMR kinematics is discussed in more detail below.

3.1.1 Kinematic Equations of Motion by Muir and Newman

Muir and Newman [MN86] developed a comprehensive process to determine the kinematic equations of motion for wheeled mobile robots. Their formulation was based on manipulator kinematic principles. Muir and Newman [MN86] define a wheeled mobile robot as:

> *"A robot capable of locomotion on a surface solely through the actuation of wheel assemblies mounted on the robot and in contact with the surface. A wheel assembly is a device, which provides or allows relative motion between its mount and a surface on which it is intended to have a single point of rolling contact."*

A wheeled mobile robot maintains one more wheels in contact with the world plane. The kinematic chain from the robot to the world plane forms a closed kinematic chain. Thus a wheeled mobile robot consisting of multiple wheels in contact with a surface can be modeled a multiple closed-link chains. In this respect wheeled robot modeling differs from manipulator modeling in the following ways:

1. WMRs consist of multiple closed link chains. Each wheel forms a closed-link chain. Conversely, manipulators generally consist of a single open-link chain that is closed only when the end-tool is contact with a stationary object.
2. The contact between a wheel and the floor is a higher-order pair, whereas stationary manipulators contain only lower-order pair joints.
3. A manipulator may actuate all degrees of freedom (DOF) of each joint. Most WMRs actuate a subset of the DOF of each wheel.
4. All DOF of each joint on a manipulator have both position and velocity sensors. Only a subset of the DOF of a WMR wheel have position and velocity

A right-handed homogeneous co-ordinate system is used as shown in Figure 3-1 below. The Denavit-Hartenburg convention commonly used for manipulator co-ordinate frame assignment, gives ambiguous transformations since joint ordering is not obvious. Muir and Newman overcome this problem by using the Sheth-Uicker [SU71] convention for co-ordinate frame assignment. Each robot wheel is modeled as a planar pair at the wheel/surface contact point.

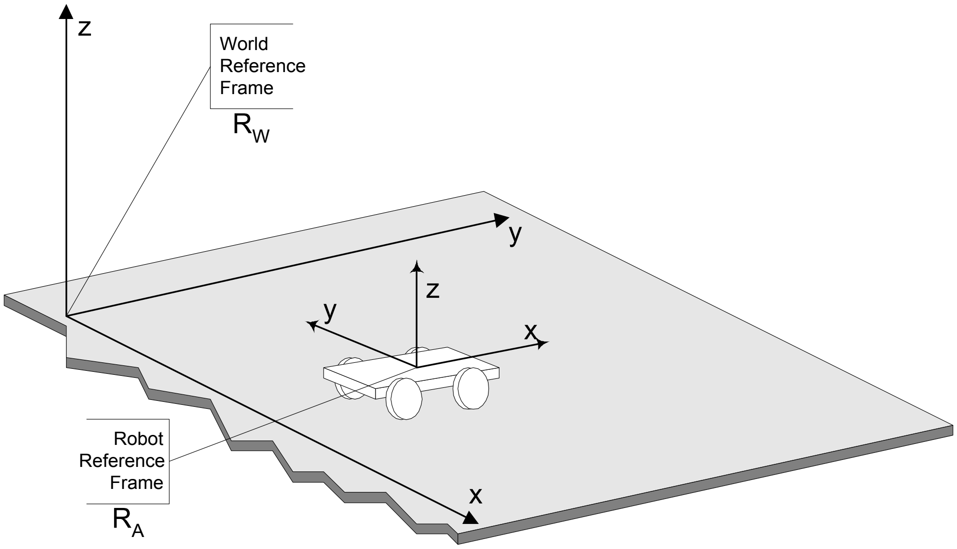


Figure 3-1 Homogeneous Co-ordinate System Convention

Figure 3-2 illustrates the co-ordinate frame conventions used for a generic mobile robot. The robot is given a body reference frame. This frame is usually at the geometric center of the robot. Each wheel is also given a frame.

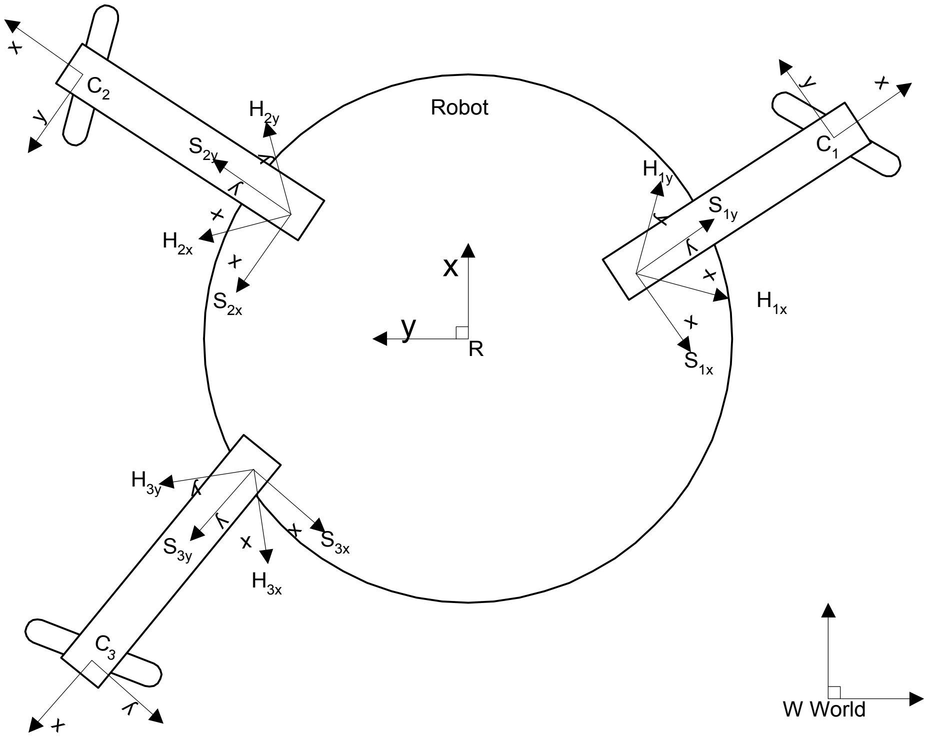


Figure 3-2 WMR Co-ordinate Conventions

The definitions of the robot frame conventions used are listed as follows:

W World: Arbitrary stationary reference co-ordinate system with the z-axis orthogonal to the surface of travel.

R Robot: Co-ordinate frame that moves with the robot body, with the z-axis orthogonal to the surface of travel.

H*ᵢ* Hip: $(i \in I)$ Co-ordinate frame which moves with the tractor body, with the z-axis coincident with the axis of the steering joint i if there is one; coincident with the contact point co-ordinate frame C_i if there is no steering joint.

S*ᵢ* Hip: $(i \in I)$ Co-ordinate frame which moves with the steering link *i*, with the z-axis coincident with the z-axis of tractor body, with the z-axis coincident with the z-axis of H_i, and the origin coincident with the origin of H_i.

C$_i$ Hip: $(i \in I)$ Co-ordinate frame which moves with the steering link *i,* with the origin at the point-of-contact between the wheel and the surface, the y-axis is parallel to the wheel and the x-y plane is tangent to the surface.

$\bar{R}$ **Robot**: Instantaneously coincident co-ordinate frame with frame R and stationary relative to the frame W.

$\bar{\mathbf{C}}_i$ **Hip**: $(i \in I)$ Instantaneously coincident co-ordinate frame with frame C$_i$ and stationary relative to the frame W.

In general, any frame R may be related to another frame N by a homogeneous co-ordinate transform $^{R}T_{N}$. **Figure 3-3** is a typical transform graph that illustrates the transform relationships between various points on the robot co-ordinate system and an arbitrary world reference frame.

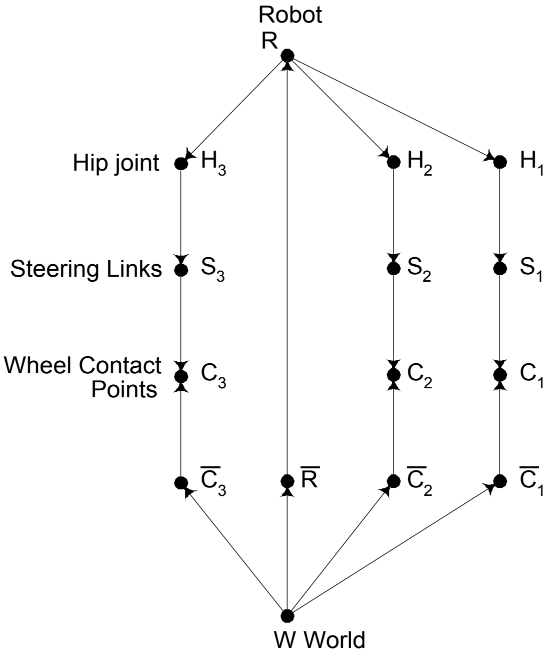


Figure 3-3 Transform Graph for a 3-Wheeled Mobile Robot

The transform between any two co-ordinate frames is given by the multiplication of the intermediate frames. The transforms between the robot frame and the world can be applied through any of the wheel contacts. The resultant transforms are:

$$\begin{aligned} {}^{W}T_{R} &= {}^{R}T_{H_1}\,{}^{H_1}T_{S_1}\,{}^{S_1}T_{C_1}\,{}^{C_1}T_{\bar{C}_1}\,{}^{\bar{C}_1}T_{W} \\ &= {}^{R}T_{H_2}\,{}^{H_2}T_{S_2}\,{}^{S_2}T_{C_2}\,{}^{C_2}T_{\bar{C}_2}\,{}^{\bar{C}_2}T_{W} \\ &= {}^{R}T_{H_3}\,{}^{H_3}T_{S_3}\,{}^{S_3}T_{C_3}\,{}^{C_3}T_{\bar{C}_3}\,{}^{\bar{C}_3}T_{W} \end{aligned} \tag{3-3}$$

The development of the kinematic model is based upon the following assumptions [Muir86]:

- The world consists of a smooth flat plane (z=0). Also known as a Ground Plane Constraint (GPC).
- The robot body is a rigid body.
- The steering axes are normal to the ground.
- No translational slip occurs between a wheel and the ground.
- The rotational friction at the point of contact between a wheel and the ground is small enough for the wheel to turn about a vertical axis through that point.

Note that Muir and Newman assume all wheels are identical in all respects including their radii. This assumption is not suitable practice since WMR wheels are not identical. The following general process [MN86] will be applied to develop the kinematic model of the vehicle.

1. Geometrically characterize the system (sketch the vehicle).
2. Assign a co-ordinate system and reference frame(s) to vehicle.
3. Develop the transform graph of the vehicle.
4. Develop the co-ordinate transform matrices.
5. Formulate position equations of motion.
6. Formulate velocity equations of motion.
7. Formulate acceleration equations of motion.

3.1.2 ICR Based Kinematic Models

The instantaneous center of rotation (ICR) method is a geometrically intuitive approach to develop a WMR kinematic model. The ICR method of WMR kinematic modeling assumes that the rolling axes of all wheels intersect at single point. If this assumption is true, each wheel may roll without slippage, and the robot will move along a circular arc whose center is the ICR.

Figure 3-4 illustrates two three-wheeled mobile robot configurations. In the left configuration, the rolling axes of each wheel intersect at a single point. This permits rolling motion without wheel slippage. In the right configuration, the rolling axes do not intersect at a single point. Rolling motion is not possible and the robot can only move subject to wheel slippage.

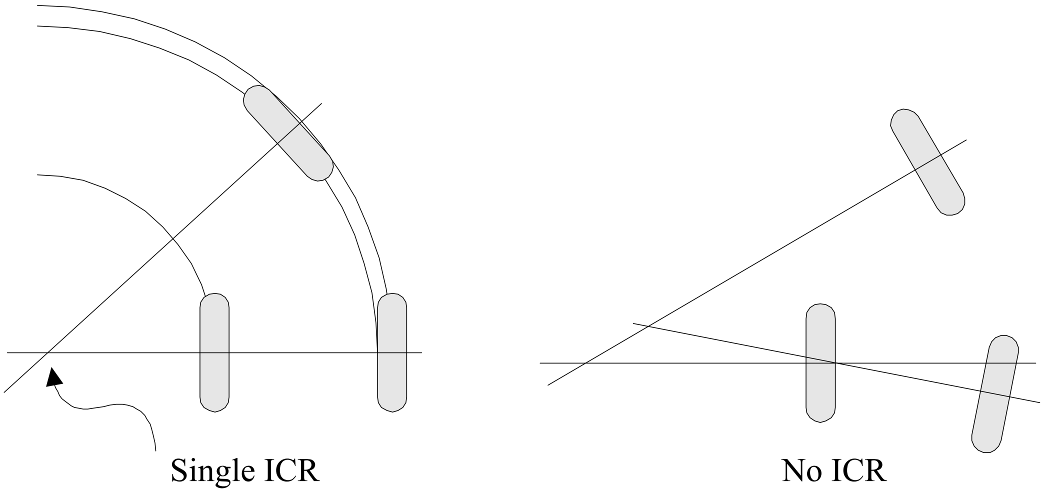


Figure 3-4 WMR Configurations with and without ICR

In the case of a kinematic configuration with an ICR, robot will move along a circular arc about the ICR. Each wheel traverses an equivalent arc angle over a fixed time duration. The ICR method determines a WMR kinematic model according to the following process:

1. Draw the robot kinematic configuration.

2. Draw lines through the rolling axes of each wheel and identify the ICR.

3. Derive the equation relating arc length path to wheel rotation for each wheel.

4. Derive the ICR as a function of robot parameters.

5. Combine the equations from step 3 and step 4 into a system of equations from robot geometry.

6. Solve the system of equations to yield the WMR kinematic model.

3.2 Unicycle WMR Kinematic Model

The unicycle is the simplest form of wheeled mobile robot. Although the unicycle is not practical as a WMR configuration, it provides the basic model for use in more complex wheeled configurations.

The basis of the wheel kinematic model is a planar pair. A general planar pair has three degrees of freedom, two of which are translational, and one of which is rotational, as shown in Figure 3-5(a). Thus, a planar pair may change x, y and θ independently without constraint.

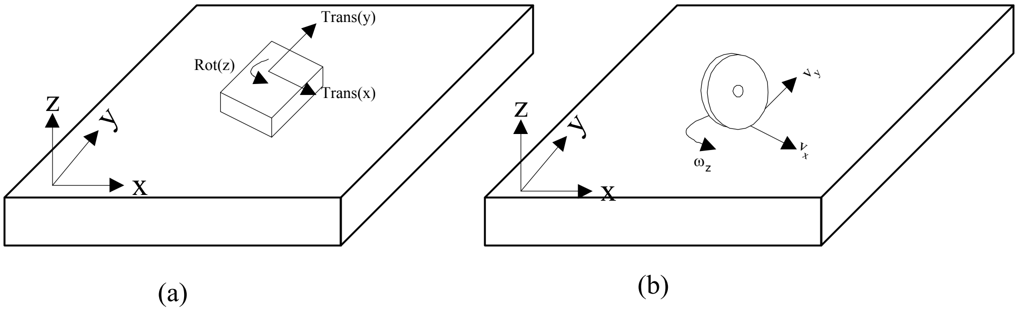


Figure 3-5 Conventional Wheel Modeled as a Planar Pair

A conventional wheel has two degrees of freedom. Rotation of the wheel is permitted normal to the surface. Translation is allowed along the axis of rotation. The wheel model is equivalent to the planar model with a lateral motion constraint. The amount of translation is proportional to the amount of wheel rotation assuming no wheel slippage.

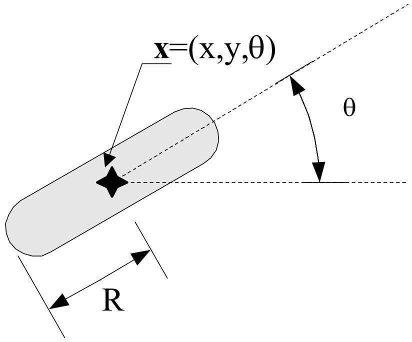


Figure 3-6 Unicycle WMR Configuration

Let $\mathbf{u} = \left[\omega \quad \phi\right]^T$ be the control vector of the unicycle where ω denotes the wheel angular rate of rotation and ϕ denotes the steering angle position. First, consider the two extreme case of robot control. If $\mathbf{u} = \left[0 \quad \phi\right]^T$, then ICR is the point where the wheel contacts the ground plane. The unicycle will rotate in place with a change in kinematic state given by $d\theta/dt = d\phi/dt = \omega_\phi$. If $\mathbf{u} = \left[w \quad 0\right]^T$, then ICR is a point an infinite distance in the y-direction robot reference frame. The robot will move along with constant heading with a linear displacement given by

$$\begin{aligned} dx &= \omega R \cos\theta dt \\ dy &= \omega R \sin\theta dt \end{aligned} \tag{3-4}$$

Figure 3-7 illustrates the non-trivial case for non-zero wheel speed and steering rate. The ICR position relative to the robot determines the arc the robot will follow for fixed control values. The arc length is $dS = \omega dt$. The angle subtended by the arc dS is $\alpha = \Delta\phi = \omega_\phi dt$ in units of radians. Only one arc length equation is required since the unicycle has only one wheel. It follows that the ICR radius is $R_{ICR} = dS/\alpha$.

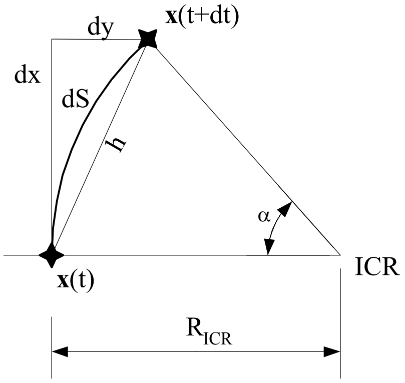


Figure 3-7 Unicycle ICR Geometry

The kinematic model is found by solving the triangle [dx dy h] in Figure 3-7. Thus, the kinematic model of a unicycle is given by

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} R\cos\theta & 0 \\ R\sin\theta & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \omega \\ \omega_\phi \end{bmatrix} \tag{3-5}$$

where x and y denote the position of the unicycle with respect to the world frame. θ denotes the heading of the unicycle. ω is the wheel angular speed and ω_ϕ is the wheel steering angle rate.

3.3 Differential WMR Kinematic Model

The differential wheeled mobile robot is the simplest and most common [DJ00] kinematic configuration of mobile robots operated indoors. A differential WMR has two independent drive wheels and one or more castors. The castor wheels, which provide body stability, are not considered in the kinematic model. Figure 3-8 illustrates the differential WMR configuration.

The kinematic state of the robot is given by the vector $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ where (x, y) are the Cartesian position of the robot and θ is its heading. The variable D denotes the distance between drive wheels. R_L and R_R denote the left and right wheel radii respectively. The control parameters are the left and right wheel angular rotations speeds ω_L and ω_R respectively.

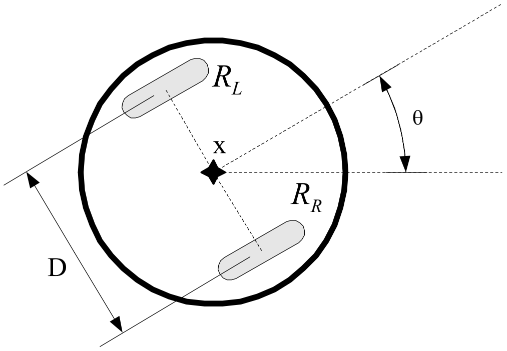


Figure 3-8 Differential WMR Configuration

Figure 3-9 shows a simple hypothetical experiment where an arc is traced by the left and right wheels over some time interval dt.

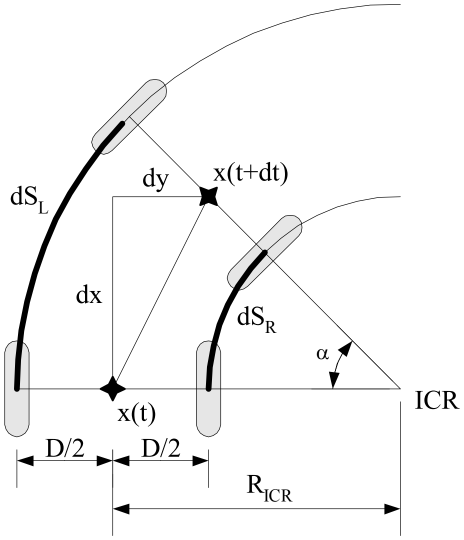


Figure 3-9 Differential WMR Motion

As discussed above the wheel arcs are circular and centered about some instantaneous center of rotation (ICR). The ICR is located at distance R_{ICR} from the robot reference frame. The arcs traced by the wheel over some time interval dt subtend an equivalent angle α. Computing the arc length that each wheel traces gives

$$
\begin{aligned}
dS_L &= \alpha\left(\frac{D}{2}+R\right) \Rightarrow \alpha = \frac{dS_L}{\left(\frac{D}{2}+R\right)} \\
dS_R &= \alpha\left(\frac{D}{2}-R\right) \Rightarrow \alpha = \frac{dS_R}{\left(\frac{D}{2}-R\right)}
\end{aligned}
\tag{3-6}
$$

The equivalency of left and right wheel arc angles in (3-6), gives the relation between wheel path arcs, the wheel base and the ICR.

$$
\frac{dS_L}{\left(R_{ICR}+\frac{D}{2}\right)} = \frac{dS_R}{\left(R_{ICR}-\frac{D}{2}\right)}
\tag{3-7}
$$

The products of wheel radii and the wheel rotation rates give the wheel path arc lengths.

$$
\begin{aligned}
dS_L &= dt\omega_L R_L \\
dS_R &= dt\omega_R R_R
\end{aligned}
\tag{3-8}
$$

It follows from substitution of (3-8) into (3-7) that the distance form the robot origin to the ICR is given by

$$
R_{ICR} = \frac{\frac{1}{2}D\left(\omega_R R_R + \omega_L R_L\right)}{\left(\omega_R R_R - \omega_L R_L\right)}
\tag{3-9}
$$

Solving for the equations (3-7) and (3-9) for the differential change in state give the kinematic model for the differential WMR as

$$\frac{d\mathbf{x}}{dt} = \frac{d}{dt}\begin{bmatrix} x & y & \theta \end{bmatrix}^T$$

$$= \begin{pmatrix} \frac{1}{2}R_L \cos\theta & \frac{1}{2}R_R \cos\theta \\ \frac{1}{2}R_L \sin\theta & \frac{1}{2}R_R \sin\theta \\ \dfrac{-R_L}{D} & \dfrac{-R_R}{D} \end{pmatrix} \begin{pmatrix} \omega_L \\ \omega_R \end{pmatrix} \tag{3-10}$$

3.4 Ackerman Steered WMR Model

The Ackerman WMR configuration is equivalent to most 4-wheeled automobiles and AGVs in common use. The Ackerman configuration is named after the inventor of Ackerman steering [Ever95], which is a mechanical steering linkage mechanism that prevents wheel slippage of the front steering wheels during turns. Figure 3-10 illustrates the Ackerman WMR kinematic configuration. The robot has four wheels. The front wheels are steered and not driven while the rear wheels are driven and un-steered. The wheel radii are denoted R_{LF} where the subscript denotes the wheel (e.g. LF = left front and RB = right back). The angular speed of each wheel is denoted ω_{LF} with identical subscript notation as for the wheel radii. The steering angles of the front left and right wheels are denoted ϕ_L and ϕ_R respectively. The separation between wheels along the rotation axis is D. The distance between wheel axes is L.

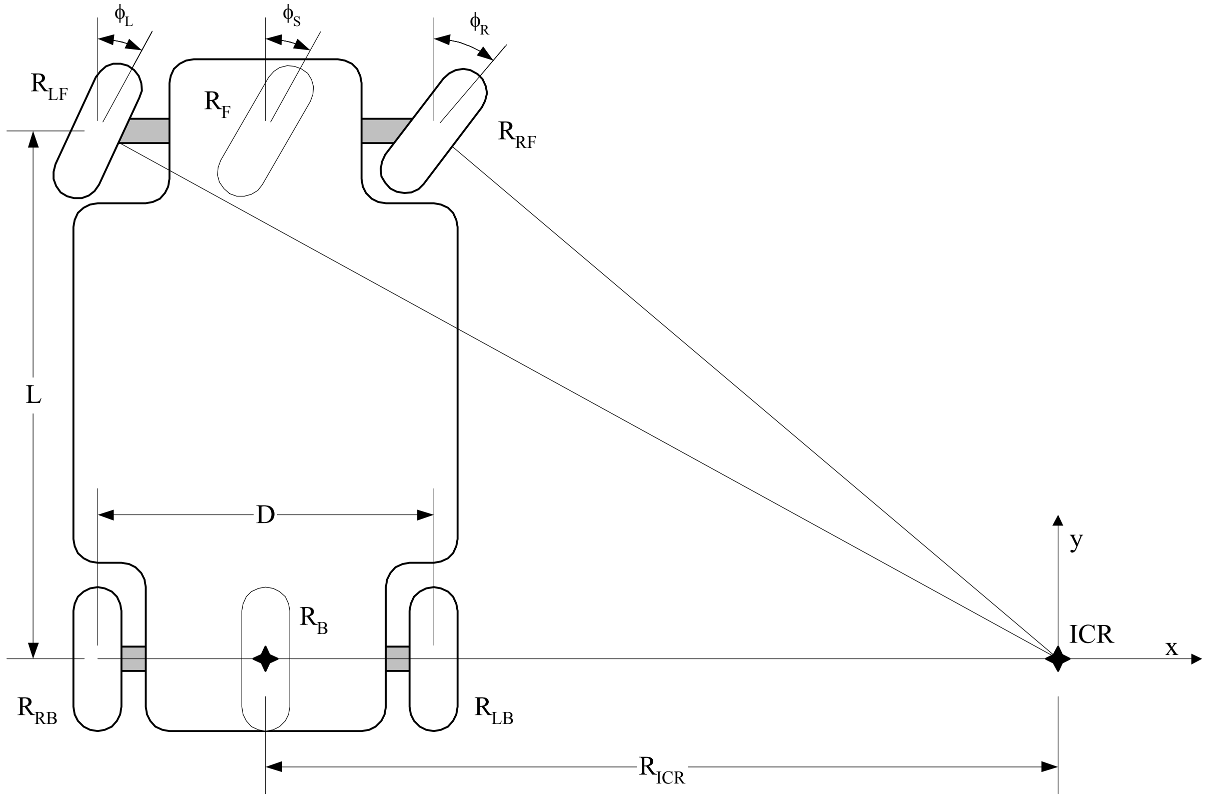


Figure 3-10 Ackerman Steered WMR Configuration

The Ackerman configuration constrains the front steering wheel angles such that there is no wheel slippage. This constraint ensures that for any non-zero steering angles all points of the vehicle will rotate about a common instantaneous center of rotation (ICR). Rotation about an ICR avoids wheel slippage thereby leading to unambiguous control of motion. The steering wheel angles are constrained by the relation in equation (3-11) to ensure they have a common ICR with the rear wheels.

$$\pm\cot\phi_R \mp \cot\phi_L = \frac{D}{L} \tag{3-11}$$

Equation (3-11) is also known as the Ackerman constraint. The minus sign in equation (3-11) is applied to the inner wheel for a right or left turn. The Ackerman constraint allows the kinematic model to be simplified by modeling the left and right wheels by a single virtual wheel with radius R_F and steering angle ϕ_S. The virtual wheel steering angle can be expressed in terms of the left and right wheel steering angles as

$$\begin{aligned}\cot\theta_S &= \cot\phi_R \pm \frac{D}{2L} \\ &= \cot\phi_L \mp \frac{D}{2L}\end{aligned} \tag{3-12}$$

The virtual wheel model equates the four-wheeled Ackerman configuration to the tricycle configuration. Most Ackerman vehicles have a single motor that drives the rear wheels via a differentially coupled axle. For such configurations, a single virtual wheel of radius R_B and angular speed ω can model the two rear wheels. Using the virtual rear drive wheel makes the Ackerman configuration kinematically equivalent to a bicycle configuration with one steering wheel and one drive wheel. The control variables for the vehicle are then the front wheel steering angle ϕ_S and the rear wheel angular speed ω.

The following text applies the ICR method to determine the kinematic model of the Ackerman WMR configuration. Consider an infinitesimal turn to right as shown in Figure 3-11 where the initial state of the robot is $\mathbf{x} = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$ and the actuated variables are ϕ_S and ω. The robot frame origin is the rear wheel contact point with the ground plane. The result is a differential change in state over an infinitesimal time dt of $d\mathbf{x} = \begin{bmatrix} dx & dy & d\theta \end{bmatrix}^T$. The distance from the rear wheel to the ICR is given by

$$R_{ICR} = L\cot\phi_S \tag{3-13}$$

and the distance from the steering wheel contact point to ICR is

$$R_S = \frac{L}{\sin\phi_S} \tag{3-14}$$

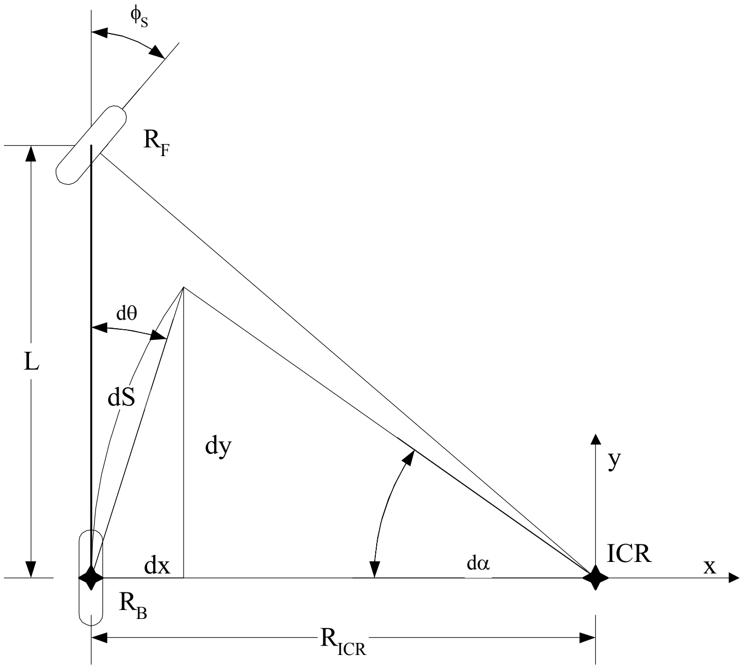


Figure 3-11 Ackerman Turn Geometry

Each wheel contact point will trace an arc dS, subtended by the angle dα, whose center is the ICR. The arc length traced by the rear wheel is given by

$$dS = R_{ICR} d\theta = R_B \omega dt \tag{3-15}$$

where R_B is the radius of the virtual rear wheel. Thus, the differential change in vehicle orientation is given by

$$d\theta = \frac{dS}{L}\tan\phi_S \tag{3-16}$$

Solving the triangle [dx dy] for dx and dy gives the differential change in position as

$$\begin{aligned} dx &= dS\cos\phi_s\cos\theta \\ dy &= dS\cos\phi_s\sin\theta \end{aligned}\,. \tag{3-17}$$

The instantaneous wheel velocity is given by

$$\frac{dS}{dt} = \omega R \tag{3-18}$$

Then the differential change in state may be expressed in terms of wheel angular velocity as

$$\begin{aligned} dx &= dt\omega R_B\cos\phi_s\cos\theta \\ dy &= dt\omega R_B\cos\phi_s\sin\theta\;. \\ d\theta &= dt\frac{\omega R_B}{L}\tan\phi_s \end{aligned} \tag{3-19}$$

Finally, the Ackerman WMR kinematic model is given by expressing equation (3-19) in state space form.

$$\begin{aligned} \dot{\mathbf{x}} &= \frac{d}{dt}\begin{bmatrix} x & y & \theta \end{bmatrix}^T \\ &= \begin{bmatrix} R_B\cos\theta\cos\phi_s & 0 \\ R_B\sin\theta\cos\phi_s & 0 \\ \frac{R_B}{L}\tan\phi_s & 0 \end{bmatrix}\begin{bmatrix} \omega \\ \phi \end{bmatrix} \end{aligned} \tag{3-20}$$

3.5 Augmented State Vector

Equation (3-10) and equation (3-20) are the state-space form of the kinematic models for differential and Ackerman wheeled mobile robots. These equations can be used to simulate and control mobile robots. The accuracy or proficiency of the simulation or control depends in part upon the accuracy of the model. An error in the kinematic parameters will result in model errors. Thus, a means is required to determine the kinematic parameters.

Chapter 5 of this thesis proposes a non-linear state estimation scheme to calibrate the kinematic model parameters from photogrammetric state measurements. The state-space models developed are adequate for state estimation. However, the proposed parameter estimation scheme will estimate process state and parameters concurrently. In order to do this the augmented state vector is introduced [Eykhoff74]. Given a dynamic system described in state form by

$$\dot{\mathbf{x}} = f\left(\mathbf{x}, \mathbf{a}, \mathbf{u}\right) \tag{3-21}$$

where $\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T$ is the n-dimensional state vector, $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_m \end{bmatrix}^T$ is an m-dimensional parameter vector and $\mathbf{u} = \begin{bmatrix} u_1 & u_2 & \cdots & u_k \end{bmatrix}^T$ is a k-dimensional control vector. The kinematic parameters to be estimated are constant in time. Therefore, the state-space formulation of the parameters is

$$\dot{\mathbf{a}} = 0 \tag{3-22}$$

Define the augmented state vector as

$$\mathbf{x}_a = \begin{bmatrix} \mathbf{x}^T & \mathbf{a}^T \end{bmatrix}^T \tag{3-23}$$

Equation (3-21) and equation (3-22) are combined into a system of equations for the augmented state-space form.

$$\begin{aligned} \dot{\mathbf{x}}_a &= \begin{bmatrix} \dot{\mathbf{x}} \\ \dot{\mathbf{a}} \end{bmatrix} \\ &= \begin{bmatrix} f\left(\mathbf{x}, \mathbf{a}, \mathbf{u}\right) \\ 0 \end{bmatrix} \end{aligned} \tag{3-24}$$

It is important to note that the augmented state-space form is non-linear even if the process is linear. This restricts the use of linear estimation techniques to the state estimation problem.

The kinematic parameters of the differential WMR are wheel radii and wheel separation. Thus the, augmented state-space form of the differential WMR model (3-10) is

$$\begin{aligned}\dot{\mathbf{x}} &= \frac{d}{dt}\begin{bmatrix} x & y & \theta & R_L & R_R & D \end{bmatrix}^T \\ &= \begin{pmatrix} \frac{1}{2}R_L\cos\theta & \frac{1}{2}R_R\cos\theta \\ \frac{1}{2}R_L\sin\theta & \frac{1}{2}R_R\sin\theta \\ \frac{-R_L}{D} & \frac{-R_R}{D} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \omega_L \\ \omega_R \end{pmatrix}\end{aligned} \tag{3-25}$$

The kinematic parameters of the Ackerman WMR are virtual rear wheel radius and the wheel axis separation. Thus the, augmented state-space form of the Ackerman WMR model (3-20) is

$$\dot{\mathbf{x}} = \frac{d}{dt}\begin{bmatrix} x & y & \theta & R_B & L \end{bmatrix}^T = \begin{bmatrix} R_B\cos\theta\cos\phi_s & 0 \\ R_B\sin\theta\cos\phi_s & 0 \\ \frac{R_B}{L}\tan\phi_s & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \omega \\ \phi_S \end{bmatrix} \tag{3-26}$$

Equation (3-25) and equation (3-26) are the augmented kinematic models that will be used for the calibration scheme proposed in this thesis.

4. Computer Vision and Photogrammetry

This chapter introduces photogrammetry as a non-intrusive means to measure mobile robot kinematic state, also referred to as pose. As will be shown, photogrammetry has several advantages versus alternatives for the robot calibration application. Photogrammetry provides precise measurement of robot kinematic state without robot disassembly and expensive physical measurements. Photogrammetry provides a passive measurement scheme that requires minimal instrumentation and modest computing capability.

The proposed photogrammetric process uses a non-metric[1] camera to reduce cost. There are a vast number of low cost non-metric cameras available in the commercial marketplace. A measurement scheme for a low grade digital charge coupled device (CCD) cameras is proposed below. The proposed scheme is advantageous in that reasonable accuracy is achieved and no operator intervention is required throughout the process. However, accurate calibration of the camera is required for precise photogrammetry.

This chapter is organized as follows. First the field of photogrammetry is introduced. The photogrammetric error metrics are defined. An accurate model of the imaging process is required to support non-metric photogrammetry. The perspective model for a CCD digital camera is derived. A modified version of Tsai's camera calibration algorithm is described as a means to determine the imaging model. First, Tsai's original algorithm [Tsai87] is described. Then a modified form of the Tsai algorithm is described [HS97]. Finally, a fully projective formulation of Lowe's pose recovery algorithm is described [ACB98]. This fully projective form of Lowe's algorithm is the pose recovery algorithm selected for the robot calibration methodology proposed in this thesis.

[1] A metric camera is a photographic device specifically designed and fabricated for photogrammetric application. In application, a metric camera is precisely controlled so that all intrinsic and extrinsic properties are known. Generally a metric camera has fixed focus and focal length, known lens distortion and known principle point position relative to the image plane. Most commercially available metric cameras are expensive and require frequent calibration.

4.1 Photogrammetry

Photogrammetry is the measurement of the spatial dimension, position and orientation of three-dimensional objects from two-dimensional images [Kara79]. Photogrammetry provides a non-intrusive metrology that has received extensive use in aerial photography, cartography and remote sensing [Yuan89]. More recently, photogrammetry has been successfully applied in industrial inspection systems [JHG00].

Figure 4-1 illustrates the photogrammetric process. A camera captures and image of an object. The process of creating the object image is termed as the imaging process. Pose recovery is the reverse of the imaging process where object position and orientation (pose) are estimated from the object image.

Let **M** denote a model that characterises a physical object as a set of n features

$$\mathbf{M} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \dots & \mathbf{p}_n \end{bmatrix}, \qquad \mathbf{p}_i = \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T \in \mathbb{R}^3 \tag{4-1}$$

where each feature is given as a point in 3-dimensional world reference frame $\mathbf{R}_\mathrm{W}$. The imaging process projects the light reflected from object onto an imaging plane denoted by $\mathbf{R}_\mathrm{I}$. The projected light is detected and sampled by sensors coincident with the imaging plane to form an image **I**. The image contains 2-dimensional features corresponding to the 3-dimensional object features. The pose recovery process detects the corresponding features in the captured image, matches them to the model features and calculates their estimated position in the original reference frame $\mathbf{R}_\mathrm{W}$. The pose estimation algorithm requires an accurate model of the imaging process to correspond image point co-ordinates to world co-ordinates of the model. Pose is calculated by matching the estimated feature positions to the object model **M**.

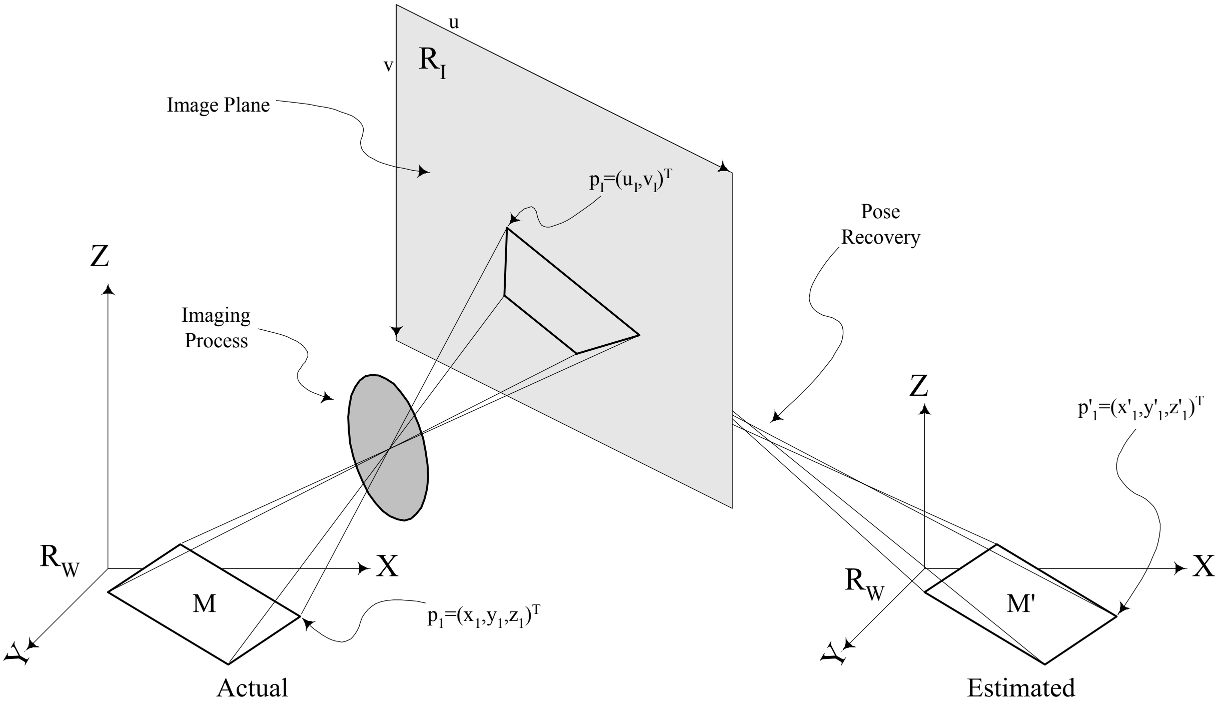


Figure 4-1 Photogrammetric Process

The goal in photogrammetry is to estimate object dimension and pose with minimal error. Define the distance error as the norm distance error (NDE) which is the norm of the difference between the true feature position and estimated feature position

$$
\begin{aligned}
NDE &= \left\| \Delta \mathbf{p}_i \right\| \\
&= \left\| \mathbf{p}_i^{'} - \mathbf{p}_i \right\| \\
&= \left\| \begin{bmatrix} x_i & y_i & z_i \end{bmatrix}^T - \begin{bmatrix} x_i^{'} & y_i^{'} & z_i^{'} \end{bmatrix}^T \right\|
\end{aligned}
\tag{4-2}
$$

Define the pose error as the norm pose error (NPE) which is the norm of difference between the true object pose and estimated object pose

$$
\begin{aligned}
NPE &= \left\|\Delta\mathbf{P}\right\| \\
&= \left\|pose_{actual} - pose_{estimated}\right\| \\
&= \left\|\mathbf{P}-\mathbf{P}'\right\| \\
&= \left\|\begin{bmatrix} x & y & z & \theta & \phi & \psi \end{bmatrix}^T - \begin{bmatrix} x' & y' & z' & \theta' & \phi' & \psi \end{bmatrix}^T\right\|
\end{aligned}
\tag{4-3}
$$

where object pose **P** is the six-tuple $\begin{bmatrix} x & y & z & \theta & \phi & \psi \end{bmatrix}^T \in \mathbb{R}^6$. ϕ, θ, ψ denote the Euler angles where θ is yaw, ϕ is pitch and ψ is roll. x, y and z denote the translation of the object origin with respect to the world reference frame along the X, Y and Z axes respectively.

4.2 Imaging Process Model

A low cost CCD camera is used as the photogrammetric sensor for robot calibration. The CCD camera is characterized as a non-linear electro-optical device that maps points in a 3-dimensional world space to a 2-dimensional image space. The simplest form of a camera model is the perspective transform which assumes that points, imaged in a 3-dimensional viewing volume, project through a camera focal point onto an imaging plane. The imaging plane is normal to the focal axis. Figure 4-2 illustrates this model, referred to as the pinhole camera model. The pinhole camera model is based on the principle of co-linearity, where each point in the world space is projected by a straight line through the projection center (focal point) into the image space (image plane).

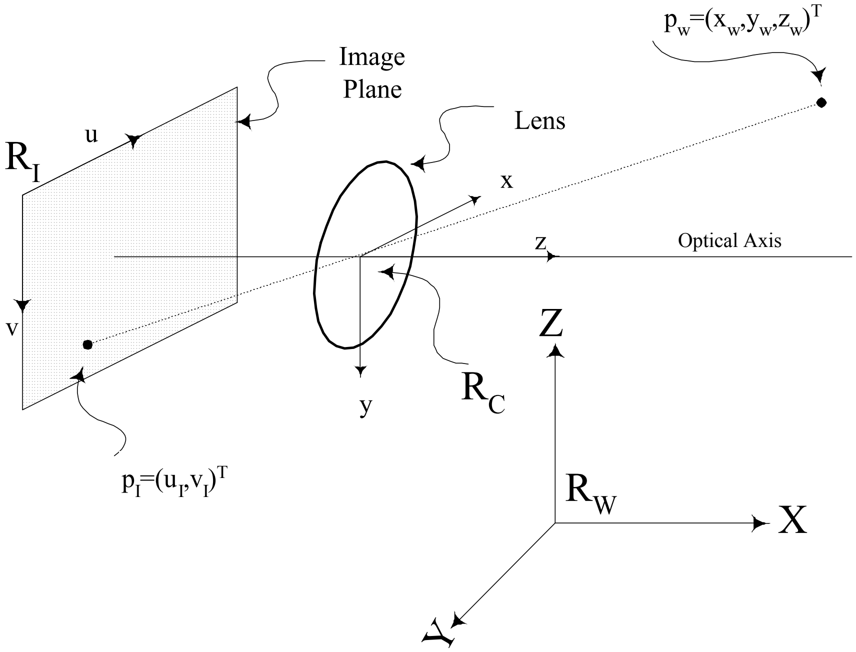


Figure 4-2 Pinhole Camera Model

The pinhole camera model is mathematically modeled as a homogeneous transformation by combining kinematic reference frame transformations with perspective geometry [GW93]. Let **A** denote the perspective image transform which maps a point in the world reference frame $\mathbf{R}_w \in \mathbb{R}^3$ to the corresponding points in the 2-dimensional image space $\mathbf{R}_I \in \mathbb{Z}^2$

$$\left\{ \mathbf{A} : \mathbf{R}_W \to \mathbf{R}_I \middle| \mathbf{R}_W \in \mathbb{R}^3, \mathbf{R}_I \in \mathbb{Z}^2 \right\} \tag{4-4}$$

where $\mathbb{R}^3$ denotes the 3-dimensional vector space of real numbers and $\mathbb{Z}^2$ denotes the 2-dimensional vector space of integers. The imaging transform **A** is decomposed into four transforms (**P**,**C**, **R** and **T**) which relate to measurable camera properties.

A point $\mathbf{p}_w$ in the world reference frame maps to a point $\mathbf{p}_I$ on the image plane by the camera imaging transform A

$$\begin{aligned}\mathbf{p}_I &= \mathbf{A}\mathbf{p}_w \\ &= \mathbf{PCRT}\mathbf{p}_w\end{aligned} \tag{4-5}$$

where **P** is the perspective transform given by

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \tag{4-6}$$

and λ is the camera lens effective focal length. **C** is the camera image plane displacement from the gimbal given by

$$C = \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-7}$$

R is the camera attitude given by

$$R = \begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta\cos\alpha & \cos\theta\cos\alpha & \sin\alpha & 0 \\ \sin\theta\sin\alpha & -\cos\theta\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-8}$$

where θ is the pan angle an α is the tilt angle. **T** is the position of the camera gimbal in the world frame given by

$$T = \begin{bmatrix} 1 & 0 & 0 & -X_o \\ 0 & 1 & 0 & -Y_o \\ 0 & 0 & 1 & -Z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4-9}$$

Thus a point world point $[X\ Y\ Z]^T$ is given in the image space homogeneous coordinates by

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = A \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{-1}{\lambda} & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -r_1 \\ 0 & 1 & 0 & -r_2 \\ 0 & 0 & 1 & -r_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times$$

$$\begin{bmatrix} \cos\theta & \sin\theta & 0 & 0 \\ -\sin\theta\cos\alpha & \cos\theta\cos\alpha & \sin\alpha & 0 \\ \sin\theta\sin\alpha & -\cos\theta\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -X_o \\ 0 & 1 & 0 & -Y_o \\ 0 & 0 & 1 & -Z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} (X-X_o)\cos\theta + (Y-Y_o)\sin\theta - r_1 \\ -(X-X_o)\sin\theta\cos\alpha + (Y-Y_o)\cos\theta\cos\alpha + (Z-Z_o)\sin\alpha - r_2 \\ (X-X_o)\sin\theta\sin\alpha - \cos\theta\sin\alpha(Y-Y_o) + (Z-Z_o)\cos\alpha - r_3 \\ -\frac{1}{\lambda}\left((X-X_o)\sin\theta\sin\alpha - \cos\theta\sin\alpha(Y-Y_o) + (Z-Z_o)\cos\alpha - r_3\right) + 1 \end{bmatrix} \tag{4-10}$$

And the Cartesian image co-ordinates of the point are given by

$$\begin{aligned} x &= \lambda \frac{(X-X_o)\cos\theta + (Y-Y_o)\sin\theta - r_1}{-\left((X-X_o)\sin\theta\sin\alpha - \cos\theta\sin\alpha(Y-Y_o) + (Z-Z_o)\cos\alpha - r_3\right) + \lambda} \\ y &= \lambda \frac{-(X-X_o)\sin\theta\cos\alpha + (Y-Y_o)\cos\theta\cos\alpha + (Z-Z_o)\sin\alpha - r_2}{-\left((X-X_o)\sin\theta\sin\alpha - \cos\theta\sin\alpha(Y-Y_o) + (Z-Z_o)\cos\alpha - r_3\right) + \lambda} \end{aligned} \tag{4-11}$$

Knowledge of the camera model allows computation of point 3-D position from 2-D image co-ordinates. Given an image point $\mathbf{p}_I$ the corresponding world point $\mathbf{p}_W$ can be computed by the inverse perspective transform

$$\begin{aligned} \mathbf{p}_I &= \mathbf{A}^{-1}\mathbf{p}_w \\ &= \mathbf{T}^{-1}\mathbf{R}^{-1}\mathbf{C}^{-1}\mathbf{P}^{-1}\mathbf{p}_w \end{aligned} \tag{4-12}$$

Note, however, that the solution is ambiguous since depth information was lost in the imaging process (4-11). To remove the depth ambiguity constraints must be introduced. For example, multiple images from different viewpoints may be captured. This is the approach taken by stereospy and active vision. Alternately, if the imaged object is constrained (e.g. z is known), then the ambiguity is also removed. Thus, a geometric world may be measured by computer vision with suitable constraints. Every visible point can be measured, and a geometric model can be constructed for pose recovery. The choice of monocular vision versus other configurations is primarily determined by a priori knowledge of the world. If no a priori information is given, then stereoscopic, active or structured vision is required [GW93].

4.3 Camera Calibration

The previous section introduced the imaging process as a geometric model given by the perspective transformation **A**. The camera model **A** must be known in advance of photogrammetric measurement of real objects. The determination of the camera (imaging) model **A** is referred to as camera calibration [GW93]. Inspection of equation (4-10) yields the following unknowns, which are directly related to measurable camera parameters:

$$
\begin{array}{rl}
\lambda = & \text{camera focal length} \\
X_o = & \text{camera gimble x world frame coordinate} \\
Y_o = & \text{camera gimble y world frame coordinate} \\
Z_o = & \text{camera gimble z world frame coordinate} \\
r_1 = & \text{image plane x gimbal frame coordinate} \\
r_2 = & \text{image plane y gimbal frame coordinate} \\
r_3 = & \text{image plane z gimbal frame coordinate} \\
\theta = & \text{camera pan} \\
\alpha = & \text{camera tilt}
\end{array}
\tag{4-13}
$$

The above camera parameters can be determined in either of two ways. The traditional approach is to physically measure all of the camera parameters to determine the camera model **A**. This method is laborious, lengthy and requires very accurate instrumentation. The alternative is to estimate the camera model by imaging known world points. This approach requires more computation. However, it has the advantage that no accurate instrumentation is required and the calibration time is only limited by computational speed.

To calibrate a camera, the camera images a set of N known points $P_{Wi}=[X_i, Y_i, Z_i]$ (I=1..N) in the world frame $\mathbf{R}_W$. The corresponding image points $\mathbf{P}_I$ in the image frame $\mathbf{R}_I$. are determined by measuring the resulting image (i.e. estimate the pixels corresponding to each calibration point). This results in a system of linear equations that can be solved by means of a linear estimation technique such as the least-squares estimator. Theoretically, only six known world points are required to solve calibration problem [GW93]. Typically, at least sixteen points are used with linear regression techniques to obtain results that are more robust.

4.3.1 Direct Linear Transformation

The pinhole camera model presented above is a simple linear model that serves as the foundation of computer vision. However, it has been shown that the pinhole model is insufficient for most computer vision applications [Kara79]. The pinhole model does not account for non-linear effects due to manufacturing tolerances, lens imperfections, assembly errors and intensity and spatial quantization due to image sampling.

A more accurate camera calibration model is the Direct Linear Transformation (DLT). The DLT was originally developed by Abdel-Aziz and Karara [AK71] in the field of non-metric photogrammetry. It was later modified based on extensive experimentation [KA74]. The concept of the DLT is based on transformation directly from comparator co-ordinates into object-space co-ordinates thus eliminating the requirement for metric camera fiducial marks. The reader is referred to [Kara79], [AK71] and [KA74] for the original DLT formulation. The method is based on the following pairs of equations

$$\begin{aligned}
&x+\left(x-x_o\right)\left(K_1r^2+K_2r^4+K_3r^6+\ldots\right)\\
&+\left(r^2+2\left[x-x_o\right]^2\right)P_1+2\left(y-y_o\right)\left(x-x_o\right)P_2=\frac{L_1X+L_2Y+L_3Z+L_4}{L_9X+L_{10}Y+L_{11}Z+1}\\
&y+\left(y-y_o\right)\left(K_1r^2+K_2r^4+K_3r^6+\ldots\right)\\
&+2\left(x-x_o\right)\left(y-y_o\right)P_1+\left(r^2+2\left[y-y_o\right]^2\right)P_2=\frac{L_5X+L_6Y+L_7Z+L_8}{L_9X+L_{10}Y+L_{11}Z+1}
\end{aligned}\tag{4-14}$$

where

$\left[\mathrm{x,y}\right]^{\mathrm{T}} =$ measured image coordinates

$\left[\mathrm{x_o,\ y_o}\right]^{\mathrm{T}} =$ focal point projected into image coordinates

$\mathrm{r}^2 = x^2 + y^2$

$\left\{ \mathrm{K1,\ K2,\ K3,\ P1,\ P2} \right\} =$ lens distortion coefficients

$[\ \mathrm{X,Y,Z}]^{\mathrm{T}} =$ imaged point coordinates in the camera reference frame

$\left\{\mathrm{L_1,L_2,L_3,L_4,L_5,L_6,L_7,L_8,L_9,L_{10},L_{11}}\right\} = 11$ unkown camera model parameters

On the basis of experimental investigations, Karara and Abdel-Aziz [KA74] concluded that, for all practical purposes, only the term K_1 needs to be taken into account in modeling lens and image plane distortions. Thus, equation (4-14) can be simplified to

$$x + \left(x - x_o\right)\left(K_1 r^2\right) = \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z + 1}$$

$$y + \left(y - y_o\right)\left(K_1 r^2\right) = \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z + 1} \qquad (4\text{-}15)$$

Two approaches have been identified [Wong76] and used in formulating a least squares solution to equation (4-15). The first approach is direct solution derived from equation (4-15). A minimum of 6 calibration points is required for this solution. The points may be coplanar. The direct approach is tenuous as it does not account for noise in the imaging process. The second approach is an iterative solution that is allows for random errors in the measurement process. This approach requires that the calibration points not be coplanar. Convergence is optimal when the calibration points span the full depth of view. The DLT method has been used in numerous applications with much success [KA74], [Faig71], [Faig72], [Tsai87], and [GT90].

4.3.2 Tsai's Camera Model

The DLT serves as the foundation for camera models in computer vision. However, the DLT is not optimal since it was originally formulated for non-metric photogrammetry with film and did not consider the aspects of electronic imaging with CCD or SIT cameras. In 1987, Roger Tsai presented a seminal paper [Tsai87] that summarized the state of the art in camera calibration and proposed an extension of the

DLT. The reader is encouraged to review this paper as it provides an excellent synopsis of the camera calibration problem and rigorous development of a solution. Tsai proposed a two-stage technique based on a four-step camera model as shown in the figure below.

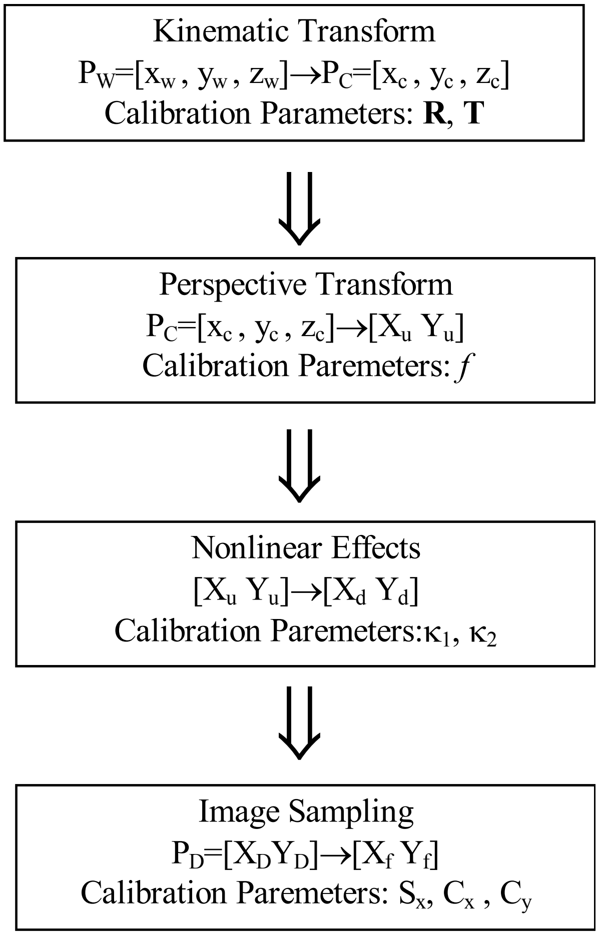

Kinematic Transform
$P_W=[x_w , y_w , z_w]\rightarrow P_C=[x_c , y_c , z_c]$
Calibration Parameters: **R**, **T**
Perspective Transform
$P_C=[x_c , y_c , z_c]\rightarrow[X_u\ Y_u]$
Calibration Paremeters: f
Nonlinear Effects
$[X_u\ Y_u]\rightarrow[X_d\ Y_d]$
Calibration Paremeters: κ_1, κ_2
Image Sampling
$P_D=[X_D Y_D]\rightarrow[X_f\ Y_f]$
Calibration Paremeters: S_x, C_x , C_y

Figure 4-3 Tsai's 4-Step Non-linear Imaging Model [Tsai87]

First, the revised camera model is described by identifying the set of camera parameters to be calibrated followed by Tsai's two-stage calibration technique. The camera imaging process starts with an object point P_W in the world reference frame (object space) as shown in **Figure 4-4**. Note the lens distortion factors have been incorporated into this model differentiating it from the ideal pinhole camera model.

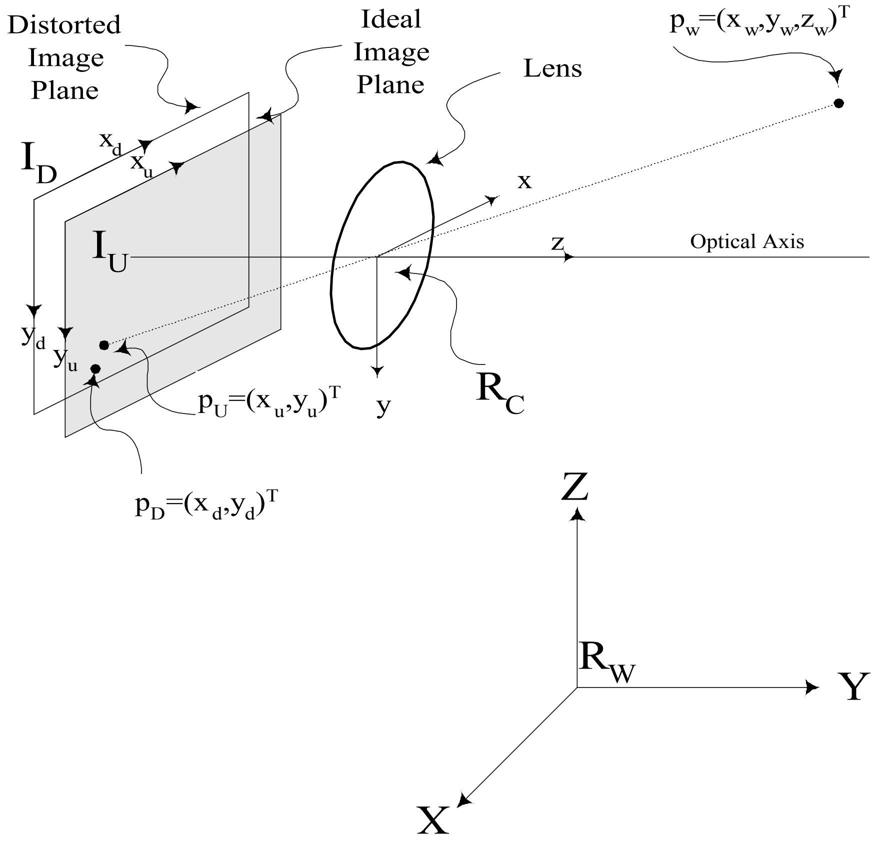


Figure 4-4 Camera Geometry with Perspective Projection and Lens Distortion

The Tsai camera model maps a point in the world reference frame to the image plane accounting for lens distortion. The steps of Tsai's model are as follows.

Step 1: A point $\mathbf{P}_W$ in the object space $\mathbf{R}_W$ is mapped into the camera frame $\mathbf{R}_C$ by the homogeneous transformation

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{4-16}$$

where R is a 3x3 rotation matrix given by

$$R \equiv \begin{bmatrix} \cos\psi\,\cos\theta & \sin\psi\,\cos\theta & -\sin\theta \\ -\sin\psi\,\cos\phi + \cos\psi\,\sin\theta\,\cos\phi & \cos\psi\,\cos\phi + \sin\psi\,\sin\theta\,\sin\phi & \cos\theta\,\sin\phi \\ \sin\psi\,\sin\phi + \cos\psi\,\sin\theta\,\cos\phi & -\cos\psi\,\sin\phi + \sin\psi\,\sin\theta\,\cos\phi & \cos\theta\,\cos\phi \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \tag{4-17}$$

and ϕ, θ, ψ denote the Euler angles where θ is yaw, ϕ is pitch and ψ is roll. **T** is the translation vector given by

$$T \equiv \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \tag{4-18}$$

Note that in this model the transformation from object space to camera space must be rotation followed by translation.

Step 2: The object point $\mathbf{P}_C$ is projected onto the ideal (undistorted) image plane as image co-ordinate $\mathbf{P}_U$=[X_U, Y_U] using the perspective transformation for the pinhole camera

$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \begin{bmatrix} f\dfrac{x_c}{z_c} \\ f\dfrac{y_c}{z_c} \end{bmatrix} \tag{4-19}$$

where f is the effective camera focal length.

Step 3: The ideal image co-ordinate $\mathbf{P}_U$ is distorted from the ideal image plane by radial lens distortion to the true image co-ordinate $\mathbf{P}_D$=[x_d, y_d] by

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} x_u - D_x \\ y_u - D_y \end{bmatrix} \tag{4-20}$$

where

$$D_x = x_d\left(\kappa_1 r^2 + \kappa_2 r^4 + \ldots\right)$$
$$D_y = y_d\left(\kappa_1 r^2 + \kappa_2 r^4 + \ldots\right) \qquad (4\text{-}21)$$

$$r = \sqrt{x_d^2 + y_d^2}$$

and κ_1, κ_2 are the lens distortion coefficients. Note, that two types of lens distortion are considered: radial and tangential. An infinite series is required to model each. Tsai has shown [Tsai87] that only radial distortion need be considered for close-range cameras, and only to the first term.

Step 4: The true image co-ordinate $\mathbf{P}_D$ is mapped to the computer image co-ordinate $\mathbf{P}_f=[x_f, y_f]$. This step accounts for the digital imaging process where light incident on an array of sensors is spatially quantized. The image sampling is given by

$$\begin{bmatrix} x_f \\ y_f \end{bmatrix} = \begin{bmatrix} s_x d_x'^{-1} x_d + C_x \\ d_y'^{-1} x_y + C_y \end{bmatrix} \qquad (4\text{-}22)$$

where

$\begin{bmatrix} x_f \\ y_f \end{bmatrix} \equiv$ row and column numbers of the image pixel in computer frame memory,

$\begin{bmatrix} C_x \\ C_y \end{bmatrix} \equiv$ the row and column numbers of the center of the computer frame memory,

$d'_x \equiv d_x \dfrac{N_{cx}}{N_{fx}}$,

$d_x \equiv$ center to center distance between adjacent CCD sensor elements in x direction,

$d_y \equiv$ center to center distance between adjacent CCD sensor elements in y direction,

$N_{cx} \equiv$ number of sensor elements in the x direction,

$N_{fx} \equiv$ number of pixels sampled per line in computer memory,

$s_x \equiv$ uncertainty image scale factor.

The above 4 steps can be combined into a single set of equations transforming a object in world co-ordinates to computer image co-ordinates

$$s_x^{-1}d'_x x_f + s_x^{-1}d'_x x_f \kappa_1 r^2 = f\frac{r_1 x_w + r_2 y_w + r_3 z_w + X_o}{r_7 x_w + r_8 y_w + r_9 z_w + Z_o}$$

$$d'_y y_f + d'_y y_f \kappa_1 r^2 = f\frac{r_4 x_w + r_5 y_w + r_6 z_w + Y_o}{r_7 x_w + r_8 y_w + r_9 z_w + Z_o} \tag{4-23}$$

where

$$r = \sqrt{\left(s_x^{-1}d'_x x_f\right)^2 + \left(d_y y_f\right)^2} \tag{4-24}$$

Given the image model (4-23) the following parameters must be estimated by the calibration process

T = camera position
R = camera rotation
f = camera effective focal length
κ_1 = lens radial distortion factor
s_x = uncertainty x scale factors_x = uncertainty x scale factor
$\begin{bmatrix} C_x \\ C_y \end{bmatrix}$ = computer image coordinate origin

T and **R** are referred to as extrinsic camera parameters. There are six extrinsic parameters to be calibrated. The parameters *f*, κ_1, s_x, and (C_x, C_y) are referred to as intrinsic parameters. There are six intrinsic parameters to be calibrated. Thus, there are a total of twelve camera parameters to be calibrated.

The first stage of Tsai's technique is to determine the R, X_o, Y_o and s_x. The second stage is to determine *f*, κ_1, and Z_o. These two stages may be implemented with either multiple coplanar calibration targets or a single non-coplanar target using mono-view or multi-view imaging. Tsai's two-stage camera calibration technique has been used extensively in computer vision [WCH92], [WM94], [GT90], [BMZ92], [FLM92], [KC98a], [KC98b], [KC99] with great success.

1.1.1 Modified Tsai Camera Model

Heikkilä and Silvén [HS96], [HS97] proposed and implemented and extension of Tsai's camera calibration technique. The calibration method was implemented as a 4 step procedure in MATLAB™ [HS99]. The camera model is almost identical to Tsai's model with the exceptions that a second radial distortion term and two tangential distortion terms are added. The additional terms were added because first order radial distortion was insufficient to account for distortion in currently available CCD cameras with increased precision. The camera model and calibration procedure follows as presented in [HS97].

Again, an object in world co-ordinates is mapped into the camera frame by

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T \tag{4-25}$$

where R is a 3x3 rotation matrix given by

$$\begin{aligned} R &\equiv \begin{bmatrix} \cos\psi\,\cos\theta & \sin\psi\,\cos\phi\,\cos\theta - \cos\psi\,\sin\phi & \cos\psi\,\sin\phi\,\cos\theta + \sin\psi\,\sin\theta \\ \cos\phi\,\sin\theta & \sin\psi\,\sin\phi\,\sin\theta + \cos\psi\,\cos\theta & \cos\psi\,\sin\phi\,\sin\theta - \sin\psi\,\sin\theta \\ -\sin\phi & \sin\psi\,\cos\phi & \cos\psi\,\cos\phi \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \end{aligned} \tag{4-26}$$

and ψ, ϕ, θ, denote the X-Y-Z Euler angles for rotation about the x-axis, y-axis and z-axis sequentially. Note, that this angle convention is different from Tsai's. T is the translation vector given by

$$T \equiv \begin{bmatrix} X_o \\ Y_o \\ Z_o \end{bmatrix} \tag{4-27}$$

The object point P_C is projected onto the ideal (undistorted) image plane as image co-ordinate $P_U=[x_U, y_U]$ using the perspective transformation for the pinhole camera

$$\begin{bmatrix} x_u \\ y_u \end{bmatrix} = \frac{f}{z_c}\begin{bmatrix} x_c \\ y_c \end{bmatrix} \tag{4-28}$$

where f is the effective camera focal length. The image co-ordinate P_U is mapped to pixel co-ordinates $P_D=[x_d, y_d]$by

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} = \begin{bmatrix} D_u s_u x_u \\ D_v y_u \end{bmatrix} + \begin{bmatrix} u_o \\ v_o \end{bmatrix} \tag{4-29}$$

where

$$\begin{bmatrix} \mathrm{u_o} \\ \mathrm{v_o} \end{bmatrix} \equiv \text{the row and column numbers of the center of the computer frame memory,}$$

$$\begin{bmatrix} \mathrm{D_u} \\ \mathrm{D_v} \end{bmatrix} \equiv \text{center to center distance between adjacent CCD sensor elements in x, y directions,}$$

$$\mathrm{s_x} \equiv \text{uncertainty image scale factor.}$$

D_u and D_v are normally manufacturer specified camera parameters. The radial distortion is given by

$$\begin{bmatrix} \delta_x^{(r)} \\ \delta_y^{(r)} \end{bmatrix} = \begin{bmatrix} x_u\left(\kappa_1 r^2 + \kappa_2 r^4 + \ldots\right) \\ y_u\left(\kappa_1 r^2 + \kappa_2 r^4 + \ldots\right) \end{bmatrix}$$

and (4-30)

$$r = \sqrt{x_u^2 + y_u^2}$$

where κ_1, κ_2 are the lens radial distortion coefficients. The tangential distortion is given by

$$\begin{bmatrix} \delta_x^{(t)} \\ \delta_y^{(t)} \end{bmatrix} = \begin{bmatrix} 2p_1 x_u y_u + p_2\left(r^2 + 2x_u^{\,2}\right) \\ p_1 y_u\left(r^2 + 2y_u^{\,2}\right) + 2p_2 x_u y_u \end{bmatrix} \tag{4-31}$$

where p_1 and p_2 are the lens tangential distortion coefficients. The aggregate pinhole model with radial and tangential distortion is given by

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} D_u s_u\left(x_d + \delta_u^{(r)} + \delta_u^{(t)}\right) \\ D_v\left(y_d + \delta_v^{(r)} + \delta_v^{(t)}\right) \end{bmatrix} + \begin{bmatrix} u_o \\ v_o \end{bmatrix} \qquad (1)\ (4\text{-}32)$$

Thus this model includes additional intrinsic parameters to yield the following 14 camera parameters for calibration

T = camera position

R = camera rotation

f = camera effective focal length

κ_1, κ_2 = lens radial distortion coefficients

p_1, p_2 = lens tangential distortion coefficients

s_x = uncertainty x scale factor

$\begin{bmatrix} u_o \\ v_o \end{bmatrix}$ = computer image coordinate origin

This camera model is expressed in linear homogeneous form as

$$\begin{bmatrix} uw \\ vw \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$
$$= A \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \qquad (4\text{-}33)$$

The matrix **A** is referred to in literature as the DLT matrix. The DLT matrix can be determined by measuring N independent points in the object space to give N image to world correspondences

$$\begin{bmatrix} u_i w_i \\ v_i w_i \\ w_i \end{bmatrix} = A \begin{bmatrix} x_{w_i} \\ y_{wi} \\ z_{wi} \\ 1 \end{bmatrix}, \quad \{i = 1..N\} \qquad (4\text{-}34)$$

By eliminating w, the resulting linear system is expressed as

$$\mathbf{La} = \mathbf{0} \qquad (4\text{-}35)$$

where

$$\mathbf{a} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{21} & a_{22} & a_{23} & a_{24} & a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}^T \qquad (4\text{-}36)$$

is the camera parameter vector and

$$\mathbf{L}=\begin{bmatrix} x_{w1} & y_{w1} & z_{w1} & 1 & 0 & 0 & 0 & 0 & -x_{w1}u_1 & -y_{w1}u_1 & -z_{w1}u_1 & -u_1 \\ 0 & 0 & 0 & 0 & x_{w1} & y_{w1} & z_{w1} & 1 & -x_{w1}v_1 & -y_{w1}v_1 & -z_{w1}v_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{wi} & y_{wi} & z_{wi} & 1 & 0 & 0 & 0 & 0 & -x_{wi}u_i & -y_{wi}u_i & -z_{wi}u_i & -u_i \\ 0 & 0 & 0 & 0 & x_{wi} & y_{wi} & z_{wi} & 1 & -x_{wi}v_i & -y_{wi}v_i & -z_{wi}v_i & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{wN} & y_{wN} & z_{wN} & 1 & 0 & 0 & 0 & 0 & -x_{wN}u_N & -y_{wN}u_N & -z_{wN}u_N & -u_1 \\ 0 & 0 & 0 & 0 & x_{wN} & y_{wN} & z_{wN} & 1 & -x_{wN}v_N & -y_{wN}v_N & -z_{wN}v_N & -v_N \end{bmatrix} \tag{4-37}$$

is the measurement matrix for N independent measurements. Now, this system can be estimated in a least-squares fashion. QR decomposition is used to obtain the calibration parameters [HS97]. The decomposition is given as

$$\mathbf{A}=\lambda\mathbf{V}^{-1}\mathbf{B}^{-1}\mathbf{FRT} \tag{4-38}$$

where λ is an overall scaling factor

$$\mathbf{V}=\begin{bmatrix} 1 & 0 & -u_o \\ 0 & 1 & -v_o \\ 0 & 0 & 1 \end{bmatrix} \tag{4-39}$$

gives the principal point(u_o,v_o),

$$\mathbf{B}=\begin{bmatrix} 1+b_1 & b_2 & 0 \\ b_2 & 1-b_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-40}$$

gives the linear distortion factors b_1 and b_2, and

$$\mathbf{F}=\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4-41}$$

gives the focal length. Direct decomposition is computationally fast, but leaves out the lens distortion factors and does not account for measurement noise.

An alternate solution is via non-linear estimation. The sum of square residuals

$$E=\sum_{i=1}^{N}\left(U_i-u_i\right)^2+\sum_{i=1}^{N}\left(V_i-v_i\right)^2 \tag{4-42}$$

is minimized using Levengberg-Marquart optimization. The preceding calibration procedure was implemented as a series of programs in MATLAB [HS99].

This camera calibration procedure is used to calibrate the CCD camera used for calibration experiments described in Chapter 6.

4.4 Image acquisition and feature detection

The first step in photogrammetric measurement is the detection of features to be measured. A means to detect image features of interest is required in order to calibrate a digital camera and use it for photogrammetric measurement. A marker detection scheme is proposed here for the detection 10-20 mm circular fiducial markers. The detection scheme is completely autonomous and is described below. Circular fiducial markers are used rather than other markers, such as squares or diamonds, because more accurate centroid estimates are expected as shown by Bose and Amir [BoA90].

The measurement process begins with image acquisition of images from the camera. The image data is obtained from the camera via parallel interface to the computer. The image data is stored in volatile memory as a 24-bit RGB bitmap. The data is later written to hard disk drive for long term storage.

It is not assumed that the sampling period is uniform. Figure 4-5 shows a histogram of sample periods for the acquisition of 1000 images. The sample period is nominally 150 msec. Sample jitter as much as 400 msec requires the images to be time stamped to support subsequent image sequence processing. The time-stamping process has negligible effect on sample period.

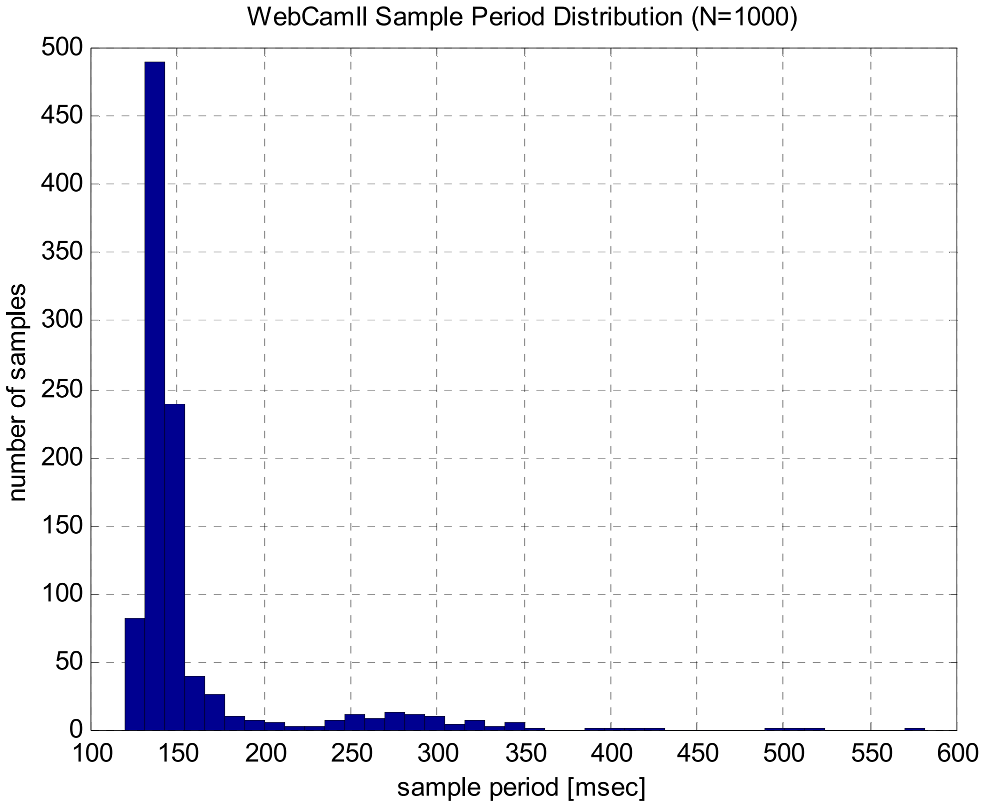


Figure 4-5 CCD camera sample period histogram

Figure 4-6 shows a calibration target that consists of a two 400 mm x 400 mm faces with 64 (8x8) calibration fiducial markers per face to give a total of 128 calibration points. The diameter of the calibration markers is 20 mm. The markers are unique in color with respect to the image scene to simplify their detection.

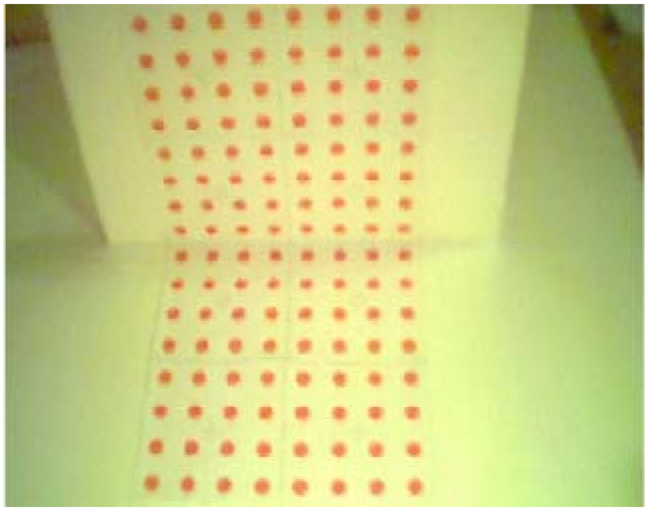


Figure 4-6 RGB Image of Calibration Target

The input image is an encoded as 24-bit RGB. The first stage of marker detection is to convert the RGB image to an 8-bit indexed image. The histogram of the indexed image is also normalized. The image conversion maps RGB pixel values [R G B] to a number on the interval [0..255] using the color map shown in **Figure 4-7**. The color map was derived such that red pixels of the same color as the markers are mapped to values in the range [200..255] other pixels are mapped to lower index levels.

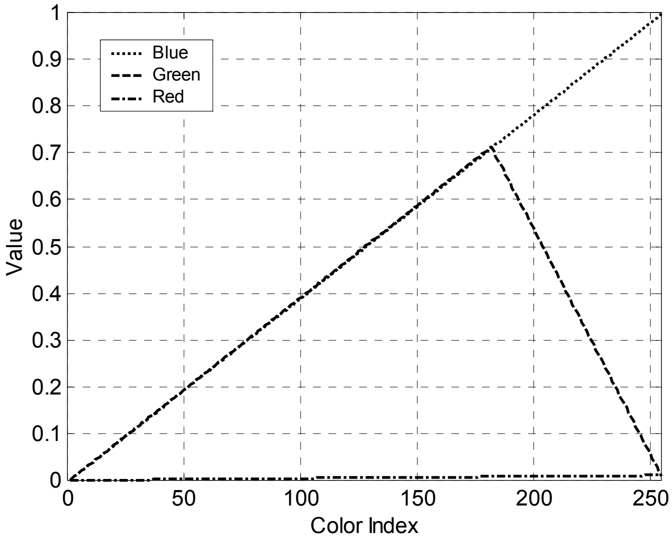


Figure 4-7 RGB Index Color map

Figure 4-8 shows the 8-bit indexed image corresponding to the RGB image in Figure 4-6. The 128 markers are clearly visible in the indexed image. However, some noise is evident. Filtering is required to improve SNR for marker detection.

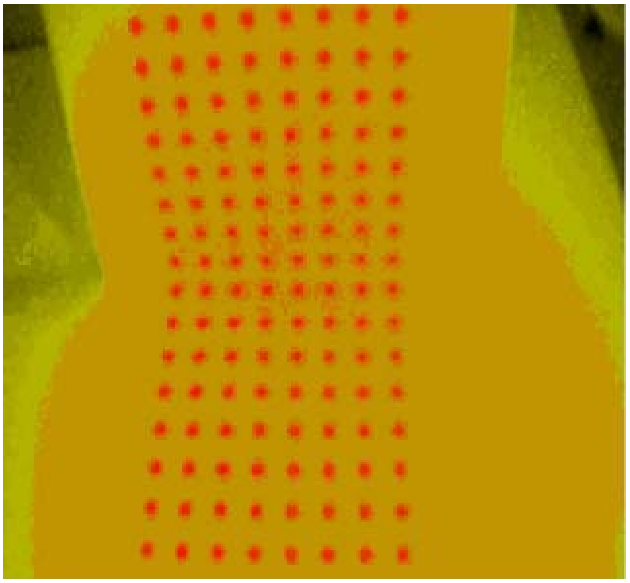


Figure 4-8 8-bit Indexed Image

The indexed image is normalized then thresholded to yield regions of interest as shown in **Figure 4-9**.

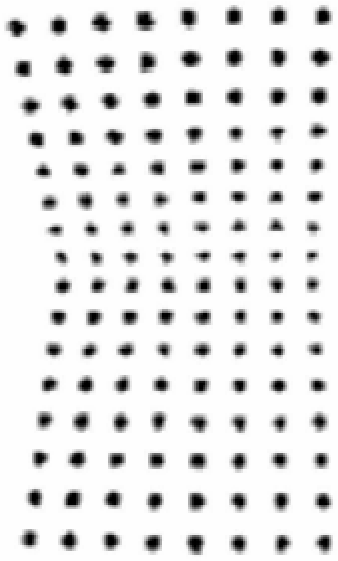


Figure 4-9 8-bit Filtered Image

A morphological erosion operator converts the thresholded image to a binary image. The erosion is performed recursively with a 3x3 square structuring element until the number of distinct labeled objects in the image equals the number of expected targets. This assumes that the thresholding process retains all targets in the image.

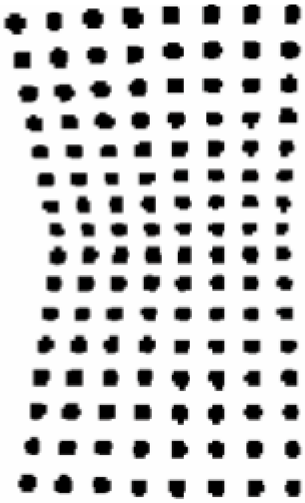


Figure 4-10 Binary Morphed Image

Each contiguous region in the morphed image is labeled to uniquely identify each marker as shown in **Figure 4-11**.

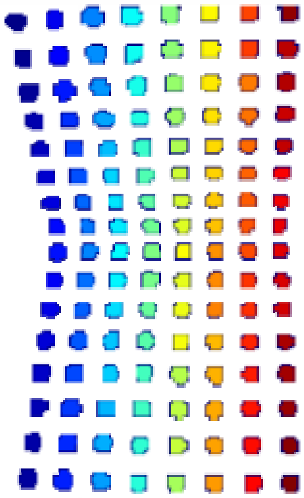


Figure 4-11 Labeled Image

Finally, the center of each labeled region calculated using the centroid method [BoA90] and marker image co-ordinates estimated as shown in **Figure 4-12**.

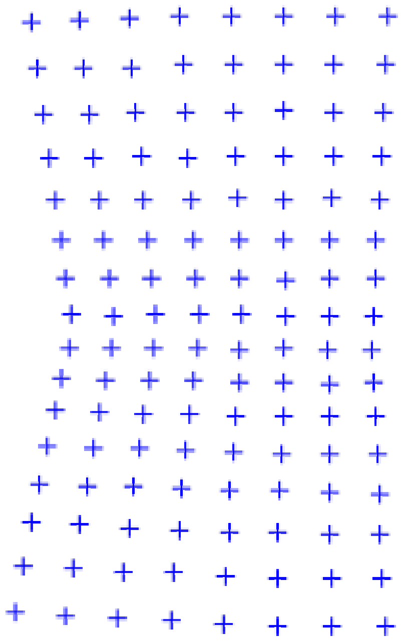


Figure 4-12 Detected Calibration Targets

Figure 4-13 shows the estimated fiducial marker positions overlaid on the original target image. The close agreement of centroid estimates is apparent in **Figure 4-13**.

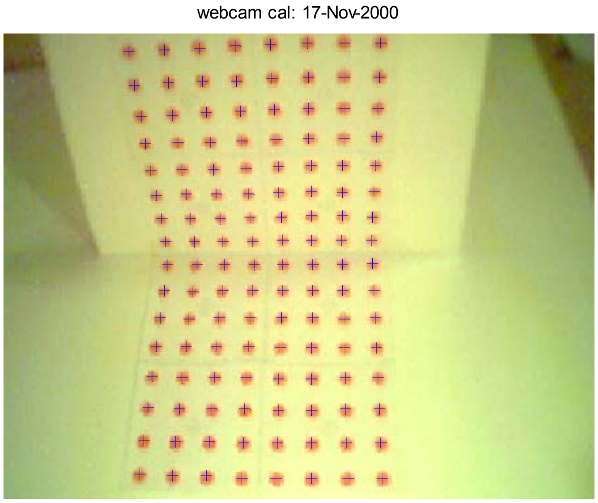


Figure 4-13 Calibration Target with Overlaid Centroids

Centroid accuracy was estimated by conducting an experiment where a fixed target was imaged repeatedly. 16000 images of the target were captured. The centre of the target in each image was estimated by the process described above. Figure 4-14 is a scatter plot of the centroid estimates for the 16000 samples.

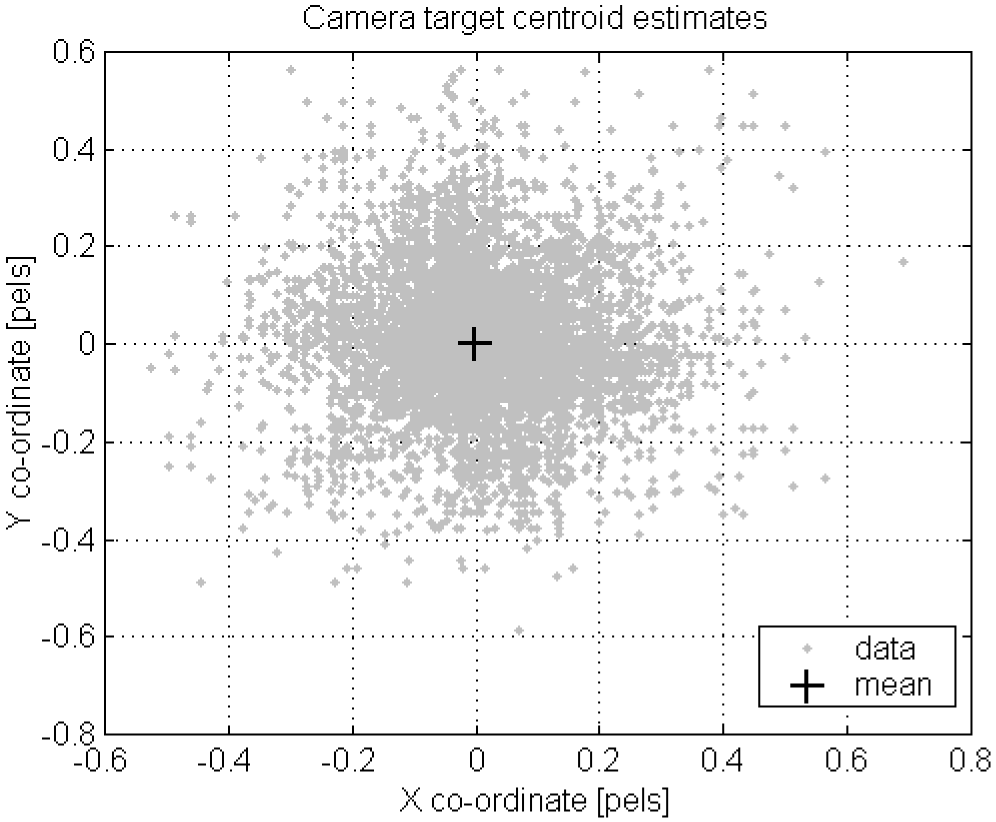


Figure 4-14 Target Centroid Estimates [N=16000]

Figure 4-15 shows the estimated distribution of the centroid estimate error in the horizontal and vertical axes.. The red curve is the normal distribution of equivalent mean and variance. The error variance of centroid estimates can be expected to be within 0.015 pixels. Note that the centroid estimate is unbiased. This error is important in the pose estimation scheme described in the following section.

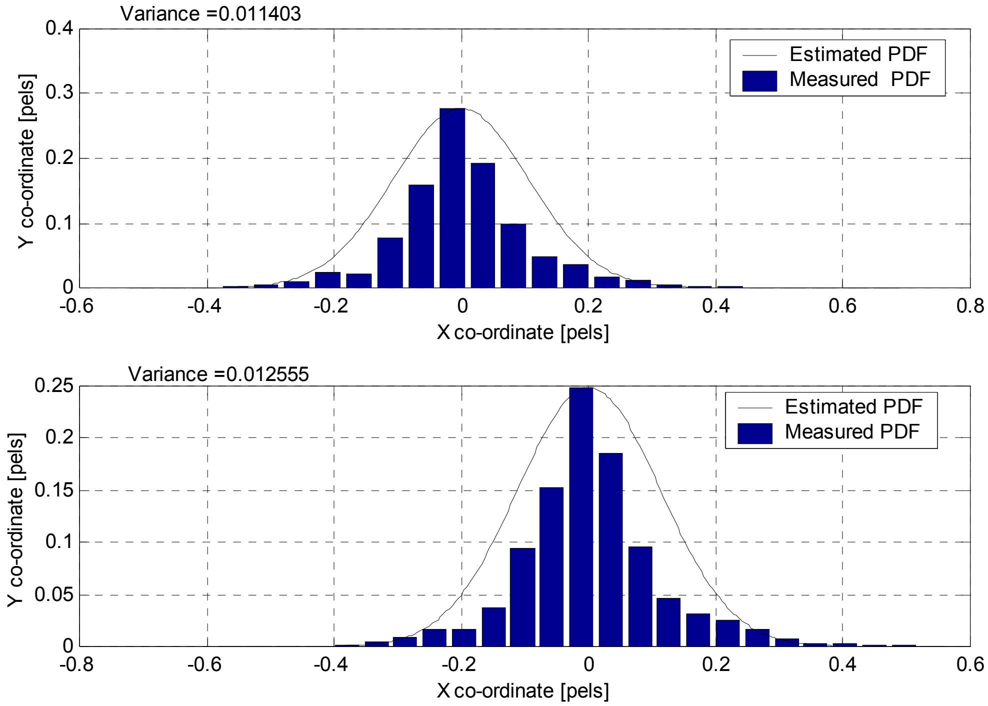


Figure 4-15 Image Centroid Estimate Distributions

Figure 4-16 illustrates the proposed target feature detection process.

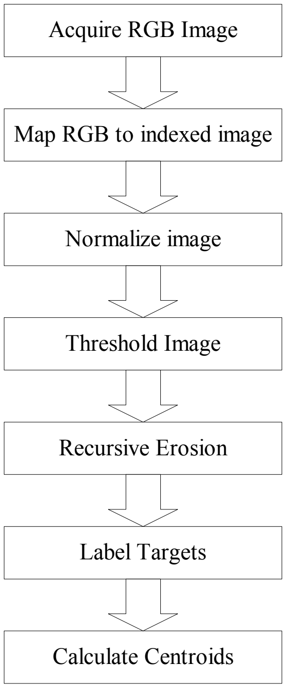


Figure 4-16 Image Feature Detection Process

4.5 Pose Estimation

Pose estimation is a fundamental problem encountered in computer vision applications. The ability to determine the position and orientation of an object by camera measurements has numerous benefits for industrial and scientific applications. The pose estimation problem is defined as determining the position (x,y,z) and orientation (roll, pitch, yaw) from sampled images of the object. The position and orientation of the object are defined the camera reference frame.

A fully projective formulation of Lowe's pose recovery algorithm [ACB98] is the pose estimation method used in the robot calibration methodology proposed in this thesis. David Lowe at the University of British Columbia [Lowe91], [Lowe87] proposed a solution to the pose estimation problem known throughout computer vision literature as Lowe's algorithm. Figure 4-17 illustrates the concept of Lowe's pose recovery algorithm. Lowe's algorithm solves the inverse perspective problem described in equation (4-12) by optimizing the pose of an object model to match the corresponding features in an image of the object. Lowe proposed use of Newton's method to find a least-squares solution for the object pose. This requires the object 3-D model a reasonable initial guess of object pose. Otherwise, the algorithm will converge in an incorrect local minimum. Lowe's algorithm has received much attention and use because of its simplicity and generality [ACB98]. Lowe's formulation assumed local linearity in the perspective transform to make the least-squares solution more tractable.

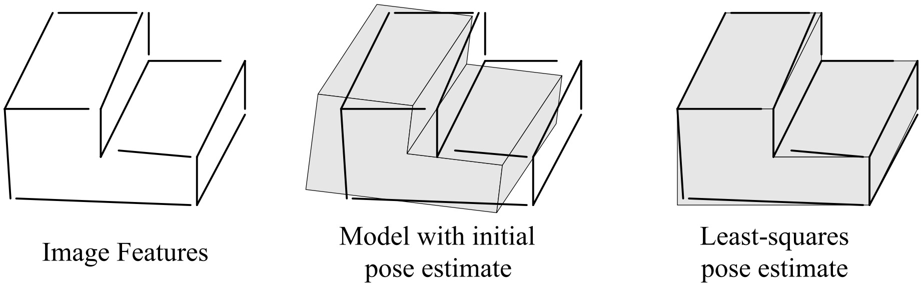


Figure 4-17 Lowe's Pose Recovery Algorithm

Araújo, Carceroni and Brown [ACB98], [ACB97], [CB97] removed Lowe's assumptions of local linearity to give a fully projective formulation of Lowe' algorithm. This is the algorithm used for pose estimation of mobile robot kinematic state. This algorithm is derived in the following text as described in [ACB98].

Let equation (4-43) define the projection of a three-dimensional object point $\mathbf{p} = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T$ onto the image plane $(u,v) \in \mathbb{R}^2$ with respect to the object reference frame

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathbf{R}(\mathbf{p} - \mathbf{t})$$
$$[u\ v] = \begin{bmatrix} \frac{fx}{z} & \frac{fy}{z} \end{bmatrix} \tag{4-43}$$

f is the camera focal length. **R** denotes the 3x3 rotation matrix of the roll, pitch and yaw angles (ψ, φ, θ).

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$
$$= \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi\cos\phi + \cos\psi\sin\theta\cos\phi & \cos\psi\cos\phi + \sin\psi\sin\theta\sin\phi & \cos\theta\sin\phi \\ \sin\psi\sin\phi + \cos\psi\sin\theta\cos\phi & -\cos\psi\sin\phi + \sin\psi\sin\theta\cos\phi & \cos\theta\cos\phi \end{bmatrix} \tag{4-44}$$

T denotes the translation vector given by

$$\mathbf{t} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \tag{4-45}$$

Lowe [Lowe87] proposed a re-parameterization of perspective model in (4-43) equation to remove the translation terms. This is done by expressing equation (4-43) in the camera reference frame. The perspective projection model becomes

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{Rp}$$
$$(u, v) = \left(f\frac{x' + D_x}{z' + D_z}, f\frac{y' + D_y}{z' + D_z} \right) \tag{4-46}$$

where the new parameters D_x, D_y, D_z are given by

$$\begin{aligned} D_x &= -\left(r_{11}t_x + r_{12}t_y + r_{13}t_z\right) \\ D_y &= -\left(r_{21}t_x + r_{22}t_y + r_{23}t_z\right) \\ D_z &= -\left(r_{31}t_x + r_{32}t_y + r_{33}t_z\right) \end{aligned} \tag{4-47}$$

Equation (4-46) represents a non-linear system of equations that can be solved by application of the vector case of Newton's method. Application of Newton's method requires computation of the partial derivatives of the u and v with respect to the translation and rotation parameters (x, y, z, ψ, ϕ, θ). Table 4-1 summarizes the partial derivatives.

	u	v
$\frac{\partial}{\partial D_x}$	$f\left(x'+D_x\right)$	0
$\frac{\partial}{\partial D_y}$	0	$f\left(x'+D_x\right)$
$\frac{\partial}{\partial D_z}$	$-f\frac{}{\left(z'+D_z\right)^2}$	$-f\frac{\left(y'+D_y\right)}{\left(z'+D_z\right)^2}$
$\frac{\partial}{\partial \psi}$	$-f\frac{\left(x'+D_x\right)y'}{\left(z'+D_z\right)^2}$	$-\frac{f}{\left(z'+D_z\right)}\left(z'+\frac{y'\left(y'+D_y\right)}{z'+D_z}\right)$
$\frac{\partial}{\partial \phi}$	$\frac{f}{z'+D_z}\left(z'+\frac{x'\left(x'+D_x\right)}{z'+D_z}\right)$	$\frac{fx'\left(y'+D_y\right)}{\left(z'+D_z\right)^2}$
$\frac{\partial}{\partial \theta}$	$\frac{-fy'}{z'+D_z}$	$\frac{-fx'}{z'+D_z}$

Table 4-1 Partial derivatives of *u* and *v* with respect to camera extrinsic parameters

Let $\mathbf{q}^{\mathbf{i}}$ denote the current pose estimate. Lowe's algorithm applies Newton's method to compute a pose vector correction $\Delta\mathbf{q} = \left[\Delta D_x \quad \Delta D_y \quad \Delta D_z \quad \Delta\psi \quad \Delta\phi \quad \Delta\theta\right]$ used calculate the next pose estimate

$$\mathbf{q}^{i+1} = \mathbf{q}^{i} - \Delta\mathbf{q} \tag{4-48}$$

The image point e error terms are related to the pose vector correction by the linear equation in equation (4-49) where **J** is the Jacobian of the error terms. The Jacobian is computed using the partial derivative terms in Table 4-1.

$$\mathbf{J}\Delta\mathbf{q} = \mathbf{e} \qquad \mathbf{J}_{ij} = \frac{\partial \mathbf{e}_i}{\partial \mathbf{x}_j} \tag{4-49}$$

Taking the pseudo-inverse of the linear equation (4-49) gives the estimated correction pose correction.

$$\Delta\mathbf{q} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{e} \tag{4-50}$$

The error is given by the difference between the measured image feature points projected image feature points. The projected feature points are determined applying the calibrated camera model to the object model and current pose estimate. If the current pose estimate is identical to the pose in the image the error is zero.

Recalling equation (4-1), let **M** denote the object model as a set of n features points

$$\mathbf{M} = \left[\mathbf{p}_1 \quad \mathbf{p}_2 \quad \ldots \quad \mathbf{p}_n\right], \qquad \mathbf{p}_i = \left[x_i \quad y_i \quad z_i\right]^T \in \mathbb{R}^3 \tag{4-51}$$

Let $\mathbf{I}_{\mathrm{act}}$ be the measured image with n corresponding feature points given by

$$\mathbf{I} = \left[\mathbf{i}_1 \quad \mathbf{i}_2 \quad \cdots \quad \mathbf{i}_n\right], \qquad \mathbf{i}_i = \left(u_i, v_i\right) \in \mathbb{R}^2 \tag{4-52}$$

The feature points are detected by the algorithm described in Section 4.4. Thus the error in equation (4-49) is given by

$$\mathbf{e} = \mathbf{I}_{act} - \mathbf{I}_{proj} \tag{4-53}$$

where the $\mathbf{I}_{\mathrm{proj}}$ is the set of features points from the projection of the current pose estimate. The image projection uses the calibrated model derived in Section 4.3.3.

Hence, the fully projective formulation of Lowe's pose recovery algorithm is given by repeatedly adjusting the estimated pose as a function of the computed difference between the feature points in actual and projected images. The algorithm is repeated for a fixed number of iterations or until user specified error criteria are satisfied. Araújo, Carceroni and Brown [ACB98] showed that 20 iterations are sufficient practical applications. Figure 4-18 illustrates the overall pose recovery algorithm.

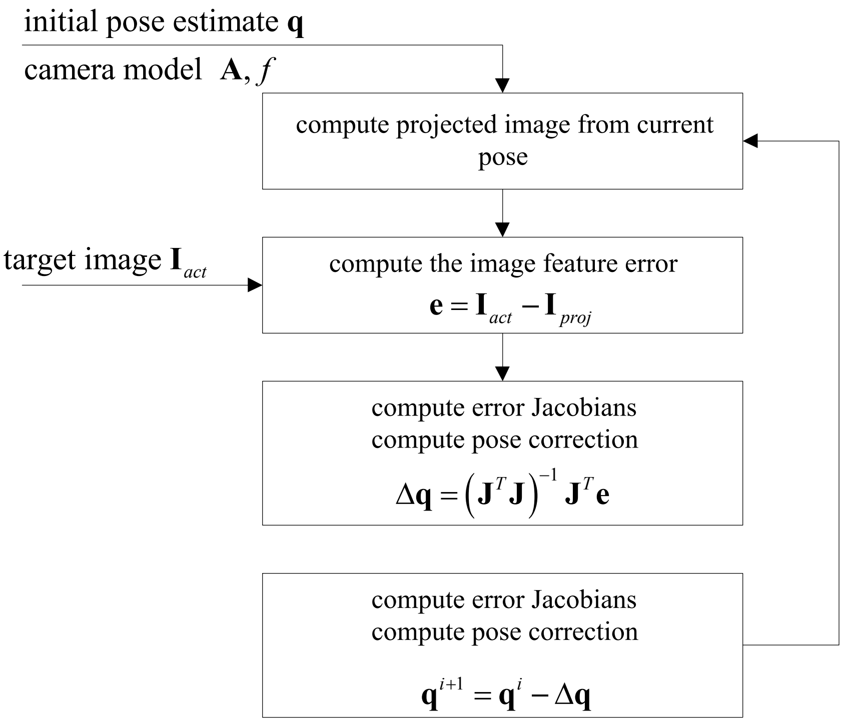

initial pose estimate $\mathbf{q}$
camera model $\mathbf{A}, f$
compute projected image from current pose
target image $\mathbf{I}_{act}$
compute the image feature error
$\mathbf{e} = \mathbf{I}_{act} - \mathbf{I}_{proj}$
compute error Jacobians
compute pose correction
$\Delta\mathbf{q} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{e}$
compute error Jacobians
compute pose correction
$\mathbf{q}^{i+1} = \mathbf{q}^i - \Delta\mathbf{q}$

Figure 4-18 Fully Projective Lowe Pose Recovery Algorithm

5. Model Parameter Estimation

Chapter 3 established kinematic models of mobile robots. Chapter 4 proposed photogrammetry as a non-intrusive means to measure kinematic state. This chapter addresses the problem of how to estimate the parameters of a mobile robot kinematic model given the model structure and noisy measurements of state for known control inputs.

The proposed solution to the state estimation problem is a stochastic estimation algorithm. The inputs to the algorithm are the kinematic model and measurements of state coupled with robot control values. The algorithm is required to meet the following criteria:

- Good estimation accuracy,
- Ease of implementation,
- High stability and robustness, and
- Low computational load.

The proposed algorithm is the second order divided difference filter, referred to as the DD2 filter, and developed by Norgaard [NPR00a]. The DD2 filter is an improvement of the extended Kalman filter that uses Stirling interpolation, rather than the Taylor series derivative based approximation, to linearize the system and observation processes. The selection of the filter is due to its improved performance, compared to the extended Kalman filter, and its ease of implementation and use.

This chapter is organized as follows. The first section of this chapter derives the linear form of the Kalman filter since it is the foundation of the EKF and DD2 filters. The second section derives the extended Kalman filter to account for non-linear processes. The third section derives the second order divided difference filter as an improvement on the extended Kalman filter.

5.1 Linear Kalman Filter

The Kalman filter, named after Rudolf E. Kalman, is an optimal solution to the discrete-data linear filtering problem [Kalman60]. The filter is derived by finding the estimator for a linear system, subject to additive white Gaussian noise, that meets the following three criteria:

1. The estimator provides an unbiased estimate of the system state.

2. The estimator provides a minimum variance estimate of system state.

3. The estimator is a linear function.

The Kalman filter derivation begins with definition of the system and assumptions. Consider a process *f(.)* given by the linear difference equation

$$\begin{aligned}\mathbf{x}_{k+1} &= f\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k\right)\\ &= \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k\end{aligned} \tag{5-1}$$

where

$\mathbf{x}_k = (\mathrm{n}\times 1)$ system state vector at time $\mathrm{t_k}$, $\mathbf{x} \in \mathbb{R}^n$
$\mathbf{u}_k = (l\times 1)$ system control vector at time $\mathrm{t_k}$, $\mathbf{u} \in \mathbb{R}^l$
$\mathbf{A}_k = (\mathrm{n}\times \mathrm{n})$ state transition matrix mapping $\mathbf{x}_\mathrm{k}$ to $\mathbf{x}_\mathrm{k+1}$
$\mathbf{B}_k = (\mathrm{n}\times l)$ control matrix mapping $\mathbf{u}_\mathrm{k}$ to $\mathbf{x}_\mathrm{k}$
$\mathbf{w}_k = (\mathrm{n}\times 1)$ system noise vector given by n dimensional Wiener process
of known covariance $\mathbf{Q}$

Note that many descriptions of the Kalman filter do not include the control vector $\mathbf{u}_\mathrm{k}$. The control vector is included here in anticipation of the application to the kinematic models developed in Chapter 3.

The process *f()* may not be directly observable. This means that the state **x** cannot be completely determined from measurement. To account for this an observation model *h()* is introduced. The observation or measurement model is a mapping of the actual system state **x** to an observed state **y**. The observation model is given by the discrete time linear difference equation

$$\begin{aligned}\mathbf{y}_k &= h\left(\mathbf{x}_k, \mathbf{v}_k\right) \\ &= \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k\end{aligned} \tag{5-2}$$

where

$\mathbf{y}_k = (\mathrm{m} \times 1)$ observed state vector at time $\mathrm{t_k}$, $\mathbf{y} \in \mathbb{R}^m$
$\mathbf{H}_k = (\mathrm{n} \times \mathrm{n})$ measurement matrix mapping $\mathbf{x}_k$ to $\mathbf{y}_k$ at time $\mathrm{t_k}$
$\mathbf{v}_k = (\mathrm{n} \times 1)$ mesurement noise vector given by n dimensional Wiener process
of known covariance $\mathbf{R}$

Note that equations (5-1) and (5-2) constitute the normal state space formulation of a linear system used extensively in control theory. The system and measurement noise are assumed to be unbiased normally distributed random processes

$$\begin{aligned}p(\mathbf{w}) &\sim N(0,\mathbf{Q}) \\ p(\mathbf{v}) &\sim N(0,\mathbf{R})\end{aligned} \tag{5-3}$$

The covariance matrices of the **w** and **v** are given by

$$\begin{aligned}E\left[\mathbf{v}_k \mathbf{v}_i^T\right] &= \begin{cases}\mathbf{R}_k & ,i=k \\ 0 & ,i \neq k\end{cases} \\ E\left[\mathbf{w}_k \mathbf{w}_i^T\right] &= \begin{cases}\mathbf{Q}_k & ,i=k \\ 0 & ,i \neq k\end{cases}\end{aligned} \tag{5-4}$$

where $E[x] = \sum_{i=1}^{n} p_i x_i$ denotes the mathematical expectation of the discrete random variable x where $\mathrm{p_i}$ is the probability of the occurrence event $\mathrm{x_i}$. It is assumed that **w** and **v** are independent random processes and hence have zero cross-correlation

$$E\left[\mathbf{w}_k \mathbf{v}_i^T\right] = 0, \ \forall i,k \tag{5-5}$$

Given the above definitions, the filtering problem can be defined as arriving at an optimal estimate of the state $\mathbf{x}_\mathrm{k}$ conditioned by the measurement $\mathbf{y}_\mathrm{k}$. We wish to implement a recursive linear estimator, as shown in **Figure 5-1**, which provides the optimal, in the least squares sense, estimate of system state $\mathbf{x}_{\mathrm{k+1}}$ given the measurement $\mathbf{y}_\mathrm{k}$ and the current state $\mathbf{x}_\mathrm{k}$.

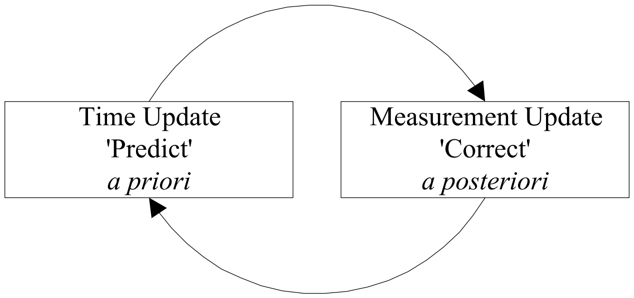


Figure 5-1 Recursive Estimator

Let $\hat{\mathbf{x}}_k^- \in \mathbb{R}^n$ be the *a priori* state estimate at time t$_k$ given knowledge of the system history, and let $\hat{\mathbf{x}}_k \in \mathbb{R}^n$ be the a posteriori state estimate at time t$_k$ given measurement $\mathbf{z}_k$. The a priori (before measurement) and a posteriori (after measurement) estimate errors are defined as

$$\begin{aligned} e_k^- &= \mathbf{x}_k - \hat{\mathbf{x}}_k^- \quad && \textit{a priori} \\ e_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k \quad && \textit{a posteriori} \end{aligned} \tag{5-6}$$

The *a priori* estimate error covariance is then given by

$$\begin{aligned} P_k^- &= E\left[e_k^- e_k^{-T}\right] \\ &= E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right)^T\right] \end{aligned} \tag{5-7}$$

and the *a posteriori* estimate error covariance is

$$\begin{aligned} P_k &= E\left[e_k e_k^{\,T}\right] \\ &= E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T\right] \end{aligned} \tag{5-8}$$

As stated above the desired estimator should be linear recursive and hence of the form

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right) \tag{5-9}$$

The difference $\left(\mathbf{y}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)$in equation (5-9) is known as the residual or measurement innovation. It is a measure of the error between the *a priori* measurement $\mathbf{H}\hat{\mathbf{x}}_k^-$ and the *a posteriori* measurement $\mathbf{y}_k$. Zero residual implies the prediction matches the measurement. The n×m matrix $\mathbf{K}_k$ is the gain factor that weights the residual.

The Kalman filter chooses $\mathbf{K}_k$ such that the a posteriori estimate is optimal in terms of the linear minimum mean-square error. We want to determine $\mathbf{K}_k$ to provide an optimum unbiased minimum variance estimate of state. Thus, we seek $\mathbf{K}_k$ such that P_k is minimized.

A minimum solution for P_k is found as follows. Substitute the observation model (5-2) into the linear recursive estimator equation (5-9)

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right) \tag{5-10}$$

Substitute equation (5-10) into the *a posteriori* estimate error covariance

$$\begin{aligned}\mathbf{P}_k &= E\left[e_k e_k{}^T\right]\\ &= E\left[\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right)^T\right]\\ &= E\left[\left(\mathbf{x}_k - \left(\hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)\right)\right)\left(\mathbf{x}_k - \left(\hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)\right)\right)^T\right]\\ &= E\left[\left(\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right) - \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)\right)\left(\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right) - \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)\right)^T\right]\end{aligned} \tag{5-11}$$

Note that the term $\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right)$ is the a priori estimate error $\mathbf{e}_k^-$ and equation (5-11) becomes

$$\begin{aligned}
\mathbf{P}_k &= E\left[\left(\mathbf{e}_k^- - \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)\right)\left(\mathbf{e}_k^- - \mathbf{K}_k\left(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}\hat{\mathbf{x}}_k^-\right)\right)^T\right] \\
&= E\left[\left(\mathbf{e}_k^- - \mathbf{K}_k\mathbf{H}_k\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right) - \mathbf{K}_k\mathbf{v}_k\right)\left(\mathbf{e}_k^- - \mathbf{K}_k\mathbf{H}_k\left(\mathbf{x}_k - \hat{\mathbf{x}}_k^-\right) - \mathbf{K}_k\mathbf{v}_k\right)^T\right] \\
&= E\left[\left(\mathbf{e}_k^- + \mathbf{K}_k\mathbf{H}_k\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k\right)\left(\mathbf{e}_k^- + \mathbf{K}_k\mathbf{H}_k\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k\right)^T\right] \\
&= E\left[\left(\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k\right)\left(\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k\right)^T\right] \\
&= E\left[\left(\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k\right)\left(\mathbf{e}_k^{-T}\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T - \mathbf{v}_k^T\mathbf{K}_k^T\right)\right]
\end{aligned} \tag{5-12}$$

Expanding the terms in (5-12) yields

$$\mathbf{P}_k = E\begin{bmatrix} \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{e}_k^-\mathbf{e}_k^{-T}\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T - \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{e}_k^-\mathbf{v}_k^T\mathbf{K}_k^T \\ -\mathbf{K}_k\mathbf{v}_k\mathbf{e}_k^{-T}\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T + \mathbf{K}_k\mathbf{v}_k\mathbf{v}_k^T\mathbf{K}_k^T \end{bmatrix} \tag{5-13}$$

The only random variables in equation (5-13) are e_k^- and V_k. Thus, the expected values of the cross terms in equation (5-13) vanish. The covariance of the *a posteriori* estimate error becomes

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k{}^T \tag{5-14}$$

The optimum K_k is determined by minimizing the trace of equation (5-14), which gives

$$\begin{aligned}
\frac{d\left(Tr\mathbf{P}_k\right)}{d\mathbf{K}_k} &= \frac{d}{d\mathbf{K}_k}Tr\left[\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^-\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k{}^T\right] \\
&= -2\left(\mathbf{H}_k\mathbf{P}_k^-\right)^T + 2\mathbf{K}_k\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k{}^T + \mathbf{R}_k\right)
\end{aligned} \tag{5-15}$$

and thus the optimum gain $\mathbf{K}_k$ is given by the solution to

$$0 = -2\left(\mathbf{H}_k\mathbf{P}_k^-\right)^T + 2\mathbf{K}_k\left(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k{}^T + \mathbf{R}_k\right) \tag{5-16}$$

Thus, the gain for the Kalman filter that provides an unbiased minimum variance estimate is given by

$$\mathbf{K}_k = \mathbf{P}_k^{-}\mathbf{H}_k{}^T\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k{}^T + \mathbf{R}_k\right)^{-1} \tag{5-17}$$

We now compute the estimate error covariance associated with the optimal filter gain in equation (5-17). Recall equation (5-14) that is expanded to give

$$\begin{aligned}\mathbf{P}_k &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^{-}\left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \\ &= \mathbf{P}_k^{-} - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^{-} - \mathbf{P}_k^{-}\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{R}_k\right)\mathbf{K}_k^T\end{aligned} \tag{5-18}$$

This expression is valid for any filter gain **K** whether it is optimal or suboptimal. Substituting equation (5-17) into equation (5-18) gives the optimal *a posteriori* error covariance

$$\begin{aligned}\mathbf{P}_k &= \mathbf{P}_k^{-} - \left(\mathbf{P}_k^{-}\mathbf{H}_k{}^T\left(\mathbf{H}_k P_k^{-}\mathbf{H}_k{}^T + \mathbf{R}_k\right)^{-1}\right)\mathbf{H}_k\mathbf{P}_k^{-} - \mathbf{P}_k^{-}\mathbf{H}_k^T\left(\mathbf{P}_k^{-}\mathbf{H}_k{}^T\left(\mathbf{H}_k P_k^{-}\mathbf{H}_k{}^T + \mathbf{R}_k\right)^{-1}\right)^T \\ &\quad + \left(\mathbf{P}_k^{-}\mathbf{H}_k{}^T\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k{}^T + \mathbf{R}_k\right)^{-1}\right)\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k^T + \mathbf{R}_k\right)\left(\mathbf{P}_k^{-}\mathbf{H}_k{}^T\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k{}^T + \mathbf{R}_k\right)^{-1}\right)^T \\ &= \mathbf{P}_k^{-} - \mathbf{P}_k^{-}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_k^{-}\mathbf{H}_k{}^T + \mathbf{R}_k\right)^{-1}\mathbf{H}_k\mathbf{P}_k^{-} \\ &= \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k\right)\mathbf{P}_k^{-}\end{aligned} \tag{5-19}$$

Equations (5-10), (5-17) and (5-19) form the *a posteriori* step of the discrete Kalman filter. The a priori step of the filter involves computing the *a priori* state estimate $\mathbf{x}_{k+1}^{-}$and the *a priori* estimate error covariance $\mathbf{P}_{k+1}^{-}$. The *a priori* state estimate is given by equation (5-1) with $\mathbf{w}_k$ removed since it has zero mean and is un-correlated with other noise values $\mathbf{w}_{j\neq k}$

$$\hat{\mathbf{x}}_{k+1}^{-} = \mathbf{A}_k\hat{\mathbf{x}}_k + \mathbf{B}_k\mathbf{u}_k \tag{5-20}$$

The *a priori* estimate error covariance is found first by expressing equation (5-20) in terms of the a priori error

$$
\begin{aligned}
\mathbf{e}_{k+1}^{-} &= \mathbf{x}_k - \hat{\mathbf{x}}_k^{-} \\
&= \left(\mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{w}_k\right) - \left(\mathbf{A}_k\hat{\mathbf{x}}_k + \mathbf{B}_k\mathbf{u}_k\right) \\
&= \mathbf{A}_k\mathbf{e}_k + \mathbf{w}_k
\end{aligned}
\tag{5-21}
$$

Substituting equation (5-21) into equation (5-7) yields the *a priori* estimate error covariance

$$
\begin{aligned}
\mathbf{P}_{k+1}^{-} &= E\left[\mathbf{e}_{k+1}^{-}\mathbf{e}_{k+1}^{-T}\right] = \\
&= E\left[\left(\mathbf{A}_k\mathbf{e}_k + \mathbf{w}_k\right)\left(\mathbf{A}_k\mathbf{e}_k + \mathbf{w}_k\right)^T\right] \\
&= E\left[\mathbf{A}_k\mathbf{e}_k\mathbf{e}_k^T\mathbf{A}_k^T + \mathbf{A}_k\mathbf{e}_k\mathbf{w}_k^T + \mathbf{w}_k\mathbf{e}_k^T\mathbf{A}_k^T + \mathbf{w}_k\mathbf{w}_k^T\right]
\end{aligned}
\tag{5-22}
$$

And since $\mathbf{w}_k$ and $\mathbf{e}_k$ are un-correlated, $E\left[\mathbf{w}_k\mathbf{e}_k^T\right] = 0$, the a priori estimate error covariance is given by

$$
\begin{aligned}
\mathbf{P}_{k+1}^{-} &= E\left[\mathbf{A}_k\mathbf{e}_k\mathbf{e}_k^T\mathbf{A}_k^T + \mathbf{w}_k\mathbf{w}_k^T\right] \\
&= \mathbf{A}_k\mathbf{P}_k\mathbf{A}_k^T + \mathbf{Q}_k
\end{aligned}
\tag{5-23}
$$

The complete Kalman filter algorithm is given by equations (5-10), (5-17) and (5-19) for the *a posteriori* step, and equations (5-20) and (5-23) for the *a priori* step. Figure 5-2 illustrates the complete Kalman filter loop.

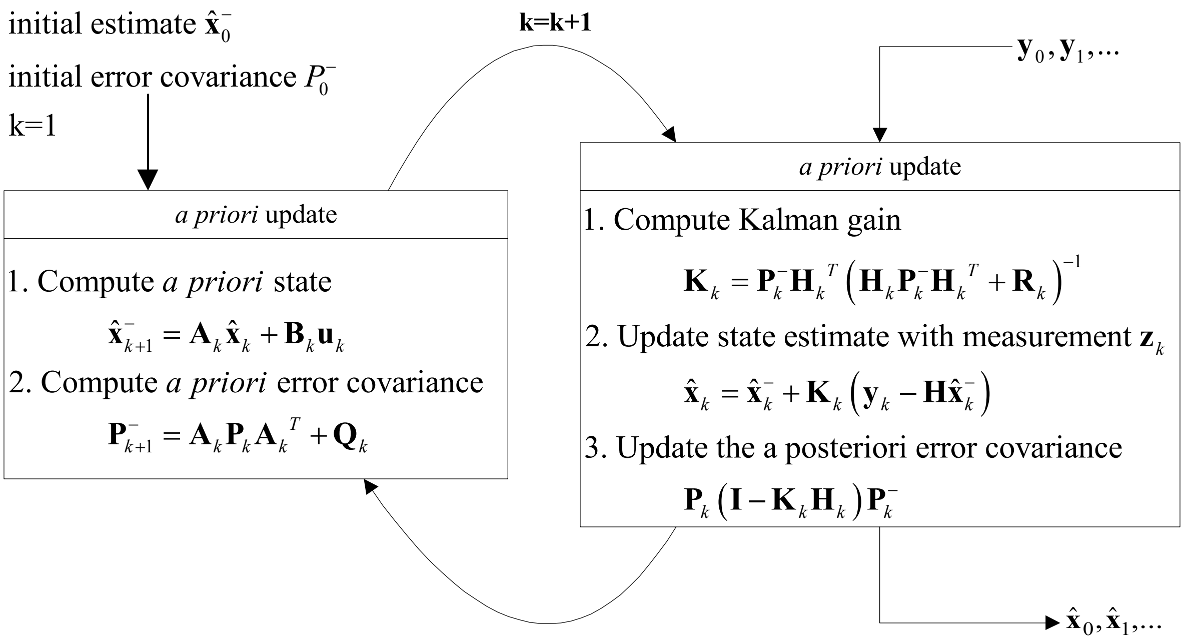

initial estimate $\hat{\mathbf{x}}_0^-$
initial error covariance P_0^-
k=1
k=k+1
$\mathbf{y}_0, \mathbf{y}_1, \ldots$
a priori update
1. Compute *a priori* state
$\hat{\mathbf{x}}_{k+1}^- = \mathbf{A}_k \hat{\mathbf{x}}_k + \mathbf{B}_k \mathbf{u}_k$
2. Compute *a priori* error covariance
$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{Q}_k$
a priori update
1. Compute Kalman gain
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k \right)^{-1}$
2. Update state estimate with measurement $\mathbf{z}_k$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{H} \hat{\mathbf{x}}_k^- \right)$
3. Update the a posteriori error covariance
$\mathbf{P}_k \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k \right) \mathbf{P}_k^-$
$\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots$

Figure 5-2 Linear Kalman Filter

It is important to note the information required in advance for filter implementation. **A, B, H, Q** and **R** must be known in advance in order to implement the Kalman filter. The control input **u** must be known during filter operation. This important restriction must be considered in selecting the optimal estimator for a particular problem.

5.2 Extended Kalman Filter (EKF)

The Kalman filter derivation in the previous section assumed the system and observation processes are linear with additive zero mean normally distributed noise. There are a great many systems where these assumptions are false. The mobile robot kinematic models developed in Chapter 3 are non-linear. The linear Kalman filter is not optimal or suitable for application in such non-linear cases.

However, the linear Kalman filter can be extended to the non-linear case, known as the extended Kalman filter (EKF). The extended Kalman filter (EKF) addresses this problem by linearizing the system about the current mean and covariance. This derivation follows the tutorial on Kalman filters given at SIGGRAPH 2001 [WB01]. More detailed derivations of the EKF can be found in [May82] and [Kay93].

The derivation of the EKF begins by replacing the linear system in equation (5-1) with a more general non-linear model

$$\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k\right) \tag{5-24}$$

Again, as in the previous section, the problem is to estimate the state, $\mathbf{x} \in \mathbb{R}^n$, of a dynamic system *f()* given measurements **y** where *h()* denotes the non-linear observation model.

$$\mathbf{y}_k = h\left(\mathbf{x}_k, \mathbf{v}_k\right) \tag{5-25}$$

$\mathbf{v}_\mathrm{k}$ and $\mathbf{w}_\mathrm{k}$ are the process and observation noise terms assumed as zero mean normally distributed random variables. Furthermore, the process and observation noise are assumed to be independent of current and past states,

$$\begin{aligned} \mathbf{w}_k &\sim N\left(\bar{\mathbf{w}}_k, \mathbf{Q}_k\right) \\ \mathbf{v}_k &\sim N\left(\bar{\mathbf{v}}, \mathbf{R}_k\right) \end{aligned} \tag{5-26}$$

We approximate the *a priori* state estimate by setting the process noise term to its mean in equation (5-24)

$$\tilde{\mathbf{x}}_{k+1} = f\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \bar{\mathrm{w}}_\mathrm{k}\right) \tag{5-27}$$

Likewise, the measurement is approximating by setting the measurement noise term to its mean equation (5-26)

$$\tilde{\mathbf{y}}_k = h\left(\hat{\mathbf{x}}_k, \bar{\mathbf{v}}_k\right) \tag{5-28}$$

where $\hat{\mathbf{x}}_k$ is the *a posteriori* state estimate. The critical aspect of EKF is to transform the non-linear process and observation models to linear models by linearizing equations (5-24) and (5-25) about the current state estimate. The linearization is achieved by the Taylor series expansion of the process and observation model functions. The inherent assumption is that the deviation of state between successive measurements is very small and hence the error between the linearized state estimate and true state is small. Recall that the Taylor series of a function *j(x)* about some point, $\mathbf{x} = \mathbf{x}_0$, is given by [Weiss99]

$$\begin{aligned} j(\mathbf{x}) &= j(\mathbf{x}_0) + \left| \frac{\partial j(\mathbf{x})}{\partial x} \right|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x}-\mathbf{x}_0) + \frac{1}{2!} \frac{\partial^2 j(\mathbf{x})}{\partial x^2} \bigg|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x}-\mathbf{x}_0)^2 + \cdots \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \frac{\partial^n j(\mathbf{x})}{\partial x^n} \bigg|_{\mathbf{x}=\mathbf{x}_0} (\mathbf{x}-\mathbf{x}_0)^n \end{aligned} \tag{5-29}$$

Taking the first and second terms of the Taylor series expansion of equation (5-24) gives the linearized process model as

$$\begin{aligned} \mathbf{x}_{k+1} &\approx f(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{w}_k) + \mathbf{A}_k (x_k - \hat{x}_k) + \mathbf{W}_k (\mathbf{w}_k - \bar{\mathbf{w}}_k) \\ &\approx \tilde{\mathbf{x}}_k + \mathbf{A}_k (x_k - \hat{x}_k) + \mathbf{W}_k (\mathbf{w}_k - \bar{\mathbf{w}}_k) \end{aligned} \tag{5-30}$$

where **A** denotes Jacobian matrix of partial derivatives of *f* with respect to the state **x**

$$\mathbf{A}_k = \frac{\partial f(\mathbf{x}, \mathbf{u}_k, \bar{\mathbf{w}}_k)}{\partial \mathbf{x}} \bigg|_{\mathbf{x}=\hat{\mathbf{x}}_k} \tag{5-31}$$

and **W** denotes Jacobian matrix of partial derivatives of *f* with respect to the noise **w**

$$\mathbf{W}_k = \frac{\partial f(\hat{\mathbf{x}}_k, \mathbf{u}_k, \mathbf{w})}{\partial \mathbf{w}} \bigg|_{\mathbf{w}=\bar{\mathbf{w}}_k} \tag{5-32}$$

Taking the first and second terms of the Taylor series expansion of equation (5-25) gives the linearized observation model as

$$\begin{aligned} \mathbf{y}_k &\approx h(\tilde{\mathbf{x}}_k, \bar{\mathbf{w}}_k) + \mathbf{H}_k (\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{V}_k (\mathbf{v}_k - \bar{\mathbf{v}}_k) \\ &\approx \tilde{\mathbf{y}}_k + \mathbf{H}_k (\mathbf{x}_k - \tilde{\mathbf{x}}_k) + \mathbf{V}_k (\mathbf{v}_k - \bar{\mathbf{v}}_k) \end{aligned} \tag{5-33}$$

where **H** denotes Jacobian matrix of partial derivatives of *h* with respect to the state **x**

$$\mathbf{H}_k = \left.\frac{\partial h\left(\mathbf{x}, \overline{\mathbf{v}}_k\right)}{\partial \mathbf{x}}\right|_{\mathbf{x}=\tilde{\mathbf{x}}_k} \tag{5-34}$$

and **V** denotes Jacobian matrix of partial derivatives of *h* with respect to the noise **v**

$$\mathbf{V}_k = \left.\frac{\partial h\left(\tilde{\mathbf{x}}_k, \mathbf{v}\right)}{\partial \mathbf{v}}\right|_{\mathbf{v}=\overline{\mathbf{v}}_k} \tag{5-35}$$

We now have linear process and observation models and can proceed to derive the extended Kalman filter as for the linear Kalman filter. Define the *a priori* state estimate error as

$$\tilde{\mathbf{e}}_{\mathbf{x}_k} = \mathbf{x}_k - \tilde{\mathbf{x}}_k \tag{5-36}$$

and the measurement residual error is

$$\tilde{\mathbf{e}}_{\mathbf{y}_k} = \mathbf{y}_k - \tilde{\mathbf{y}}_k \tag{5-37}$$

Substituting the error terms in equations (5-36) and (5-37) for the actual state in the process and observation models gives

$$\tilde{\mathbf{e}}_{\mathbf{x}_{k+1}} \approx \mathbf{A}_k\left(\mathbf{x}_k - \hat{\mathbf{x}}_k\right) + \mathbf{W}_k\left(\mathbf{w}_k - \overline{\mathbf{w}}_k\right) \tag{5-38}$$

$$\begin{aligned} \tilde{\mathbf{e}}_{\mathbf{y}_k} &\approx \mathbf{H}_k\left(\mathbf{x}_k - \tilde{\mathbf{x}}_k\right) + \mathbf{V}_k\left(\mathbf{v}_k - \overline{\mathbf{v}}_k\right) \\ &\approx \mathbf{H}_k\tilde{\mathbf{e}}_{\mathbf{x}_k} + \mathbf{V}_k\left(\mathbf{v}_k - \overline{\mathbf{v}}_k\right) \end{aligned} \tag{5-39}$$

The *a posteriori* state estimate is given by

$$\hat{\mathbf{x}}_k = \tilde{\mathbf{x}}_k + \hat{\mathbf{e}}_k \tag{5-40}$$

The filter equation used to estimate $\hat{\mathbf{e}}_k$ is given by

$$\hat{\mathbf{e}}_k = \mathbf{K}_k\tilde{\mathbf{e}}_{\mathbf{y}_k} \tag{5-41}$$

Substituting equations (5-37) and (5-40) into equation (5-41) yields the a posteriori state estimate in terms of the filter gain and measurement

$$\begin{aligned}\hat{\mathbf{x}}_k &= \tilde{\mathbf{x}} + \mathbf{K}_k \tilde{\mathbf{e}}_{\mathbf{y}_k} \\ &= \tilde{\mathbf{x}} + \mathbf{K}_k\left(\mathbf{y}_k - \tilde{\mathbf{y}}_k\right) \\ &= \tilde{\mathbf{x}} + \mathbf{K}_k\left(\mathbf{y}_k - h\left(\hat{\mathbf{x}}_k, \overline{\mathbf{v}}_k\right)\right)\end{aligned} \tag{5-42}$$

Applying the same derivation process as for the linear Kalman filter in section 5.1 yields the extended Kalman filter [WB01]. The *a priori* state prediction is given by equation (5-43)

$$\hat{\mathbf{x}}_{k+1}^- = f\left(\hat{\mathbf{x}}_k, \mathbf{u}_{k+1}, \overline{\mathbf{w}}_k\right) \tag{5-43}$$

The *a priori* state covariance prediction is given by equation (5-44). Note that the process Jacobian **A** and observation Jacobian **W** must be computed for each *a priori* update.

$$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T \tag{5-44}$$

The Kalman gain factor is given by

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R} \mathbf{V}_k^T\right)^{-1} \tag{5-45}$$

The a posteriori state estimate is given by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k\left(\mathbf{y}_k - \tilde{\mathbf{y}}_k\right) \tag{5-46}$$

And, the *a posteriori* state estimate covariance is given by

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^- \tag{5-47}$$

The complete extended Kalman filter algorithm is depicted in Figure 5-3.

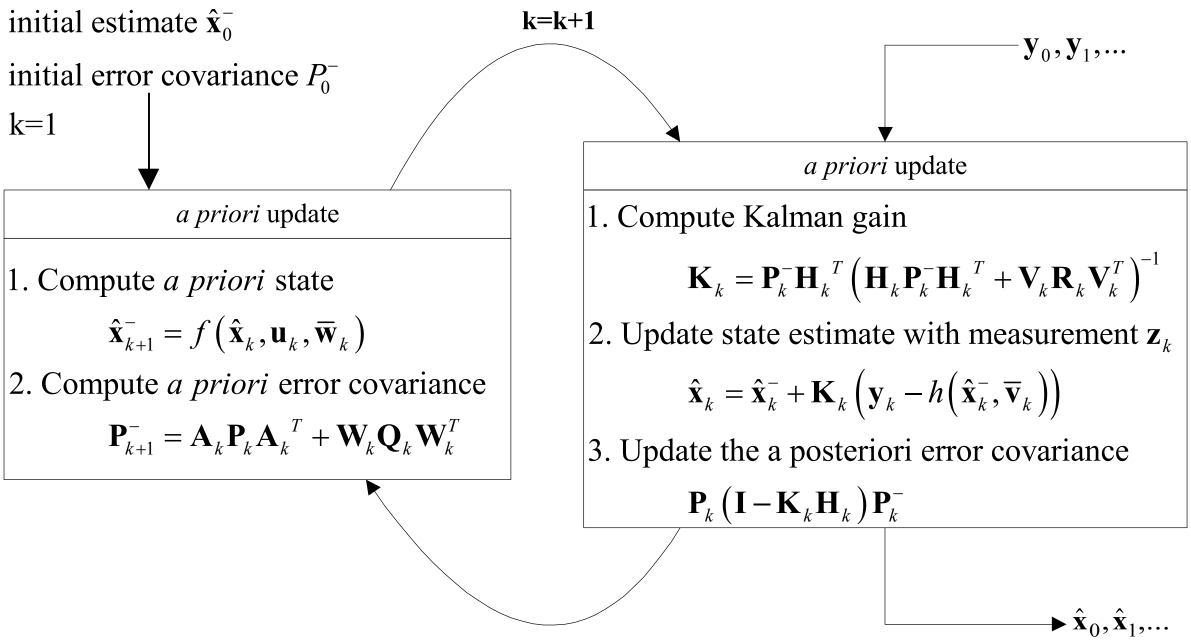

initial estimate $\hat{\mathbf{x}}_0^-$
initial error covariance P_0^-
k=1
k=k+1
$\mathbf{y}_0, \mathbf{y}_1, \ldots$
a priori update
1. Compute *a priori* state
$\hat{\mathbf{x}}_{k+1}^- = f\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k\right)$
2. Compute *a priori* error covariance
$\mathbf{P}_{k+1}^- = \mathbf{A}_k \mathbf{P}_k \mathbf{A}_k^{\ T} + \mathbf{W}_k \mathbf{Q}_k \mathbf{W}_k^T$
a priori update
1. Compute Kalman gain
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^{\ T} \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^{\ T} + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T\right)^{-1}$
2. Update state estimate with measurement $\mathbf{z}_k$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - h\left(\hat{\mathbf{x}}_k^-, \overline{\mathbf{v}}_k\right)\right)$
3. Update the a posteriori error covariance
$\mathbf{P}_k \left(\mathbf{I} - \mathbf{K}_k \mathbf{H}_k\right) \mathbf{P}_k^-$
$\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots$

Figure 5-3 Extended Kalman Filter

5.2.1 EKF Comments

The linear Kalman filter is the best estimator for a given linear system. The EKF allows estimation of non-linear systems by approximating the optimal linear Kalman filter. This approximation allows the EKF to be applied to a variety of problems not suited to the linear KF. For this reason, the EKF has found numerous applications such as target tracking, inertial navigation and system identification [BH96].

The penalty of the approximations in the EKF is a loss of optimality. For a given non-linear system, the EKF will not in general be the best estimator. Theoretically, there exists a non-linear estimator that will outperform an EKF for a given non-linear estimation problem. However, the EKF will be the best linear estimator of a non-linear problem with respect to the minimum mean-squared error.

The EKF does have limitations that require consideration. The linearization of the process and observation models is valid provided the estimate maintained by the filter is close to the true state of the system. If the linearized estimate deviates from the true state the filter will experience convergence problems. Furthermore, the linearization transforms the normally distributed process and measurement noise to other than normal distributions. This may result in biased estimates.

Another factor in considering the EKF is ease of implementation and use. The linear KF is very easy to use, as it only requires the state-space form of the process and observation models with associated noise covariance estimates and initial state. The EKF requires computation of the Jacobian of the process and observation models in addition to the elements required for the linear KF. Computation of the Jacobian is non-trivial for some non-linear problems [JU97].

5.3 Second Order Divided Difference Filter

Recently, a number of derivative-free forms of the EKF have been developed to address the limitations discussed above [MW01][JU97][NPR00a]. These include the *Unscented Kalman Filter* (UKF) [JUW95][JU97], the *Central Difference Filter* (CDF) [IX00], and the *Divided Difference Filter* (DDF) [NPR00a]. These filters consistently outperform the EKF for state estimation of non-linear systems [MW01]. The DDF is the simplest of the derivative-free estimators with similar computational complexity to the EKF. Thus, the divided difference filter is selected as the estimator used by the calibration algorithm proposed in this thesis.

The DD2 filter is derived in the following text. This derivation follows the original derivation by Noorgard [NPR00a]. For more detailed information about divided difference filters refer to [NPR00a] or [NPR00b]. The DD2 estimator is a formulation of the extended Kalman filter based on polynomial interpolation of non-linear functions. The DD2 filter uses Stirling's interpolation formula to approximate a non-linear function. Let the operators δ and μ denote difference operators over a selected interval of length *b*

$$\delta f(x) = f\left(x + \frac{b}{2}\right) - f\left(x - \frac{b}{2}\right) \tag{5-48}$$

$$\mu f(x) = \frac{1}{2}\left(f\left(x + \frac{b}{2}\right) + f\left(x - \frac{b}{2}\right)\right) \tag{5-49}$$

In terms of the above difference operators, Stirling's interpolation formula for a scalar function *f* around the point $x = x_0$ is [Weiss99]

$$\begin{aligned} f(x) &= f(x_0 + pb) \\ &= f(x_0) + p\mu\delta f(x_0) + \frac{p^2}{2!}\delta^2 f(x_0) + \binom{p+1}{3}\mu\delta^3 f(x_0) + \\ &\quad \frac{p^2(p^2-1)}{4!}\delta^4 f(x_0) + \binom{p+2}{5}\mu\delta^5 f(x_0) \end{aligned} \tag{5-50}$$

Define the first and second order central divided differences of the function *f*, $f_{DD}^{'}$ and $f_{DD}^{''}$ respectively as

$$\begin{aligned} f'_{DD}(x) &\triangleq \frac{f(x+b) - f(x-b)}{2b} \\ f''_{DD}(x) &\triangleq \frac{f(x+b) + f(x-b) - 2f(x)}{b^2} \end{aligned} \tag{5-51}$$

Substituting the divided difference definitions of equation (5-51) into the Sterling interpolation formula limited to first and second order terms gives

$$f(x) \approx f(x_0) + f_{DD}^{'}(x_0)(x - x_0) + \frac{f_{DD}^{''}(x_0)}{2!}(x - x_0)^2 \tag{5-52}$$

Note that equation (5-52) is equivalent to the Taylor series approximation of equation (5-29) with the derivatives replaced by central divided differences. The scalar form of the above interpolation formula is extended to the vector case by first introducing the first and second order divided difference operators $\tilde{D}_{\Delta x}, \tilde{D}_{\Delta x}^2$ for a vector function $\mathbf{y} = f(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^n$ about $\mathbf{x} = \mathbf{x}_0$

$$\begin{aligned}\tilde{D}_{\Delta\mathbf{x}}f &= \frac{1}{h}\left(\sum_{p=1}^{n}\Delta\mathbf{x}_p\mu_p\delta_p\right)f\left(\mathbf{x}_0\right)\\ \tilde{D}^2_{\Delta\mathbf{x}}f &= \frac{1}{h^2}\left(\sum_{p=1}^{n}\left(\Delta\mathbf{x}_p\right)^2\delta_p^2+\sum_{p=1}^{n}\sum_{q=1,q\neq p}^{n}\Delta\mathbf{x}_p\Delta\mathbf{x}_q\left(\mu_p\delta_p\right)\left(\mu_q\delta_q\right)\right)f\left(\mathbf{x}_0\right)\end{aligned} \tag{5-53}$$

where δ_p denotes the partial difference operator

$$\delta_p f\left(\mathbf{x}\right)=f\left(\mathbf{x}+\frac{b}{2}\mathbf{e}_p\right)-f\left(\mathbf{x}-\frac{b}{2}\mathbf{e}_p\right) \tag{5-54}$$

and e_p is the pth unit vector. Similarly, the partial average operator μ_p is given by

$$\mu f\left(\mathbf{x}\right)=\frac{1}{2}\left(f\left(\mathbf{x}+\frac{b}{2}\mathbf{e}_p\right)+f\left(\mathbf{x}-\frac{b}{2}\mathbf{e}_p\right)\right)$$

Then the second order divided difference approximation of the function is given by

$$\begin{aligned}y &= f\left(\mathbf{x}_0+\Delta\mathbf{x}\right)\\ &\approx f\left(\mathbf{x}_0\right)+\tilde{D}_{\Delta\mathbf{x}}f+\frac{1}{2!}\tilde{D}^2_{\Delta\mathbf{x}}f\end{aligned} \tag{5-55}$$

The next step in the derivation of the DD2 filter is to apply the preceding interpolation formula to the expectation and covariance of a random variable. Let **x** be a vector of random variables, of probability distribution $f_{\mathbf{x}}$, with expectation and covariance given by

$$\begin{aligned}\hat{\mathbf{x}} &= E\left[\mathbf{x}\right]=\int_{-\infty}^{+\infty}\mathbf{x}f_{\mathrm{X}}\left(x\right)dx\\ \mathbf{P}_{\mathbf{x}} &= E\left[\left(\mathbf{x}-\hat{\mathbf{x}}\right)\left(\mathbf{x}-\hat{\mathbf{x}}\right)^T\right]\end{aligned} \tag{5-56}$$

Let $\mathbf{y}=f\left(\mathbf{x}\right)$ be a non-linear function of **x**. The Kalman filter derivation requires determination of the following statistical moments

$$\hat{\mathbf{y}} = E\left[f(\mathbf{x})\right] \tag{5-57}$$

$$\mathbf{P}_{\hat{\mathbf{y}}} = E\left[\left(f(\mathbf{x}) - \hat{\mathbf{y}}\right)\left(f(\mathbf{x}) - \hat{\mathbf{y}}\right)^T\right] \tag{5-58}$$

$$\mathbf{P}_{\hat{\mathbf{x}}\hat{\mathbf{y}}} = E\left[\left(\mathbf{x} - \hat{\mathbf{x}}\right)\left(f(\mathbf{x}) - \hat{\mathbf{y}}\right)^T\right] \tag{5-59}$$

Calculation of the exact expectations is difficult since f is nonlinear. Similar to approach taken for the extended Kalman filter, the second-order polynomial approximation given by equation (5-55) is substituted into the preceding expectations. In addition, a linear transformation of **x** is introduced to decouple the member variables. The transformation matrix is selected as a square Cholesky factor of the covariance matrix

$$\mathbf{z} = \mathbf{S}_{\mathbf{x}}^{-1}\mathbf{x}, \qquad \mathbf{P}_{\mathbf{x}} = \mathbf{S}_{\mathbf{x}}\mathbf{S}_{\mathbf{x}}^T \tag{5-60}$$

This transformation provides a *stochastic decoupling* of the variables in **x** as the elements of z become mutually un-correlated, each with unity variance

$$E\left[\left(\mathbf{z} - E[\mathbf{z}]\right)\left(\mathbf{z} - E[\mathbf{z}]\right)^T\right] = \mathbf{I} \tag{5-61}$$

The approximated non-linear system becomes

$$\begin{aligned} y &\approx \tilde{f}(\bar{\mathbf{z}}) + \tilde{D}_{\Delta\mathbf{z}}\tilde{f} + \frac{1}{2}\tilde{D}_{\Delta\mathbf{z}}^2\tilde{f} \\ &= \tilde{f}(\bar{\mathbf{z}}) + \frac{1}{h}\left(\sum_{p=1}^{n}\Delta\mathbf{z}_p\mu_p\delta_p\right)\tilde{f}(\bar{\mathbf{z}}) \\ &+ \frac{1}{2b^2}\left(\sum_{p=1}^{n}\left(\Delta\mathbf{z}_p\right)^2\delta_p^2 + \sum_{p=1}^{n}\sum_{q=1,q\neq p}^{n}\Delta\mathbf{z}_p\Delta\mathbf{z}_q\left(\mu_p\delta_p\right)\left(\mu_q\delta_q\right)\right)\tilde{f}(\bar{\mathbf{z}}) \end{aligned} \tag{5-62}$$

It is assumed that Δz is Gaussian with zero mean and the elements are un-correlated. This assumption leads to an expression for the expectation of $\tilde{f}$ given by the estimate

$$
\begin{aligned}
\hat{\mathbf{y}} &= E\left[\tilde{f}\left(\bar{\mathbf{z}}\right)+\frac{1}{2}\left(\sum_{p=1}^{n}\left(\Delta\mathbf{z}_p\right)^2\delta_p^2\right)\tilde{f}\left(\bar{\mathbf{z}}\right)\right] \\
&= \tilde{f}\left(\bar{\mathbf{z}}\right)+\frac{\sigma_2}{2}\sum_{p=1}^{n}\delta_p^2\tilde{f}\left(\bar{\mathbf{z}}\right) \\
&= \tilde{f}\left(\bar{\mathbf{z}}\right)+\frac{1}{2b^2}\sum_{p=1}^{n}\left(\tilde{f}\left(\bar{\mathbf{z}}+b\mathbf{e}_p\right)+\tilde{f}\left(\bar{\mathbf{z}}-b\mathbf{e}_p\right)\right)-\frac{n}{b^2}\tilde{f}\left(\bar{\mathbf{z}}\right) \\
&= \frac{b^2-n}{b^2}\tilde{f}\left(\bar{\mathbf{z}}\right)+\frac{1}{2b^2}\sum_{p=1}^{n}\left(\tilde{f}\left(\bar{\mathbf{z}}+b\mathbf{e}_p\right)+\tilde{f}\left(\bar{\mathbf{z}}-b\mathbf{e}_p\right)\right)
\end{aligned}
\tag{5-63}
$$

$$\Updownarrow$$

$$
\hat{\mathbf{y}} = \frac{b^2-n}{b^2}f\left(\hat{\mathbf{x}}\right)+\frac{1}{2b^2}\sum_{p=1}^{n}f\left(\hat{\mathbf{x}}+b\mathbf{s}_{x,p}\right)+f\left(\hat{\mathbf{x}}-b\mathbf{s}_{x,p}\right)
\tag{5-64}
$$

Note that $\tilde{f}\left(\bar{\mathbf{z}}\pm b\mathbf{e}_p\right)=f\left(\bar{\mathbf{x}}\pm b\mathbf{s}_{\mathbf{x},p}\right)$, where $\mathbf{s}_{\mathbf{x},p}$ is the pth column of the square Cholesky factor of the covariance matrix S_x, (5-60). The covariance estimate is given by

$$
\begin{aligned}
P_{\hat{\mathbf{y}}} &= E\left[\left(y-\hat{\mathbf{y}}\right)\left(y-\hat{\mathbf{y}}\right)^T\right] \\
&= E\left[\left(y-\tilde{f}\left(\bar{z}\right)\right)\left(y-\tilde{f}\left(\bar{z}\right)\right)^T\right]-E\left[y-\tilde{f}\left(\bar{z}\right)\right]E\left[y-\tilde{f}\left(\bar{z}\right)\right]^T
\end{aligned}
\tag{5-65}
$$

Substituting in the estimate (5-64) in terms of the divided difference operators gives the covariance as

$$
\begin{aligned}
P_y &= E\left[\left(\tilde{D}_{\Delta z}\tilde{f}+\frac{1}{2}\tilde{D}_{\Delta z}^2\tilde{f}\right)\left(\tilde{D}_{\Delta z}\tilde{f}+\frac{1}{2}\tilde{D}_{\Delta z}^2\tilde{f}\right)^T\right] \\
&\quad -E\left[\tilde{D}_{\Delta z}\tilde{f}+\frac{1}{2}\tilde{D}_{\Delta z}^2\tilde{f}\right]E\left[\tilde{D}_{\Delta z}\tilde{f}+\frac{1}{2}\tilde{D}_{\Delta z}^2\tilde{f}\right]^T \\
&= E\left[\tilde{D}_{\Delta z}\tilde{f}\left(\tilde{D}_{\Delta z}\tilde{f}\right)^T\right]+\frac{1}{4}E\left[\tilde{D}_{\Delta z}^2\tilde{f}\left(\tilde{D}_{\Delta z}^2\tilde{f}\right)^T\right] \\
&\quad -\frac{1}{4}E\left[\tilde{D}_{\Delta z}^2\tilde{f}\right]E\left[\tilde{D}_{\Delta z}^2\tilde{f}\right]^T
\end{aligned}
\tag{5-66}
$$

Assuming the elements of $\Delta\mathbf{z}$ are independent with symmetric distribution, the odd order moments of equation (5-66) cancel and the covariance estimate becomes

$$
\begin{aligned}
\mathbf{P}_y &= \sigma_2 \sum_{p=1}^{n} \left(\mu_p \delta_p \tilde{f}\left(\overline{z}\right)\right)\left(\mu_p \delta_p \tilde{f}\left(\overline{z}\right)\right)^T + \frac{\sigma_4 - \sigma_2^2}{4} \sum_{p=1}^{n} \left(\delta_p^2 \tilde{f}\left(\overline{z}\right)\right)\left(\delta_p^2 \tilde{f}\left(\overline{z}\right)\right)^T \\
&= \frac{\sigma_2}{4b^2} \sum_{p=1}^{n} \left(\tilde{f}\left(\overline{z} + be_p\right) - \tilde{f}\left(\overline{z} - be_p\right)\right)\left(\tilde{f}\left(\overline{z} + be_p\right) - \tilde{f}\left(\overline{z} - be_p\right)\right)^T \\
&\quad + \frac{\sigma_4 - \sigma_2^2}{4} \sum_{p=1}^{n} \ \left(\tilde{f}\left(\overline{z} + be_p\right) + \tilde{f}\left(\overline{z} - be_p\right) - 2\tilde{f}\left(\overline{z}\right)\right) \times \\
&\quad \left(\tilde{f}\left(\overline{z} + be_p\right) + \tilde{f}\left(\overline{z} - be_p\right) - 2\tilde{f}\left(\overline{z}\right)\right)^T
\end{aligned}
\tag{5-67}
$$

Assuming $\sigma_2 = 1$ and setting $h^2 = \sigma_4$ gives the covariance estimate as

$$
\begin{aligned}
P_y &= \frac{1}{4b^2} \sum_{p=1}^{n} \left(\tilde{f}\left(\hat{\mathbf{x}} + bs_{x,p}\right) - \tilde{f}\left(\hat{\mathbf{x}} - bs_{x,p}\right)\right)\left(\tilde{f}\left(\hat{\mathbf{x}} + bs_{x,p}\right) - \tilde{f}\left(\hat{\mathbf{x}} - bs_{x,p}\right)\right)^T \\
&\quad + \frac{b^2 - 1}{4b^4} \sum_{p=1}^{n} \ \left(\tilde{f}\left(\hat{\mathbf{x}} + bs_{x,p}\right) + \tilde{f}\left(\hat{\mathbf{x}} - bs_{x,p}\right) - 2\tilde{f}\left(\hat{\mathbf{x}}\right)\right) \times \\
&\quad \left(\tilde{f}\left(\hat{\mathbf{x}} + bs_{x,p}\right) + \tilde{f}\left(\hat{\mathbf{x}} - bs_{x,p}\right) - 2\tilde{f}\left(\hat{\mathbf{x}}\right)\right)^T
\end{aligned}
\tag{5-68}
$$

As

$$
\sigma_4 - \sigma_2^2 = E\left[\left(\Delta z\right)^4\right] - E\left[\left(\Delta z\right)^2\right]^2 = \operatorname{var}\left[\left(\Delta z\right)^2\right] > 0 \tag{5-69}
$$

$\sigma_4 \geq \sigma_2^2$ for all probability distributions. Therefore, the interval b should always selected such that $b^2 \geq 1$. This implies that the covariance estimate will always be positive semi-definite.

The cross-covariance estimate, P_{xy} is given by

$$
\begin{aligned}
\mathbf{P}_{xy} &= E\left[\left(\mathbf{x}_k-\hat{\mathbf{x}}_k\right)\left(\mathbf{y}_k-\hat{\mathbf{y}}_k\right)^T \middle| \mathbf{Y}^{k\text{-}1}\right] \\
&= E\left[\left(S_x\Delta z\right)\left(\tilde{D}_{\Delta z}\tilde{f}+\frac{1}{2}\tilde{D}^2_{\Delta z}\tilde{f}\right)^2\right] \\
&= E\left[\left(S_x\Delta z\right)\left(\tilde{D}_{\Delta z}\tilde{f}\right)^2\right] \\
&= \frac{1}{2h}\sum_{p=1}^{n} s_{x,p}\left(f\left(\overline{x}+hs_{x,p}\right)-f\left(\overline{x}-hs_{x,p}\right)\right)^T
\end{aligned}
\tag{5-70}
$$

where $\mathbf{Y}_{k-1}=\left[\mathbf{y}_0 \quad \mathbf{y}_1 \quad \cdots \quad \mathbf{y}_{k-1}\right]$ denotes the matrix of past measurements.

The derivation of the DD2 filter follows EKF derivation with the Jacobians (5-30) replaced by the divided differences. Let $\mathbf{x}\in\mathbb{R}^n$ denote the state of a dynamic system (process) given by the non-linear difference equation

$$
\mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k\right) \tag{5-71}
$$

The problem is to estimate the state of the system given measurements **y** where *h()* denotes the non-linear observation model

$$
\mathbf{y}_k = h\left(\mathbf{x}_k, \mathbf{v}_k\right) \tag{5-72}
$$

$\mathbf{v}_k$ and $\mathbf{w}_k$ are the process and observation noise terms assumed as normally distributed random variables. The process and observation noise are assumed independent of state,

$$
\begin{aligned}
&\mathbf{w}_k \sim N\left(\overline{\mathbf{w}}_k,\mathbf{Q}_k\right) && \overline{\mathbf{w}}_k = E\left[\mathbf{w}_k\right] && \mathbf{Q}_k = E\left[\left(\mathbf{w}_k-\overline{\mathbf{w}}_k\right)\left(\mathbf{w}_k-\overline{\mathbf{w}}_k\right)^T\right] \\
&\mathbf{v}_k \sim N\left(,\mathbf{R}_k\right) && \overline{\mathbf{v}}_k = E\left[\mathbf{v}_k\right] && \mathbf{R}_k = E\left[\left(\mathbf{v}_k-\overline{\mathbf{v}}_k\right)\left(\mathbf{v}_k-\overline{\mathbf{v}}_k\right)^T\right]
\end{aligned}
\tag{5-73}
$$

One of the most useful aspects of the algorithm proposed by Norgaard [NPR00] is the use of Cholesky factors to represent the covariance matrices. Recall that **S** is a *Cholesky factor* for any symmetric matrix **M** if $\mathbf{S}^T\mathbf{S}=\mathbf{M}$. Now define the covariance matrices used in the EKF formulation in terms of their Cholesky factorizations

$$\begin{aligned}
\mathbf{Q}_k &= \mathbf{S}_{\mathrm{w_k}}^{\mathrm{T}}\mathbf{S}_{\mathbf{w}_k} \\
\mathbf{R}_k &= \mathbf{S}_{\mathbf{v}_k}^{T}\mathbf{S}_{\mathbf{v}_k} \\
\mathbf{P}_k^- &= \mathbf{S}_{\mathbf{x}_k}^{-T}\mathbf{S}_{\mathbf{x}_k}^- \\
\mathbf{P}_k &= \mathbf{S}_{\mathbf{x}_k}^{T}\mathbf{S}_{\mathbf{x}_k}
\end{aligned} \tag{5-74}$$

where $\mathbf{P}_k^-$ is the *a priori* error covariance and $\mathbf{P}_k$ is the *a posteriori* error covariance. Let the jth column of the covariance Cholesky factors $\mathbf{S}_{\mathbf{w}_k}, \mathbf{S}_{\mathbf{v}_k}, \mathbf{S}_{\mathbf{x}_k}^-, \mathbf{S}_{\mathbf{x}_k}$ be denoted by $s_{\mathbf{x},j}$. The error covariance Cholesky factors are approximated in terms of thee first and second order divided differences as follows

$$\begin{aligned}
&\mathbf{S}_{\mathbf{x}\hat{\mathbf{x}}}^{(1)}(k) = \left\{\mathbf{S}_{\mathbf{x}\hat{\mathbf{x}}}^{(1)}(i,j)\right\} = \left\{\left(f_i\left(\hat{\mathbf{x}}_k + bs_{x,j}^-, \mathbf{u}_k, \overline{\mathbf{w}}_k\right) - f_i\left(\hat{\mathbf{x}}_k - bs_{x,j}^-, \mathbf{u}_k, \overline{\mathbf{w}}_k\right)\right)\big/2b\right\} \\
&\mathbf{S}_{\mathbf{xw}}^{(1)}(k) = \left\{\mathbf{S}_{\mathbf{xw}}^{(1)}(i,j)\right\} = \left\{\left(f_i\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k + bs_{w,j}\right) - f_i\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k - bs_{w,j}\right)\right)\big/2b\right\} \\
&\mathbf{S}_{\mathbf{y}\hat{\mathbf{x}}^-}^{(1)}(k) = \left\{S_{\mathbf{y}\hat{\mathbf{x}}^-}^{(1)}(i,j)\right\} = \left\{\left(h_i\left(\hat{\mathbf{x}}_k^- + b\hat{s}_{x,j}^-, \overline{\mathbf{v}}_k\right) - h_i\left(\hat{\mathbf{x}}_k^- - b\hat{s}_{x,j}^-, \overline{\mathbf{v}}_k\right)\right)\big/2b\right\} \\
&\mathbf{S}_{\mathbf{yv}}^{(1)}(k) = \left\{\mathbf{S}_{\mathbf{yv}}^{(1)}(i,j)\right\} = \left\{\left(h_i\left(\hat{\mathbf{x}}_k^-, \overline{\mathbf{v}}_k + bs_{v,j}\right) - h_i\left(\hat{\mathbf{x}}_k^-, \overline{\mathbf{v}}_k - bs_{v,j}\right)\right)\big/2b\right\} \\
&\mathbf{S}_{\mathbf{x}\hat{\mathbf{x}}^-}^{(2)}(k) = \left\{\frac{\sqrt{b^2-1}}{2b^2}\left(f_i\left(\hat{\mathbf{x}}_k + b\hat{s}_{x,j}, \mathbf{u}_k, \overline{\mathbf{w}}_k\right) + f_i\left(\hat{\mathbf{x}}_k - b\hat{s}_{x,j}, \mathbf{u}_k, \overline{\mathbf{w}}_k\right) - 2f_i\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k\right)\right)\right\} \\
&\mathbf{S}_{\mathbf{xw}}^{(2)}(k) = \left\{\frac{\sqrt{b^2-1}}{2b^2}\left(f_i\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k + bs_{v,j}\right) + f_i\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k - bs_{v,j}\right) - 2f_i\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k\right)\right)\right\} \\
&\mathbf{S}_{\mathbf{y}\hat{\mathbf{x}}}^{(2)}(k) = \left\{\frac{\sqrt{b^2-1}}{2b^2}\left(h_i\left(\hat{\mathbf{x}}_k^- + b\overline{s}_{x,j}, \overline{\mathbf{v}}_k\right) + h_i\left(\hat{\mathbf{x}}_k^- - b\overline{s}_{x,j}, \overline{\mathbf{v}}_k\right) - h_i\left(\hat{\mathbf{x}}_k^-, \overline{\mathbf{v}}_k\right)\right)\right\} \\
&\mathbf{S}_{yv}^{(2)}(k) = \left\{\frac{\sqrt{b^2-1}}{2b^2}\left(h_i\left(\hat{\mathbf{x}}_k, \overline{\mathbf{v}}_k + bs_{w,j}\right) + h_i\left(\hat{\mathbf{x}}_k, \overline{\mathbf{v}}_k - bs_{w,j}\right) - 2h_i\left(\hat{\mathbf{x}}_k, \overline{\mathbf{v}}_k\right)\right)\right\}
\end{aligned} \tag{5-75}$$

The a priori state estimate is given by substituting the previous covariance Cholesky factors (5-75) into the second order process approximation (5-64). The result is given by

$$
\begin{aligned}
\hat{\mathbf{x}}_{k+1}^{-} = {} & \frac{b^2 - n_x - n_w}{b^2} f\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k\right) \\
& + \frac{1}{2b^2} \sum_{p=1}^{n_x} f\left(\hat{\mathbf{x}}_k + b\hat{s}_{x,p}^{-}, \mathbf{u}_k, \overline{\mathbf{w}}_k\right) + f\left(\hat{\mathbf{x}}_k - b\hat{s}_{x,p}^{-}, \mathbf{u}_k, \overline{\mathbf{w}}_k\right) \\
& + \frac{1}{2b^2} \sum_{p=1}^{n_w} f\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k + bs_{w,p}\right) + f\left(\hat{\mathbf{x}}_k, \mathbf{u}_k, \overline{\mathbf{w}}_k - bs_{w,p}\right)
\end{aligned}
\tag{5-76}
$$

where n_x denotes the dimension of the state vector and n_w denotes the dimension of process noise vector.

The *a priori* covariance estimate is determined by substituting the covariance Cholesky factors (5-75) into equation (5-68)

$$
\mathbf{P}_{k+1}^{-} = \mathbf{S}_{\hat{\mathbf{x}}_{k+1}}^{-} \mathbf{S}_{\hat{\mathbf{x}}_{k+1}}^{-T}
\tag{5-77}
$$

The resulting Cholesky factors in equation (5-77) are rectangular. Transforming them to square matrices is required for further evaluation. The triangular Cholesky factor of the *a priori* covariance (5-77) is obtained by Householder transformation of the compound matrix

$$
\mathbf{S}_{\mathbf{x}_{k+1}}^{-} = \begin{bmatrix} \mathbf{S}_{\mathbf{x}_k \hat{\mathbf{x}}_k}^{(1)} & \mathbf{S}_{\mathbf{x}_k \mathbf{w}_k}^{(1)} & S_{\mathbf{x}_k \hat{\mathbf{x}}_k}^{(2)} & S_{\mathbf{x}_k \mathbf{w}_k}^{(2)} \end{bmatrix}
\tag{5-78}
$$

The *a priori* estimate of the measurement and its covariance are calculated as follows

$$
\begin{aligned}
y_k^{-} = {} & \frac{b^2 - n_x - n_v}{b^2} h\left(\mathbf{x}_k^{-}, \overline{\mathbf{v}}_k\right) \\
& + \frac{1}{2b^2} \sum_{p=1}^{n_x} h\left(\mathbf{x}_k^{-} + bs_{\mathbf{x},p}^{-}, \overline{\mathbf{v}}_k\right) + h\left(\mathbf{x}_k^{-} - bs_{\mathbf{x},p}^{-}, \overline{\mathbf{v}}_k\right) \\
& + \frac{1}{2b^2} \sum_{p=1}^{n_v} h\left(\mathbf{x}_k^{-}, \overline{\mathbf{v}}_k + bs_{v,p}^{-}\right) + h\left(\mathbf{x}_k^{-}, \overline{\mathbf{v}}_k - bs_{v,p}^{-}\right)
\end{aligned}
\tag{5-79}
$$

The Cholesky factor of the measurement covariance is given by

$$\mathbf{S}_{\mathbf{y}_k} = \left[\mathbf{S}^{(1)}_{\mathbf{y}\hat{\mathbf{x}}_k^-} \quad \mathbf{S}^{(1)}_{\mathbf{y}_k\mathbf{v}_k} \quad \mathbf{S}^{(2)}_{\mathbf{y}\hat{\mathbf{x}}_k^-} \quad \mathbf{S}^{(2)}_{\mathbf{y}_k\mathbf{v}_k} \right] \tag{5-80}$$

where n_v denotes the dimension of the measurement noise vector. The *a priori* cross-covariance matrix is given by substitution of the Cholesky factors (5-75) into equation (5-70)

$$\mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} = \mathbf{S}^-_{\mathbf{x}_k} \left(\mathbf{S}^-_{\mathbf{y}\hat{\mathbf{x}}_k^-} \right)^T \tag{5-81}$$

The Kalman gain is given by the conditional expectation

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} \left(\mathbf{P}_{\mathbf{y}_k} \right)^{-1} \tag{5-82}$$

Substituting the covariance Cholesky factors into equation (5-82) gives the Kalman gain as

$$\mathbf{K}_k = \mathbf{P}_{\mathbf{x}_k\mathbf{y}_k} \left[\mathbf{S}_{\mathbf{y}_k} \mathbf{S}^T_{\mathbf{y}_k} \right]^{-1} \tag{5-83}$$

As for the extended Kalman filter, the DD2 *a posteriori* update of the state is given by

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{y}_k^- \right) \tag{5-84}$$

The *a posteriori* update of the estimation error covariance is given by

$$\begin{aligned}
\hat{\mathbf{P}}_k &= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{\mathbf{y}_k} \mathbf{K}_k^T \\
&= \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{\mathbf{y}_k} \mathbf{K}_k^T - \mathbf{K}_k \mathbf{P}_{\mathbf{y}_k} \mathbf{K}_k^T + \mathbf{K}_k \mathbf{P}_{\mathbf{y}_k} \mathbf{K}_k^T \\
&= \left(\mathbf{S}^-_{\mathbf{x}_k} - \mathbf{K}_k \mathbf{S}^{(1)}_{\mathbf{y}_k\mathbf{x}_k} \right) \left(\mathbf{S}^-_{\mathbf{x}_k} - \mathbf{K}_k \mathbf{S}^{(1)}_{\mathbf{y}_k\mathbf{x}_k} \right)^T + \mathbf{K}_k \mathbf{S}^{(1)}_{\mathbf{y}_k\mathbf{v}_k} \left(\mathbf{K}_k \mathbf{S}^{(1)}_{\mathbf{y}_k\mathbf{v}_k} \right)^T \\
&\quad + \mathbf{K}_k \mathbf{S}^{(2)}_{\mathbf{y}_k\mathbf{x}_k} \left(\mathbf{K}_k \mathbf{S}^{(2)}_{\mathbf{y}_k\mathbf{x}_k} \right)^T + \mathbf{K}_k \mathbf{S}^{(2)}_{\mathbf{y}_k\mathbf{w}_k} \left(\mathbf{K}_k \mathbf{S}^{(2)}_{\mathbf{y}_k\mathbf{w}_k} \right)^T
\end{aligned} \tag{5-85}$$

The *a posteriori* error covariance has a Cholesky factor given by

$$\hat{S}_x(k) = \left[\bar{S}_x(k) - K_k S_{yx}^{(1)}(x) \quad K_k S_{yw}^{(1)}(k) \quad K_k S_{yx}^{(2)}(k) \quad K_k S_{yw}^{(2)}(k) \right] \tag{5-86}$$

5.3.1 DD2 Filter Algorithm

The DD2 filter algorithm follows the EKF algorithm with *a priori* and *a posteriori* estimates computed recursively for each state measurement. Using the previous definitions, the DD2 algorithm is described by the following 9-step procedure a given in [NPR00]. Note, that the process and measurement noise are assumed to be normally distributed random variable and hence $h^2 = 3$ since $\sigma_4 = 3\sigma_2$.

Step 1 - Select the initial state estimate and covariance matrix $\hat{\mathbf{x}}_0, \hat{\mathbf{P}}_{\mathbf{x}_0}, k = 0$

Step 2 - Compute $\hat{\mathbf{y}}_k, \mathbf{S}^{(1)}_{\mathbf{y}_k \hat{\mathbf{x}}_k}, \mathbf{S}^{(1)}_{\mathbf{y}_k \mathbf{v}_k}, \mathbf{S}^{(2)}_{\mathbf{y}_k \hat{\mathbf{x}}_k}, \mathbf{S}^{(2)}_{\mathbf{y}_k \mathbf{v}_k}$

Step 3 - Compute P_{xy} according to (5-77) and determine $\mathbf{S}_{\mathbf{y}_k}$ using Householder triangularization on (5-80).

Step 4 - Compute the Kalman gain by solving $\mathbf{K}_k \left[\mathbf{S}^-_{\mathbf{y}_k} \mathbf{S}_{\mathbf{y}_k}{}^T \right] = \mathbf{P}_{xy}$. Since S_y is square and triangular only forward and back-substitutions are needed.

Step 5 - Compute the *a posteriori* update of the state estimate $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}^-_k + \mathbf{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_k \right)$

Step 6 - Compute the *a posteriori* update of covariance matrix factor, $\hat{S}_x(k)$ by taking the Householder triangularization of (5-86).

Step 7 - Determine the a priori state estimate and covariance Cholesky factors $\hat{\mathbf{x}}^-_{k+1}, \mathbf{S}^{(1)}_{\mathbf{x}_k \hat{\mathbf{x}}_k}, \mathbf{S}^{(1)}_{\mathbf{x}_k \mathbf{v}_k}, \mathbf{S}^{(2)}_{\mathbf{x}_k \hat{\mathbf{x}}_k}, \mathbf{S}^{(2)}_{\mathbf{x}_k \mathbf{v}_k}$.

Step 8 - Compute $\bar{S}_{\mathbf{x}_k}$ by taking the Householder triangularization of (5-78)

Step 9 - Increment k, k = k + 1, go to step 2

The DD2 filter algorithm is graphically depicted in Figure 5-4.

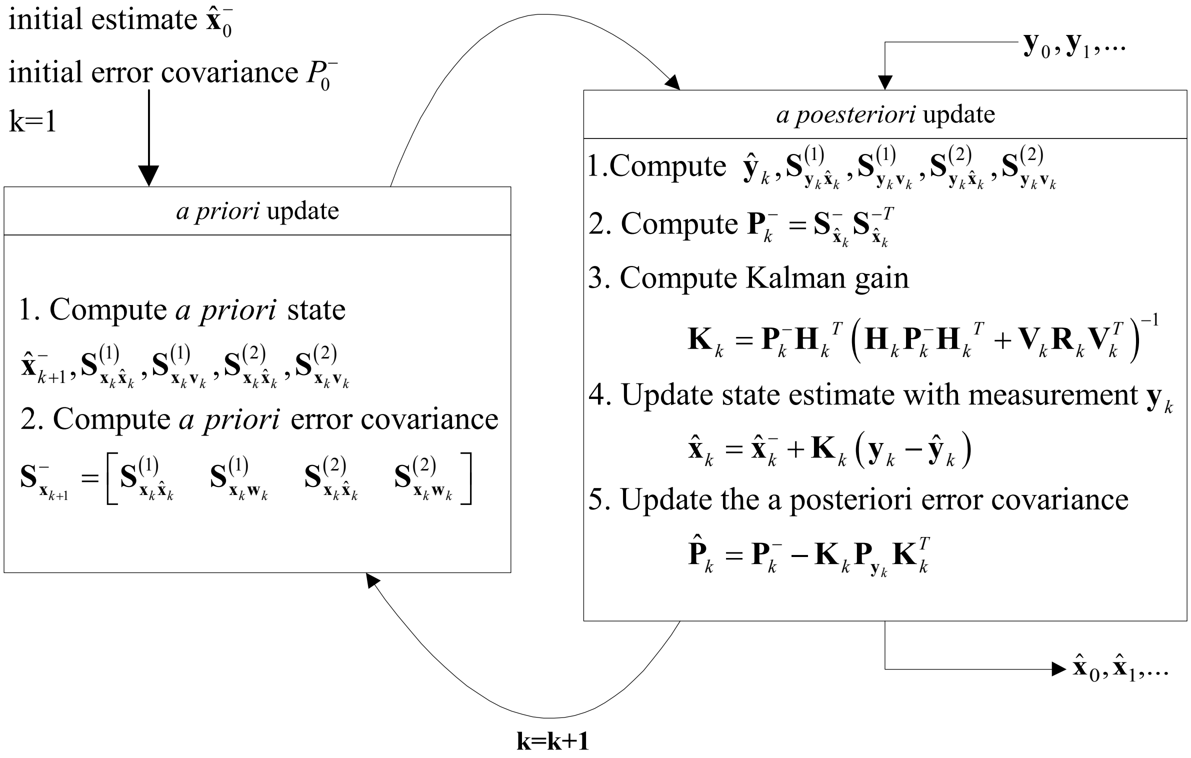

initial estimate $\hat{\mathbf{x}}_0^-$
initial error covariance P_0^-
k=1
$\mathbf{y}_0, \mathbf{y}_1, \ldots$
a poesteriori update
1.Compute $\hat{\mathbf{y}}_k, \mathbf{S}_{\mathbf{y}_k\hat{\mathbf{x}}_k}^{(1)}, \mathbf{S}_{\mathbf{y}_k\mathbf{v}_k}^{(1)}, \mathbf{S}_{\mathbf{y}_k\hat{\mathbf{x}}_k}^{(2)}, \mathbf{S}_{\mathbf{y}_k\mathbf{v}_k}^{(2)}$
2. Compute $\mathbf{P}_k^- = \mathbf{S}_{\hat{\mathbf{x}}_k}^- \mathbf{S}_{\hat{\mathbf{x}}_k}^{-T}$
3. Compute Kalman gain
$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T \left(\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{V}_k \mathbf{R}_k \mathbf{V}_k^T \right)^{-1}$
4. Update state estimate with measurement $\mathbf{y}_k$
$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k \left(\mathbf{y}_k - \hat{\mathbf{y}}_k \right)$
5. Update the a posteriori error covariance
$\hat{\mathbf{P}}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{P}_{\mathbf{y}_k} \mathbf{K}_k^T$
a priori update
1. Compute *a priori* state
$\hat{\mathbf{x}}_{k+1}^-, \mathbf{S}_{\mathbf{x}_k\hat{\mathbf{x}}_k}^{(1)}, \mathbf{S}_{\mathbf{x}_k\mathbf{v}_k}^{(1)}, \mathbf{S}_{\mathbf{x}_k\hat{\mathbf{x}}_k}^{(2)}, \mathbf{S}_{\mathbf{x}_k\mathbf{v}_k}^{(2)}$
2. Compute *a priori* error covariance
$\mathbf{S}_{\mathbf{x}_{k+1}}^- = \left[\mathbf{S}_{\mathbf{x}_k\hat{\mathbf{x}}_k}^{(1)} \quad \mathbf{S}_{\mathbf{x}_k\mathbf{w}_k}^{(1)} \quad \mathbf{S}_{\mathbf{x}_k\hat{\mathbf{x}}_k}^{(2)} \quad \mathbf{S}_{\mathbf{x}_k\mathbf{w}_k}^{(2)} \right]$
$\hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1, \ldots$
k=k+1

Figure 5-4 DD2 Filter Algorithm

6. Mobile Robot Calibration Results

This chapter presents results of robot model calibration experiments with a differential wheeled mobile robot. The calibration apparatus is described. The calibration procedure is outlined. The results of 99 calibration experiments are presented for a differential wheeled mobile robot.

6.1 Experimental Apparatus

The calibration apparatus is a photogrammetric system consisting of a charge coupled device (CCD) digital camera interfaced to a desktop computer. The camera is mounted on a tripod approximately 1,500 mm above the calibration workspace. The calibration workspace is a 1,200 mm by 1,200 mm flat painted wood fiber plane. **Figure 6-1** depicts the calibration apparatus.

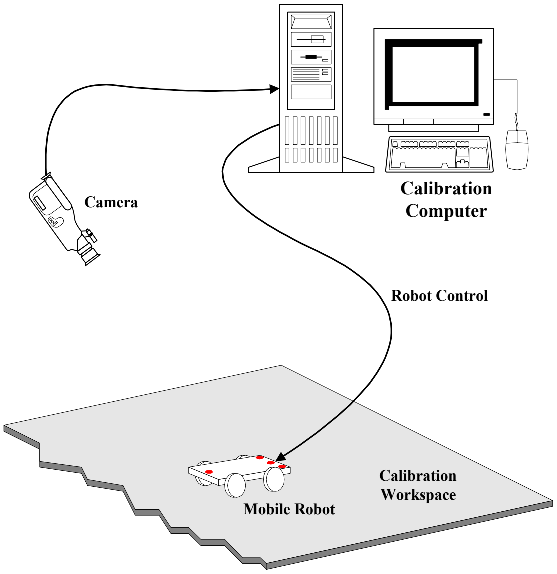


Figure 6-1 Photogrammetric Calibration Apparatus

A photogrammetric target is mounted on the robot to ensure unambiguous estimation of robot pose using Lowe's algorithm. **Figure 6-2** is a full scale drawing of the calibration target. The target measures 80 mm in width by 120 mm in length. The target consists of 4 fiducial markers identical to themarkers used on the camera calibration target as shown in Chapter 4. This allows the same fiducial marker detection algorithm to be used for both camera calibration and robot pose estimation. Each of the target markers is assigned a unique label (e.g. M1, M2, M3 and M4).

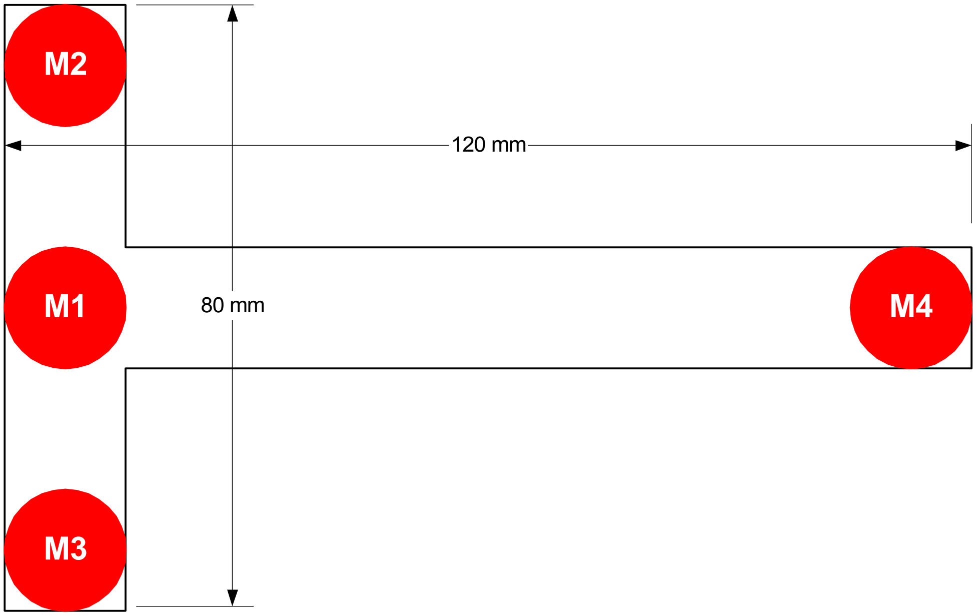


Figure 6-2 Photogrammetric Calibration Target (Scale 1:1)

6.1.1 Calibration Computer

The calibration computer performs robot control, image acquisition of the camera video stream and processing of the calibration algorithm. The computer is a Pentium 233 MHz desktop computer with an ASUS TX97 motherboard. The computer is equipped with 128 Mbytes of RAM and 13 GBytes of hard disk drive space. The operating system is Windows 98. All calibration algorithms are implemented in MATLAB.

The robot under calibration interfaces to the calibration computer by an RS-232C serial interface. The robot can not be controlled in real-time by the calibration computer due to limitations of the Windows operating system. A predetermined control vector is loaded from the PC onto the robot microprocessor for control during the calibration experiment.

6.1.2 Calibration Software

The calibration software is a suite of MATLAB programs loaded on the calibration computer. The programs are executed in the MATLAB environment.

Measured and processed data are stored in data files on the calibration computer hard disk drive. **Figure 6-3** illustrates the architecture of key calibration software programs. The cameraCal.m program acquires images from the camera and applies the camera calibration algorithm described in Chapter 4. The resulting camera model parameters are saved to a data file for subsequent use by other programs.

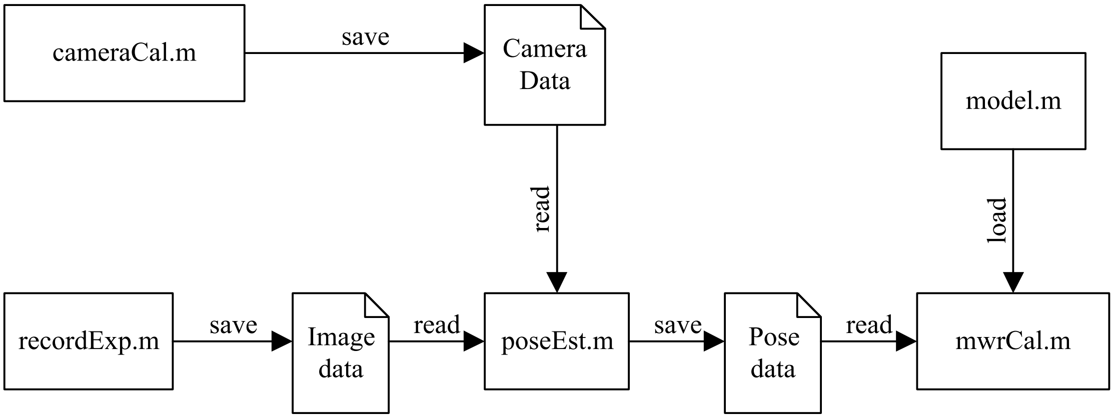


Figure 6-3 MWR Calibration Software

The recordExp.m program captures a predetermined number of images from the camera. Each of the images is time-stamped since the sampling rate is non-uniform. The images are processed to detect the fiducial markers and estimate their image co-ordinates. The vector of time-stamped fiducial marker co-ordinates is saved to an experimental data file along with other relevant experimental information such as the robot control vector and the geometry of the fiducial target.

The poseEst.m implements program the fully projective formulation of Lowe's pose recovery algorithm described in Chapter 4. First it reads the image data output from the recordExp.m program. The image data is processed to yield a time-stamped robot pose for each image frame. The pose data is saved in a pose data file with addition experiment information.

The mwrCal.m program implements the 2nd order divided difference filter for model parameter estimation described in Chapter 5. This program reads the pose data file and re-samples the time-series pose data to be uniformly sampled in time. The generic state-space form robot kinematic model is loaded from a user specified model (model.m). The divided difference filter is applied to yield and estimate of the robot model parameters.

6.1.3 Camera

The camera captures a sequence of images of the robot during a calibration experiment. The camera is the Creative® Video Blaster WebCam II, which is an inexpensive commercial video camera. The camera is capable of sampling images of up to 704 x 576 pixels at to 15 frames per second. An image of 320 x 240 pixels is typically generated and samples at 6 Hz. The camera's horizontal field of view is 52°. Depth of field is 50 mm to infinity. Nominal focal length is 8 mm. The camera system generates time-stamped image files stored in PC memory. The image files are converted into image sequences for analysis in MATLAB using the video for windows (VFW) library.

6.1.4 Robot Under Calibration

The robot under calibration is a two-wheeled differential mobile robot as shown in **Figure 6-4**. The robot is the "Rug Warrior" available from Acroname Robotics Inc. [ACR02]. This robot was specifically designed at the Massachusetts Institute of Technology as a simple and flexible platform of robotics research. Over a thousand Rug Warriors are currently in service throughput universities and homes around North America.

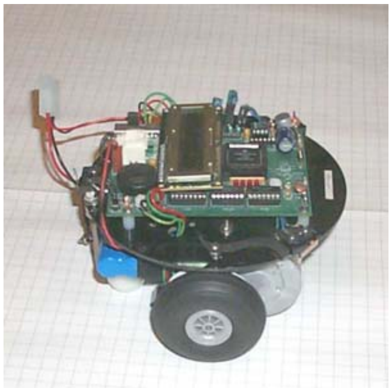


Figure 6-4 Robot Under Calibration: Rug Warrior

The Rug Warrior is a small inexpensive differential wheeled mobile robot. **Figure 6-4** illustrates the left side view of the Rug Warrior mobile robot. Two independent DC motors drive 60 millimeter diameter rubber wheels. The motors are mounted beneath an acrylic chassis plate 150 millimeters in diameter. Separate motor and logic batteries are mounted under the chassis plate behind the motors. A single caster wheel mounted at the rear of the robot provides longitudinal stability. Note, that the manufacture supplied ball caster was replaced with a high quality wheel caster. Early experiments conducted with the robot showed that frictional forces between the ball caster and floor altered the robot trajectory.

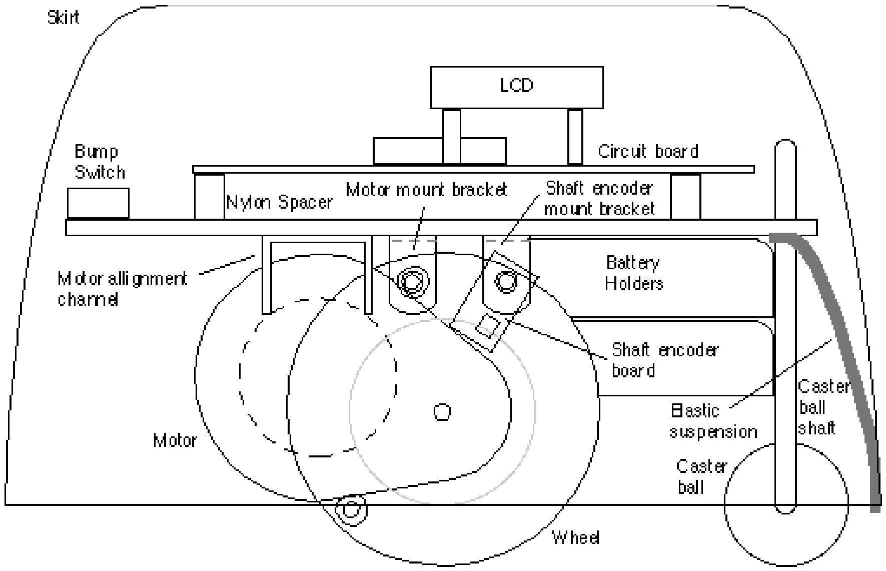


Figure 6-5 Rug Warrior Mobile Robot Side View

The robot controller is a single board computer with a MC68HC11A1FN microprocessor with 256 Kbytes of on board memory. The board provides pulse width modulated control of the two motors. The robot is equipped with tactile bump sensors around it circumference. The bump sensors detect which portion of an exterior plastic skirt comes in to contact with objects. A combination of light, infrared ranging and ultrasonic ranging sensors provide spatial sensory data to the controller for navigation. Optical shaft encoders mounted inside of each drive wheel measure wheel rate of rotation to support computation of robot position and speed.

Control programs are written in the C programming language on the calibration computer, assembled into 68HC11 binary code, and downloaded to the microprocessor. Alternately, the calibration PC may directly control the robot by an RS-232C serial interface. The robot wheel speed is commanded to the control microprocessor as an integer in the range [-50, 50]. The control function, which relates wheel speed to the commanded value, is determined by measurement of wheel rotational speed versus commanded speed. The wheel speed control function is statistically determined from the computation of mean wheel angular rate over 10 second intervals. **Figure 6-6** illustrates the control function which is non-linear. The robot kinematic models developed in Chapter 3 define the control vectors in terms of wheel angular speed. This data is stored and used as a look up table to convert between commanded speed and wheel angular rate.

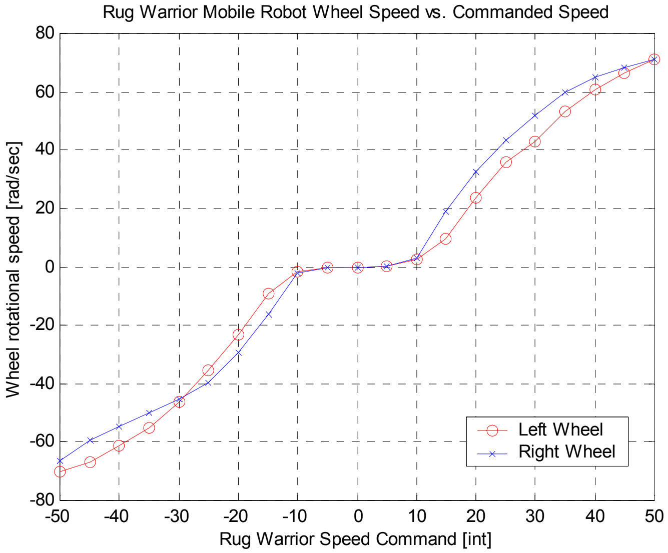


Figure 6-6 Robot Wheel Speed Control Function

Table 6-1 lists the key parameters of the Rug Warrior mobile robot. The parameters to be estimated by photogrammetric calibration are the left wheel radius, right wheel radius and base.

Table 6-1 Rug Warrior Mobile Robot Parameters

Parameter	Value
Left Wheel Radius (R_L)	31.0 mm ±0.1 mm
Right Wheel Radius (R_R)	31.3 mm ±0.1 mm
Wheel Base (D)	148 mm ±0.5 mm
Mass	890 g ±5 g

6.2 Robot Calibration Procedure

The robot calibration procedure consists of four major stages. The first stage is the preparation of the calibration apparatus and robot under calibration. The second stage is calibration of the camera. The third stage is acquisition of data from a calibration experiment. The fourth stage is the application of computer algorithms to the sampled data to estimate robot model parameters. **Table 6-2** summarizes the calibration procedure as an activity list.

Table 6-2 Robot Calibration Procedure

1. **Preparation**
    - Set up apparatus
    - Mount camera to tripod.
    - Position tripod.
    - Connect camera to computer parallel RS-232 interface.
    - Position calibration workspace in front of camera and tripod
    - Adjust camera orientation so that the workspace is fully viewed.
2. **Calibrate Camera**
    - Place calibartion target in calibration workspace.
    - Verify detection of all calibration markers.
    - Run calibration script.
    - Save calibration conifguration file.
3. **Robot Calibration Experiment**
    - Verify robot configuration including tire pressure, battery charge and caster.
    - Mount fiducial marker target on robot.
    - Program experiment control parameters in robot via serial interface from PC.
    - Place robot at initial position and orientation.
    - Start MATLAB script to capture experiment image sequence.
    - Command robot start maneovering.
    - After the required number of samples has been captured stop recording and stop the robot.
    - Store image sequence in files on computer hard drive.
4. **Calibration Processing**
    - Run fiducial marker detection script to detect the fiducial markers in the calibration images.
    - Run pose estimation MATLAB script to determine robot pose in each image.
    - Run estimation MATLAB script to determine robot model parameters.

The following sub-sections describe the calibration procedure in more detail with a typical experiment as an example.

6.2.1 Stage 1: Preparation for Calibration

Preparation of the calibration apparatus begins with installation of the computer and required software. Connect camera to computer parallel RS-232 interface. Position the calibration workspace in front of camera and tripod. Adjust camera orientation so that the workspace fills the camera filed of view. **Figure 6-7** shows a photograph of the prepared apparatus used for calibration experiments. The camera is mounted approximately 1.5 meters above the calibration workspace on a photographic tripod.

Note that uniform lighting is critical to reliable detection of the fiducial markers. The suitability of lighting conditions must be verified by testing the fiducial detection algorithm prior to camera calibration.

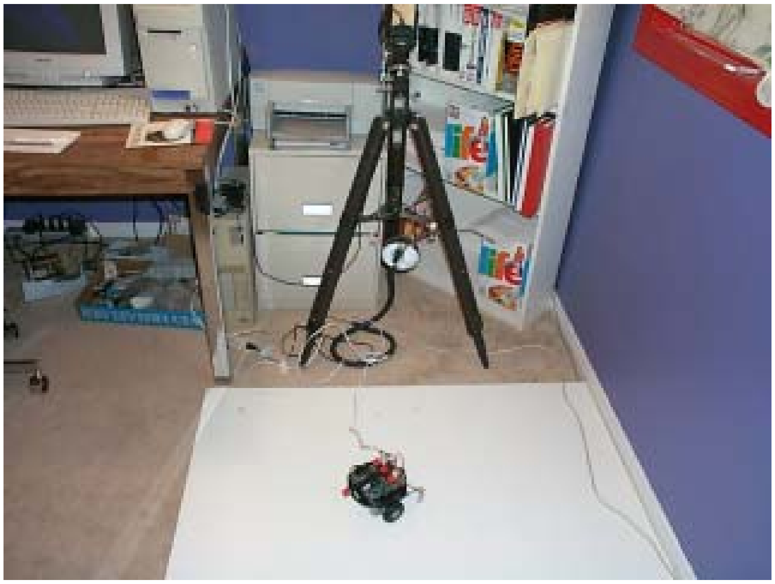


Figure 6-7 Photogrammetric Apparatus

6.2.2 Stage 3: Camera Calibration

Camera calibration begins by placing the calibration target in the calibration workspace. All of the calibration target must be within the camera field of view. A fiducial marker detection program is executed to verify that all calibration target markers can be detected during camera and robot calibration. Execute the camera calibration program (camerCal.m). If the camera calibration is successful, the camera model parameters are saved to a data file. **Table 6-3** lists typical calibration results for the WebCam CCD camera.

Table 6-3 Camera Calibration Results

Extrinsic Parameters	Position [mm]	X = -107.33
		Y = -302.45
		Z = 1589.11

	Orientation Euler Angles (XYZ)	ψ = 178.01 deg ϕ = -19.30 deg θ = 89.32 deg
Intrinsic Parameters	Scale factor	S_x = 0.52545
	Effective focal length	f = 6.8313 mm
	Principal point	(u,v) = (151.51,119.88)

	Radial distortion	K1 = -6.522413e-003
		K2 = 2.643364e-004

	Tangential distortion	T1 = -1.713880e-004
		T2 = 1.428032e-003

6.2.3 Stage 3: Robot Calibration Experiment

The first stap of the experimental stage is to verify the robot configuration. This includes checking that tire pressure is nominal, battery charge is full and the wheels and casters are in good working porder. Next, mount the fiducial marker target on robot ensuring that it is secure and can not move while the robot maneouvers.

Program experiment control parameters in robot via serial interface from PC. Place robot at initial position and orientation chosen such that the robot's expected trajectory is with the camera field of view. Start MATLAB script to capture experiment image sequence. Command robot start maneovering. The robot will maneovre according to its program as the camera records images at a rate of about 6 Hz. After the required number of samples has been captured, stop recording and stop the robot. **Figure 6-8** shows a typical image captured during an experiment.

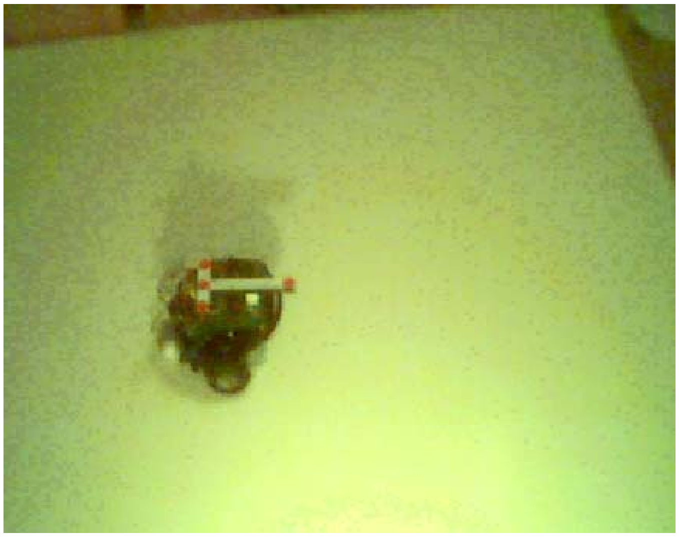


Figure 6-8 Sample Experiment Image (exp09)

Each of the captured images is processed to detect the fiducial markers and estimate their image position in the image. **Figure 6-9** illustrates six steps in the feature detection algorithm. The marker detection algorithm is described in Chapter 4 Section 4.4.

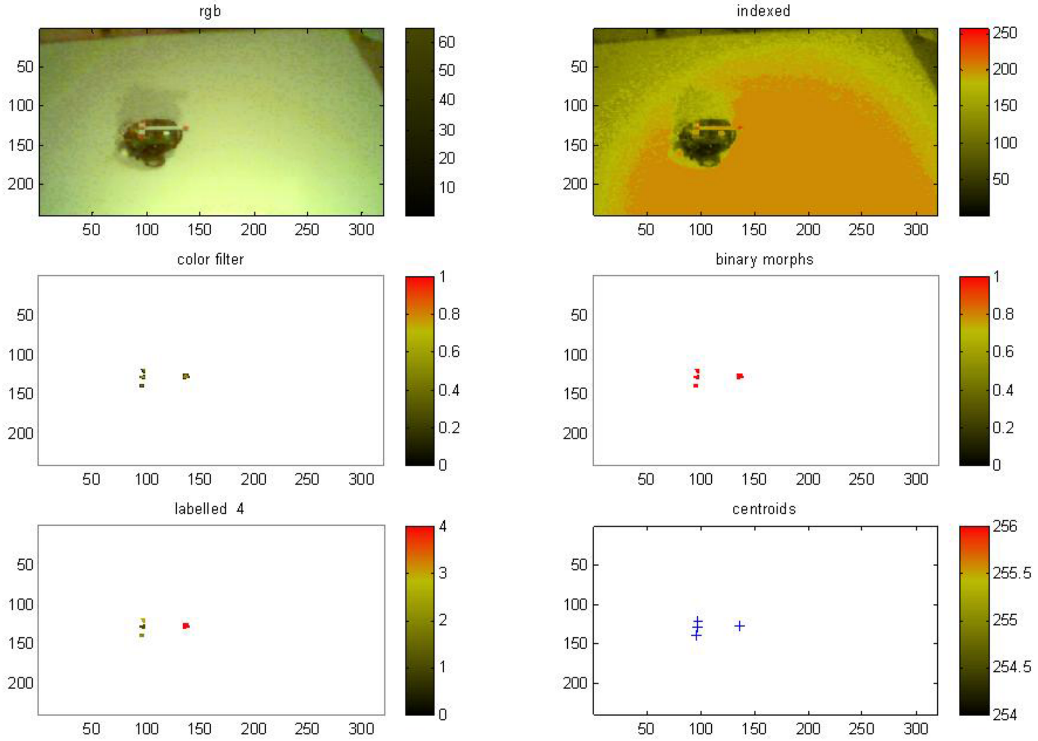


Figure 6-9 Robot Pose Detection Process (exp09)

Figure 6-10 shows the original unprocessed image with the marker estimates overlaid. As indicated in chapter 4 the variance of the marker estimate error is within 0.015 pixels. After all images are processed, the time series of image feature data is stored in a file on computer hard drive. The file contains all necessary information for further processing including camera data, the control vector, time stamps, the time series of marker positions and a model of the calibration target.

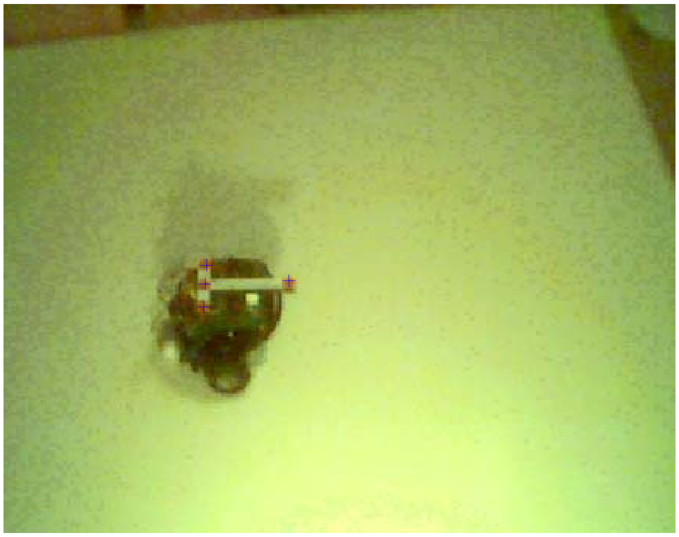


Figure 6-10 Image of Robot with Marker Position Estimate Overlay

6.2.4 Stage 4: Calibration Processing

The fourth stage of calibration is the processing of image feature data. First, the pose estimation program is applied to the time series image feature data saved to file from the image feature extraction. **Figure 6-11** shows image feature trajectory for an experiment with 128 image samples. The control vector was $\begin{bmatrix}\omega_L & \omega_R\end{bmatrix} = \begin{bmatrix}4.7 & 9.8\end{bmatrix}$ rad/sec. This control vector should cause the robot to traverse counter clock wise along an arc with a radius of 207.9 mm. The four target feature points are evident as four concentric ellipses.

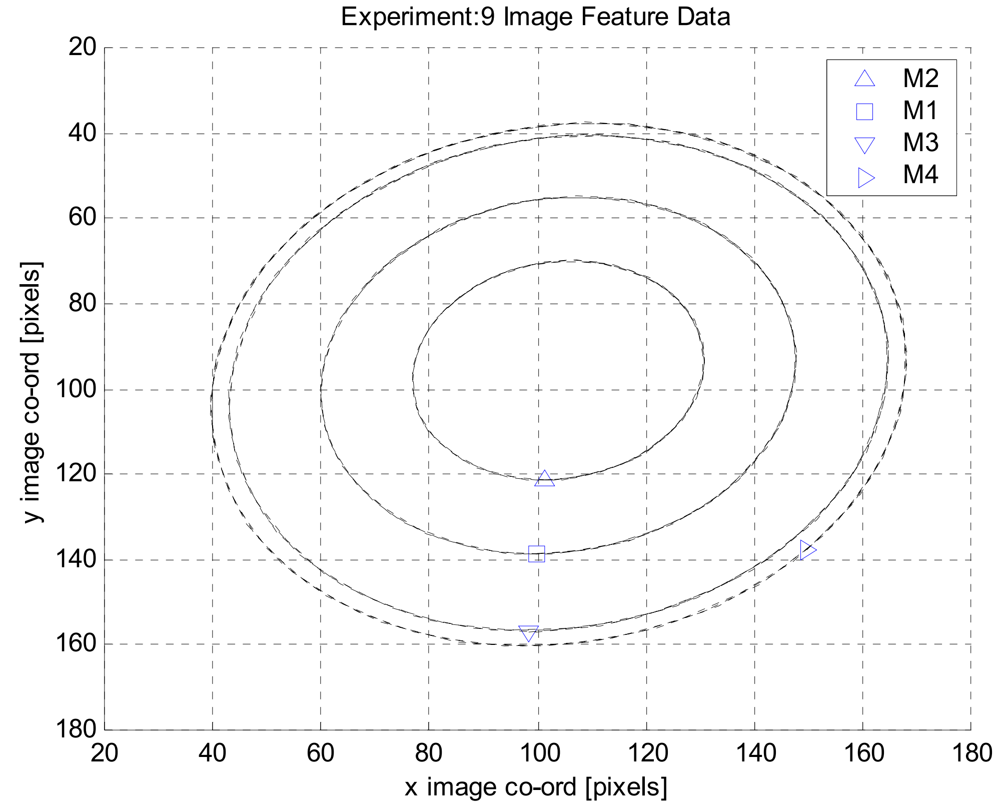


Figure 6-11 Robot Target Image Feature Trajectories

The pose estimation program estimates robot pose for each set of image features. The pose estimation program outputs time series of pose estimates. Figure 6-12 shows the estimated robot pose trajectory in three dimensions. The camera pose is also shown in the top right of the figure. This type of plot is very useful as a sanity check of the pose estimate results. A good pose estimate will constrain the robot z co-ordinate to the ground plane (z=0).

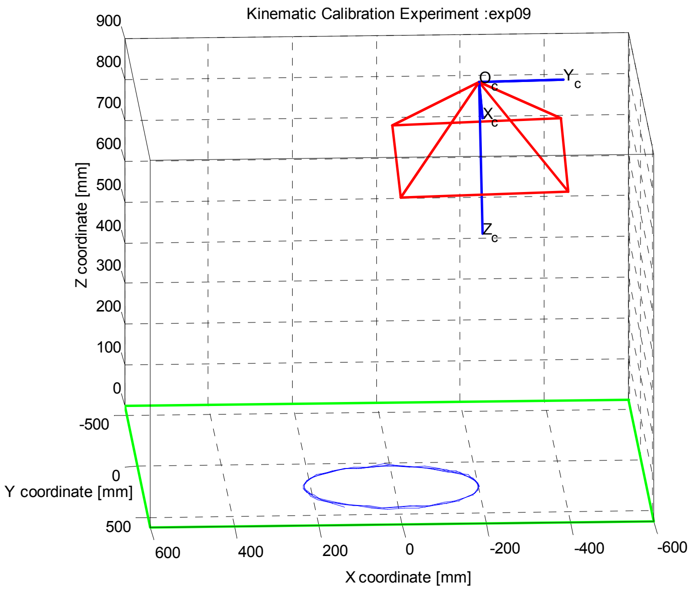


Figure 6-12 Estimated Robot Pose Trajectory

Figure 6-13 provides a plan view, that is the x-y plane, of the estimated robot x and y co-ordinates for all 128 images. The robot trajectory is circular with a radius of approximately 206 mm which is 1 mm less than the theoretical ICR radius. The robot pose data is saved to file for further processing.

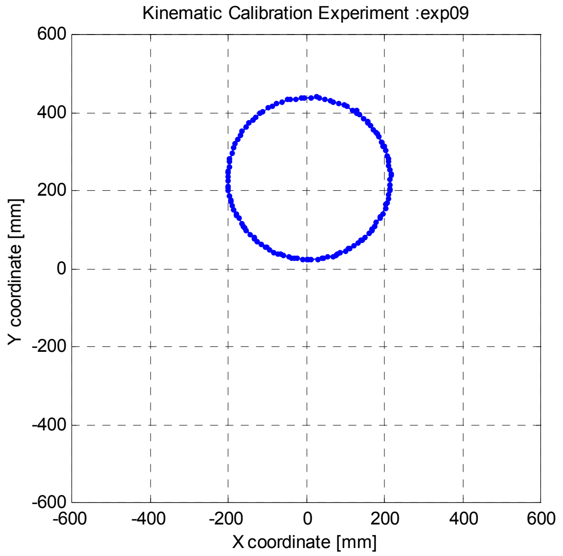


Figure 6-13 Robot Pose Estimates (plan View)

The last step of robot calibration is to run model parameter estimation program mwrCal.m. The input is the estimated pose data from poseEst.m. The program requires initial values of the model parameters and process covariance. The initial parameters are set to the nominal values specified by robot manufacturer. Process covariance is determined by conducting independent runs of the divided difference filter algorithm. Figure 6-14 shows the parameter estimates as a function of sample. The upper plot shows the left and right wheel radii estimates in red and blue respectively. The lower plot shows the wheel base estimate. The estimates quickly converge by the first twenty samples.

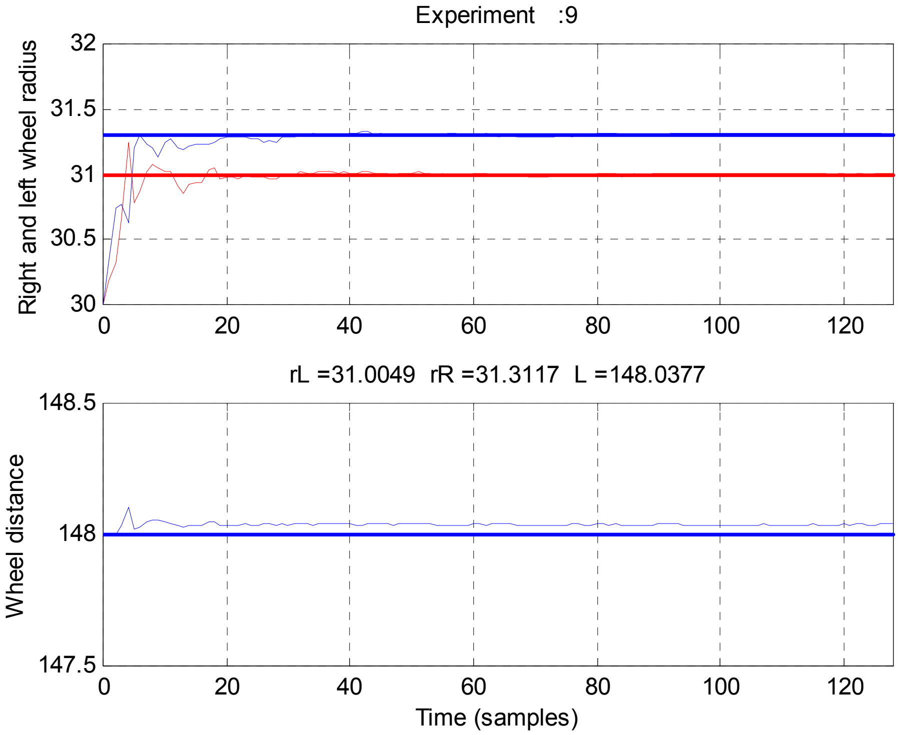


Figure 6-14 Estimated Robot Model Parameters

6.3 Differential Wheeled Mobile Robot Calibration Experiments

This section presents the results of 99 calibration experiments. Each experiment has a unique control vector. The control vector is the vector of left wheel speed and right wheel speed in units of radians per second. The control vectors are selected such that they cover the robot control space subject to a constraint on instantaneous center of rotation. The control vector for each experiment is listed in **Table 6-4**.

Table 6-4 Differential WMR Experiments

Exp #	ω_L [rad/s]	ω_R [rad/s]	Exp #	ω_L [rad/s]	ω_R [rad/s]	Exp #	ω_L [rad/s]	ω_R [rad/s]
1	2.3	7.5	34	1.2	5.1	67	29.6	4.7
2	2.3	12.9	35	14.9	35.2	68	29.6	14.6
3	2.3	43.0	36	14.9	41.1	69	29.7	12.3
4	2.4	37.7	37	15.0	30.3	70	29.8	9.5
5	2.5	18.0	38	15.0	44.6	71	27.1	10.8
6	2.6	22.7	39	15.1	25.0	72	25.2	47.2
7	2.6	27.8	40	3.2	20.2	73	15.5	39.0
8	2.9	32.6	41	15.6	4.7	74	32.5	2.3
9	4.7	9.8	42	17.2	33.0	75	34.7	14.8
10	5.0	20.0	43	17.3	2.2	76	12.9	32.6
11	5.0	24.6	44	17.6	7.0	77	5.0	1.7
12	5.0	34.4	45	19.5	2.9	78	12.0	36.6
13	5.0	44.3	46	19.6	34.9	79	35.1	5.1
14	5.1	14.8	47	19.7	37.1	80	35.1	9.5
15	5.1	29.8	48	19.8	45.2	81	35.1	19.0
16	5.3	39.3	49	19.9	9.6	82	33.1	16.6
17	7.1	21.9	50	19.9	40.2	83	37.6	1.6
18	7.4	12.4	51	32.2	10.9	84	39.4	9.8
19	7.5	17.1	52	20.3	5.0	85	39.6	4.4
20	7.6	27.2	53	17.5	37.0	86	39.6	14.6
21	7.8	2.6	54	22.8	2.6	87	39.7	19.9
22	9.9	19.6	55	22.8	7.0	88	40.0	22.7
23	9.9	29.1	56	24.5	9.6	89	38.0	12.5
24	9.9	34.9	57	24.7	4.7	90	8.6	32.5
25	10.0	4.5	58	21.8	42.8	91	45.2	22.1
26	10.0	44.9	59	24.9	14.8	92	43.1	5.8
27	2.1	46.8	60	22.9	10.0	93	44.7	9.8
28	10.1	24.4	61	6.1	31.8	94	44.8	19.2
29	10.5	40.8	62	25.1	44.4	95	44.8	15.0
30	12.2	2.2	63	25.5	12.7	96	44.9	25.0
31	12.4	27.0	64	27.1	2.0	97	17.0	7.9
32	12.6	22.1	65	27.3	7.0	98	3.0	6. 0
33	12.8	6.6	66	32.5	7.4	99	45.3	4.7

The ICR constraint is imposed to ensure the robot under calibration will remain within the calibration workspace and camera field of view. A practical limitation on ICR less than 300 mm is chosen. **Figure 6-15** illustrates the control vectors for the 99 calibration experiments. The dashed lines represent the contour of control vectors that result in an ICR of 300 millimeters. The region between these lines is the set of control vectors with ICR greater than 300 millimeters.

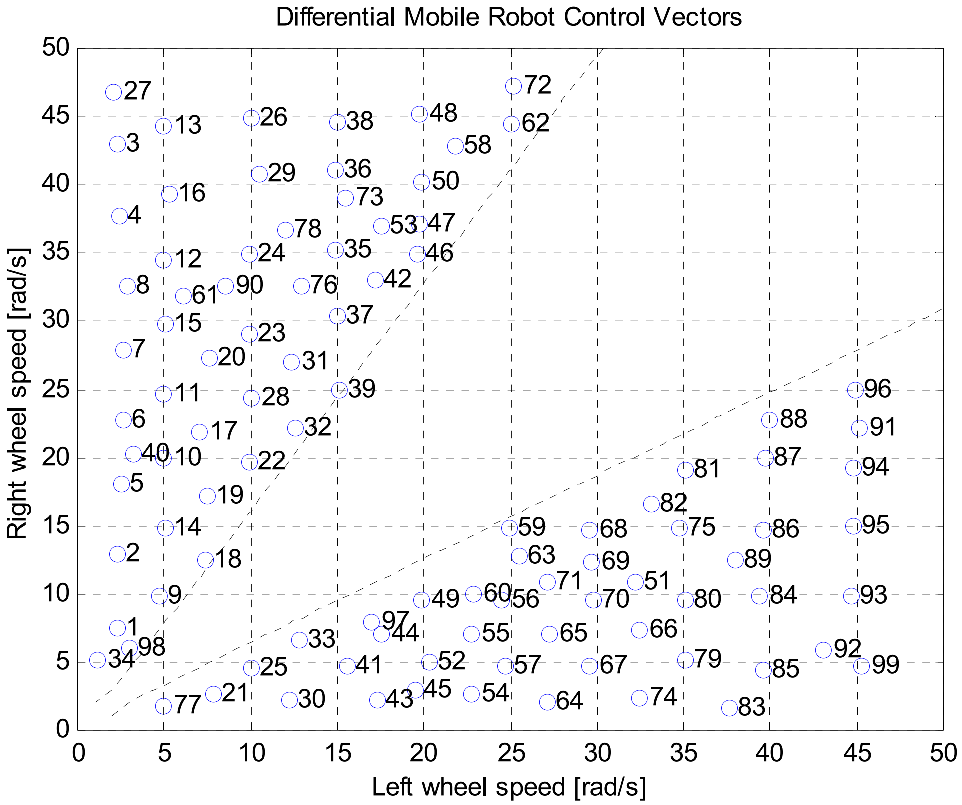


Figure 6-15 Differential WMR Calibration Experiment Control Vectors

6.4 Parameter Estimation

The procedure described in section 6.2 is applied to the robot for experiments 1 through 99. **Table 6-5** summarizes the calibration results, estimated wheel radius and wheel base, for experiments 1 through 99.

Table 6-5 Robot Calibration Results (Experiments 1-99)

#	R_L	R_R	D	#	R_L	R_R	D	#	R_L	R_R	D
1	30.970	31.322	148.130	34	30.817	31.292	148.180	67	31.038	30.932	148.390
2	30.788	31.295	148.240	35	30.948	31.293	148.130	68	31.017	31.292	148.090
3	31.670	30.591	144.610	36	30.880	31.246	148.050	69	30.955	31.191	148.160
4	30.254	31.243	148.000	37	30.995	31.307	148.040	70	30.982	31.153	148.220
5	30.786	31.375	148.370	38	30.846	31.307	148.400	71	30.998	31.251	148.130
6	30.489	31.303	148.320	39	30.991	31.300	148.030	72	30.900	31.239	148.150
7	30.417	31.279	148.110	40	30.799	31.328	148.360	73	30.893	31.276	148.230
8	30.454	31.277	148.250	41	31.029	31.238	148.230	74	31.001	30.218	148.460
9	31.005	31.312	148.040	42	31.000	31.318	148.060	75	30.985	31.242	148.190
10	30.875	31.316	148.190	43	31.069	30.907	148.560	76	30.920	31.292	148.160
11	30.817	31.281	148.220	44	31.008	31.271	148.140	77	31.056	31.246	148.180
12	30.595	31.328	148.360	45	31.032	31.010	148.500	78	30.931	31.338	148.310
13	30.298	31.200	147.890	46	30.981	31.290	148.040	79	30.985	30.746	148.380
14	30.951	31.287	148.100	47	30.980	31.306	148.080	80	31.043	31.173	148.370
15	30.670	31.295	148.350	48	30.874	31.309	148.560	81	31.024	31.303	148.100
16	30.481	31.216	148.070	49	30.992	31.276	148.090	82	31.010	31.284	148.120
17	30.930	31.284	148.120	50	30.946	31.291	148.130	83	30.916	29.899	147.770
18	30.973	31.267	148.020	51	30.976	31.193	148.150	84	31.001	31.016	148.500
19	30.983	31.310	148.070	52	31.023	31.146	148.270	85	30.985	30.602	148.290
20	30.911	31.348	148.210	53	30.925	31.269	148.130	86	30.993	31.242	148.180
21	31.024	31.256	148.190	54	31.044	30.795	148.540	87	31.005	31.265	148.200
22	31.018	31.328	148.030	55	31.027	31.247	148.230	88	30.989	31.267	148.090
23	30.963	31.324	148.150	56	30.996	31.237	148.140	89	31.000	31.156	148.310
24	30.839	31.287	148.270	57	31.028	31.074	148.390	90	30.873	31.312	148.250
25	30.971	31.253	148.110	58	30.950	31.304	148.230	91	30.994	31.219	148.320
26	30.544	31.262	148.450	59	31.015	31.307	148.050	92	30.959	30.730	148.180
27	30.743	30.953	146.460	60	31.031	31.307	148.120	93	30.977	31.062	148.120
28	31.003	31.313	148.080	61	30.744	31.344	148.410	94	30.986	31.170	148.350
29	30.689	31.237	148.240	62	30.995	31.313	148.110	95	31.039	31.114	148.610
30	31.012	31.026	148.390	63	31.012	31.281	148.090	96	30.984	31.250	148.190
31	30.987	31.298	148.040	64	31.007	30.523	148.560	97	31.007	31.295	148.100
32	31.010	31.318	148.020	65	31.011	31.266	148.130	98	30.975	31.287	148.040
33	31.039	31.326	148.080	66	30.983	31.046	148.290	99	30.864	30.589	147.750

Figure 6-16 illustrates the calibration results. The upper plot shows the calibration results for estimated left wheel radius. The middle plot shows the calibration results for estimated right wheel radius. The lower plot shows the calibration results for estimated wheel base. The blue dots represent the estimate for each experiment. The red lines represent the actual value determined through independent measurement.

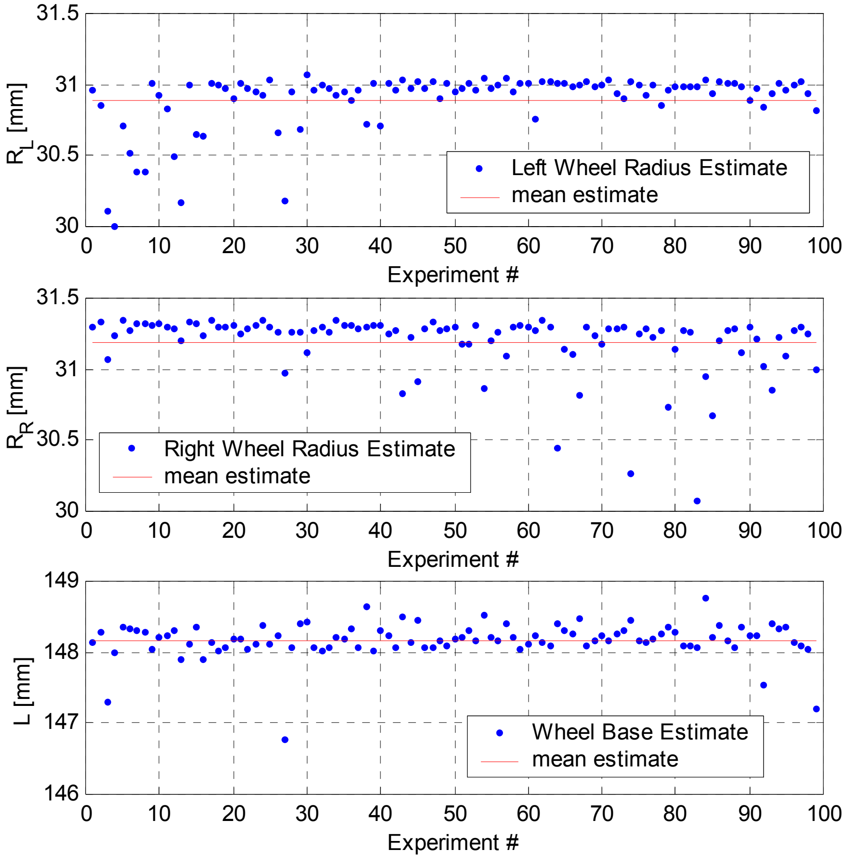


Figure 6-16 WMR Calibration Results

The mean of the left wheel radius estimates is 30.894 mm. The mean of the right wheel radius estimates is 31.189 mm. The mean of the wheel base estimates is 148.17 mm. The left wheel estimate error magnitude has a minimum, maximum and mean of 0.00024648 mm, 0.99895 mm and 0.10611 mm respectively. The right wheel estimate error magnitude has a minimum, maximum and mean of 0.00065041 mm, 1.2329 mm and 0.11069 mm respectively. The wheel base estimate error magnitude has a minimum, maximum and mean of 0.0054033 mm, 1.2224 mm and 0.17029 respectively. Figure 6-17 provides histograms of estimate errors.

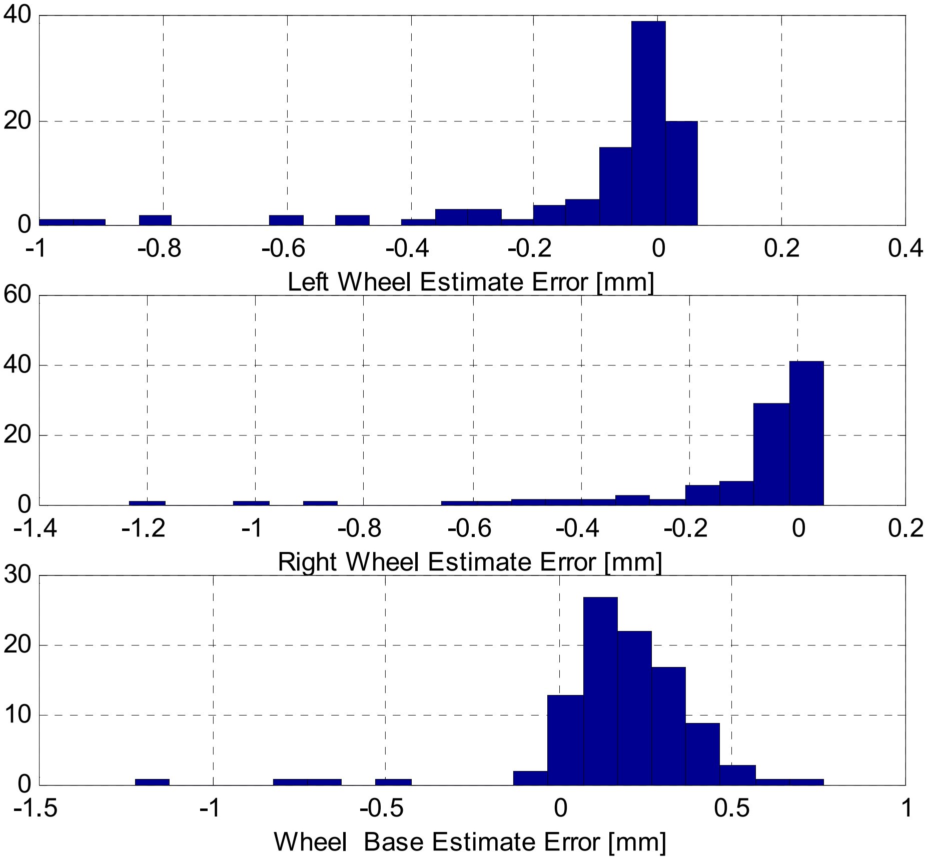


Figure 6-17 Robot Model Parameters Estimate Error Histograms

7. Discussion

This chapter discusses the results of the proposed mobile robot calibration methodology. The first section addresses the requirement that the robot under calibration be observable, controllable and identifiable. The second section proposes metrics of robot calibration. The third section develops an error model of the calibration process. The error model establishes the dependency between the calibration error and the measurement and calibration process errors.

7.1 Observable, Controllable and Identifiable Processes

A prerequisite to parameter estimation and hence the calibration process is that the robot model under calibration be observable, controllable and identifiable [Kalman60]. These three requirements must be verified prior to robot calibration. If they are not satisfied, the accuracy of calibration is questionable.

A robot is observable if it is possible to determine its kinematic state from measurements. The photogrammetric experiments conducted in this thesis clearly show that kinematic state can be determined from measurement. Thus the non-holonomic differential wheeled robot considered is observable.

A robot is controllable if it is possible to find an unconstrained control vector, which bring the robot from any initial pose to any specified pose in finite time. For a wheeled robot this is equivalent to asking the question if is it possible to move from one point to another by following a path consisting of circular arcs with controllable radius. In this context the differential wheeled mobile robot has been shown to controllable [Laumond98].

A robot is identifiable it is possible to determine the system model from the measurements of kinematic state and control vector. The calibration experiments conducted in this thesis show that a differential mobile robot is identifiable since the model parameters were estimated with small error.

7.2 Robot Calibration Metrics

A key question in calibration is what determines the goodness of the calibration results. The metric should be universal in that it applies equally to different models and different calibration techniques. The metric must account for variable dimension of the state vector. It must also account for the variable scale of the parameters to be estimated.

The simplest metric is the arithmetic difference between the estimated and true robot parameters. Recall, that the kinematic model is given by the function $\dot{\mathbf{x}} = f\left(\mathbf{x}, \mathbf{a}, \mathbf{u}\right)$, where **x** denotes robot kinematic state, **a** denotes model parameters and **u** denotes the control vector. The calibration error, **Δa**, is the difference between the estimated model parameters resulting from the calibration process and the true robot parameters. The true robot parameters are ideal in that they perfectly relate robot kinematic state with control. This metric is a vector whose elements must be considered separately.

$$\Delta\mathbf{a} = \left|\mathbf{a}_{cal} - \mathbf{a}_{true}\right| \tag{7.1}$$

However a scalar metric is required to assess overall calibration performance. In their evaluation of pose estimation performance Araujo, Carceroni and Brown [ACB94] use the norm of distance error (NDE) as the performance metric. The NDE is the norm of the difference between the estimated vector and the true vector. NDE provides a meaningful measure of the difference between two vectors. The only problem with NDE is that it does not account for the differing scale of robot parameters. The normalized NDE, which is the NDE of the normalized calibration error, is proposed as the metric for robot calibration. Let the normalized calibration error be defined as

$$\Delta\mathbf{a}^{'} = \frac{\left|\mathbf{a}_{cal} - \mathbf{a}_{true}\right|}{\mathbf{a}_{true}} \tag{7.2}$$

where division is element by element. Thus, the normalized norm of distance error (NNDE) of the calibrated parameters is given by

$$\begin{aligned}\Delta\mathbf{a}_{NNDE} &= \left\|\Delta\mathbf{a}'\right\| \\ &= \left\|\frac{\Delta\mathbf{a}_{cal} - \Delta\mathbf{a}_{true}}{\Delta\mathbf{a}_{true}}\right\|\end{aligned} \tag{7.3}$$

Recalling the calibration results from Chapter 6, the NNDE is 0.00466 or approximately 0.5 percent.

7.3 Calibration Error Model

The goal of robot calibration is to minimize the calibration error as defined above. A calibration error model can be developed to identify methods to reduce calibration error. The calibration error model relates the calibration error to measurement and processing factors. A suitable error model provides guidance to adjust these factors to reduce the calibration error. The factors that affect the calibration accuracy are determined by inspection of the calibration process as shown in Figure 7-1.

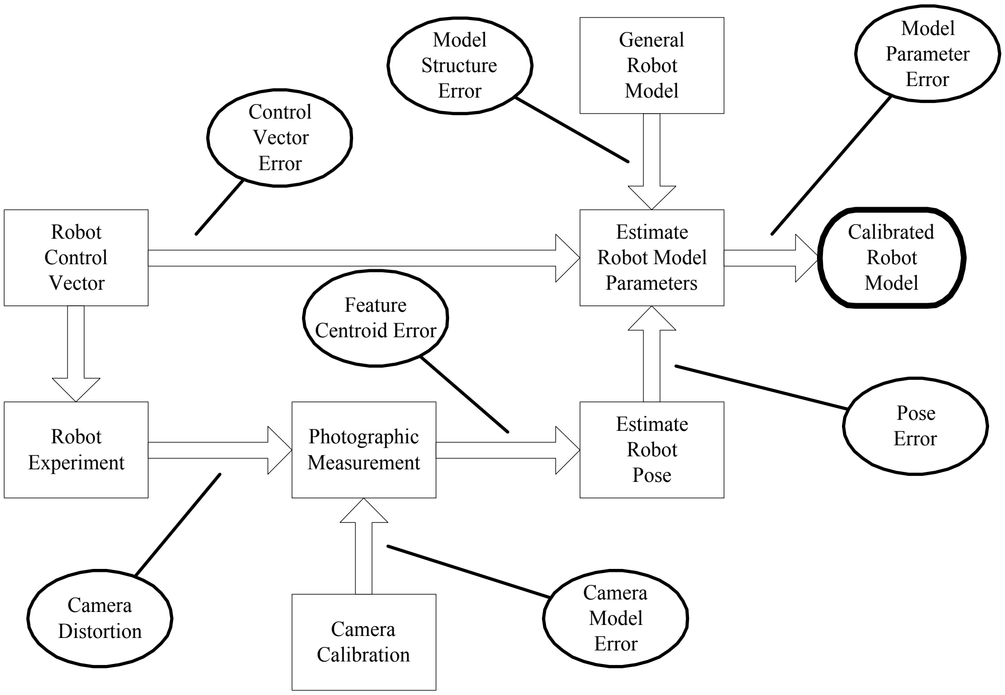


Figure 7-1 Robot Calibration Error Sources

The significant factors affecting the calibration error of the proposed process are:

a. The model structure error ***Δf()*** which is further defined in Section 7.3.1.

b. The control vector error **Δu** which is further defined in Section 7.3.2.

c. The image feature position error **Δp**, which is further, defined in Section 7.3.3.

d. Pose estimation error **Δx**, which is further, defined in Section 7.3.4.

All of the calibration processing is performed in the MATLAB environment and thus subject to floating point number computation errors. The general computation error due to floating point computation is 2.22044604925031e-16. This number is very small and is considered negligible to the calibration error.

7.3.1 Kinematic Model Error

The model structure error ***Δf()*** is the difference between the kinematic model, *f()*, used for calibration and the ideal robot model, $f^o()$. The proposed calibration process requires the model structure to be known *a prior* of calibration experiments. The model is determined from the geometric principles described in Chapter 3. There is no inherent way to correct for this error. The user must keep watch for large systematic error in the calibration results. Such errors may be attributable to the model structure error.

7.3.2 Control Vector Error

The control vector error is due to measurement error in the robot control mechanism. For example, the right wheel may been commanded to rotate at 10 rad/sec but is actually rotating at 11 rad/sec. This error is specific to the robot under calibration. As in the case of the Rug Warrior, most mobile robots use shaft encoders to measure angular rate on each wheel to determine the commanded speed. The shaft encoder measures wheel rotation by counting number of pulses from the encoder sensor. Each pulse corresponds to a predetermined rotation angle. The angular speed is given by dividing the pulse count by the sampling duration. The Rug Warrior encoders give one pulse per 10 degrees of rotation to give 36 pulses for one full rotation. The variance of the quantization error, as a function of angle quantization $\Delta\phi$=10 degrees, is given by

$$\sigma_\phi^2 = \int_{\phi_n-\Delta\phi}^{\phi+\Delta\phi} (\phi-\phi_n)^2 \, d\phi$$
$$= \frac{(\Delta\phi)^2}{12} = \frac{\left(\left(10/360\right)2\pi\right)^2}{12} \tag{7.4}$$
$$= 2.538\times10^{-3}\,rad$$

The encoder is sampled at every 64 milliseconds with a clock resolution of 0.125 microseconds. The temporal sampling variance is negligible and thus the variance of commanded wheel rate is approximately 2.5e-3 rad/sec. This error is negligible compared to the control resolution of 0.1 rad/sec.

7.3.3 Image Feature Position Error

The image feature position error is the difference between the estimated co-ordinates and the actual co-ordinates of fiducial markers detected in a digital image. Figure 7-2 illustrates the feature position error. The image feature position error is affected by image the quantization error and residual camera distortion.

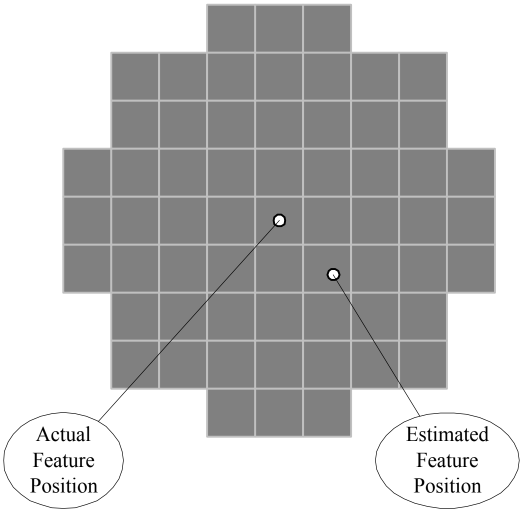


Figure 7-2 Image Feature Position Estimate

The image quantization error is due to the sampling process of the digital camera. The quantization is determined by the number of pixels in the horizontal and vertical axes of the imaging plane. Assuming the probability of each pixel being occupied is equal. Then the variance due to horizontal quantization is given by

$$\sigma_u^2 = \int_{u_n-\Delta u}^{n+\Delta u} \left(u-u_n\right)^2 du = \frac{\left(\Delta u\right)^2}{12} \tag{7.5}$$

where Δu denotes the horizontal pixel size. Note that this analysis equally applies to the vertical image dimension. The pixel size is determined by dividing the image plane size by the number of pixels assuming uniform pixel size. The unbiased estimate of the centroid [JH00] of an object consisting of N pixels with N_b border pixels is given by

$$u_g = \frac{1}{N}\left(\sum_{i=1}^{N-N_b} u_i + \frac{1}{2}\sum_{j=1}^{N_b} u_j\right) \tag{7.6}$$

Consequently the variance of the feature position [JH00] is given by

$$\sigma_{u_g}^2 = \frac{N_b}{4N^2}\sigma_u^2 = \frac{\left(\Delta u\right)^2 N_b}{48N^2} \tag{7.7}$$

This error is additive and normally distributed. Thus for fixed image quantization the position error variance can be reduced by increasing the apparent size of the fiducial marker.

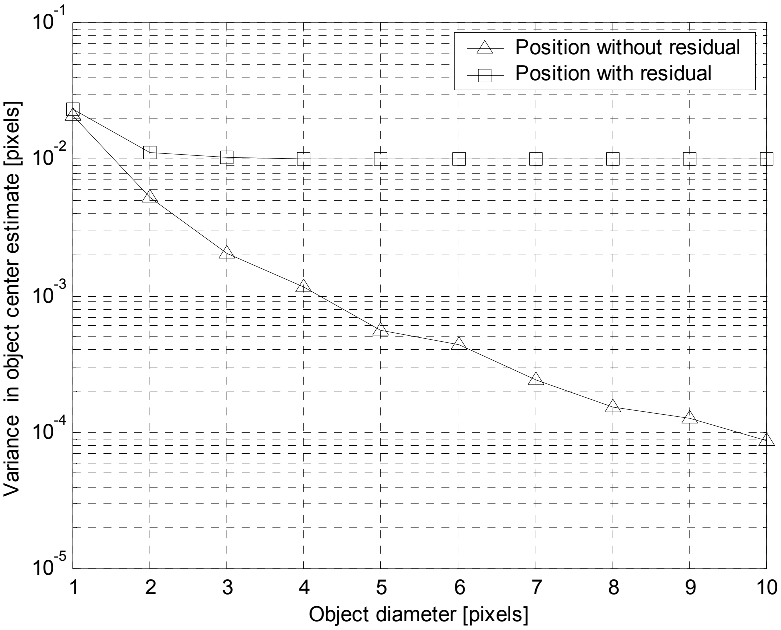


Figure 7-3 Image Centroid Estimate Error Variance vs. Feature Diameter

The residual camera distortion is the camera distortion that is not compensated for by camera calibration. The residual distortion acts as a bias upon the feature position estimate. An analytic determination of the residual distortion is very difficult and will not be attempted here. Heikkila and Silven [HS97] performed experiments to determine the residual distortion for the camera calibration process used in this thesis. They found that the residual error is always less than 0.01 pixels under realistic conditions.

The total image feature position error is found by combining the centroid variance and the residual position variance in quadrature. Figure 7-3 illustrates the centroid variance as a function of image feature diameter in pixels for normalized pixel size. The blue line denotes variance due to the centroid error only. The red line denotes the combined variance. Clearly the residual error dominates the position error for diameters greater than 5 pixels. Fiducial marker size in the calibration images ranges between 5 to 10 pixels. Thus the centroid position error is effectively 0.01 pixels for typical calibration scenarios.

7.3.4 Pose Estimation Error

The pose estimate error is the difference between the actual pose and the estimated pose of an object captured in a camera image. The fully projective formulation of Lowe's pose recovery algorithm is complex and non-linear. Analytic determination of the pose error as a function of image feature position error is not feasible. Instead, Monte Carlo simulation of the pose error due to variable image position errors is conducted. Each data point is computed as the mean of 100 independent simulations. Figure 7-4 shows the results of the Monte Carlo simulation for a stationary target. The simulation indicates that the logarithms of the position and rotation error are linearly related to the logarithm of the centroid position error.

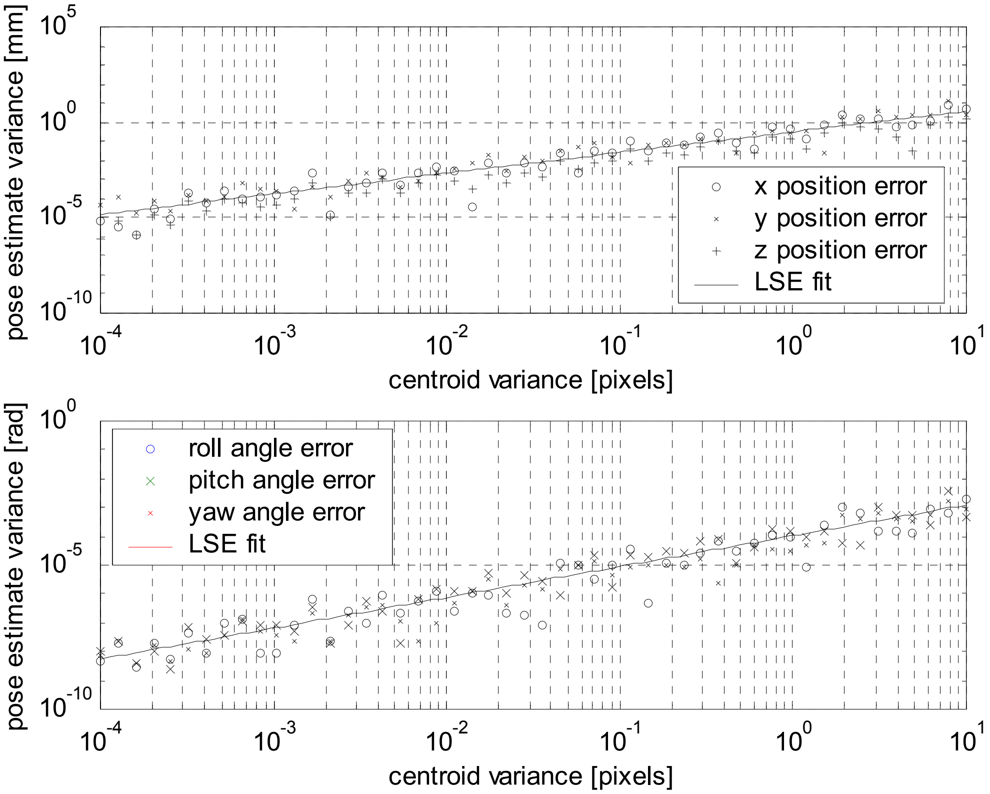


Figure 7-4 Pose Error versus Centroid Error (Fixed target)

A least squares fit of the position errors versus centroid estimate error gives the position error variance as

$$\begin{aligned}\log_{10}\left(\sigma_{pos}^{2}\right) &= 1.0841\cdot\log_{10}\left(\sigma_{u_g}^{2}\right)\text{-}0.49709\\ \sigma_{pos}^{2} &= 0.21835\cdot 10^{\left(1.0841\cdot\log_{10}\left(\sigma_{u_g}^{2}\right)\right)}\end{aligned} \tag{7.8}$$

A least squares fit of the rotation errors versus centroid estimate error gives the rotation error variance as

$$\begin{aligned}\log_{10}\left(\sigma_{rot}^{2}\right) &= 1.069\cdot\log_{10}\left(\sigma_{u_g}^{2}\right)\text{-}3.9803\\ \sigma_{pos}^{2} &= 1.0464\cdot 10^{\left(1.069\cdot\log_{10}\left(\sigma_{u_g}^{2}\right)-4\right)}\end{aligned} \tag{7.9}$$

The above relations between image noise and pose error are consistent with the findings of Carceroni and Brown [CB97] for the sensitivity of pose error to image noise for the fully projective formulation of Lowe's pose recovery algorithm.

In the case of a stationary target, the expected position error variance for the centroid error found during experimentation, of approximately 0.02 pixels, is approximately 0.01 mm. The expected rotation error variance for the centroid error found during experimentation, of approximately 0.02 pixels, is approximately 10^{-5} radians.

A second Monte Carlo simulation is conducted for a maneovering target. This represents more realistic condition where the robot is maneovering during phototgrammetric measurements. Figure 7-5 shows the results of Monte Carlo simulations. Each data point is the mean of 100 independent simulations

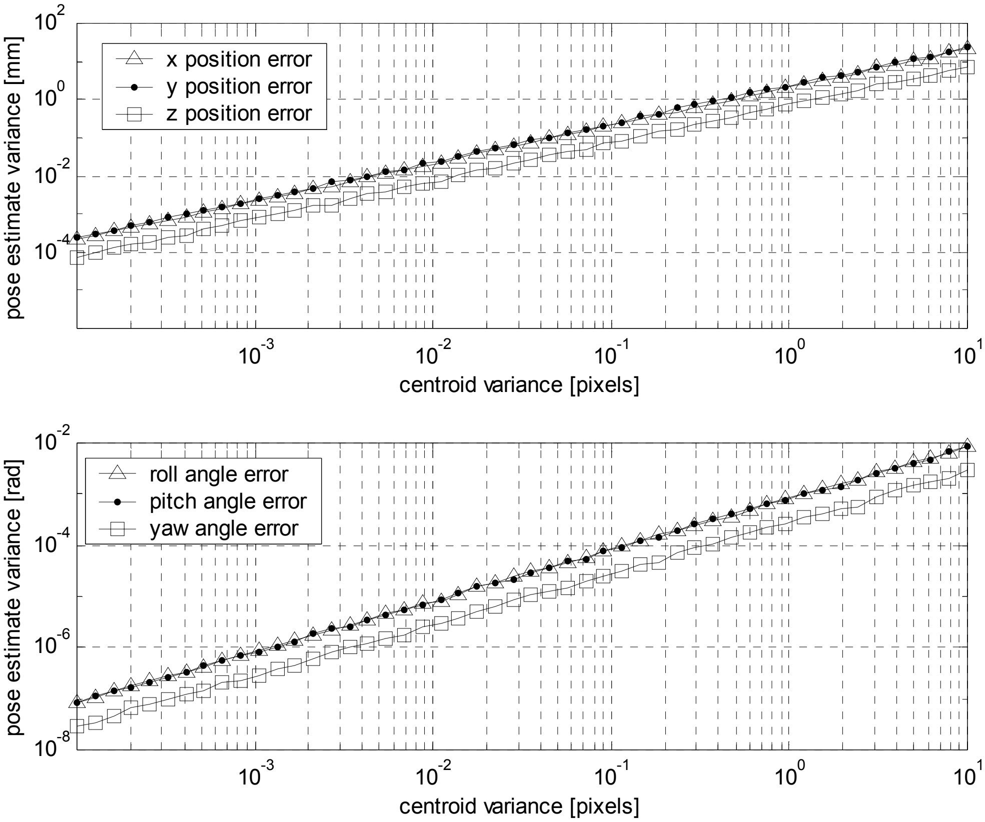


Figure 7-5 Pose Error versus Centroid Error (Moving target)

As for the stationary case, the logarithms of the position and rotation error are linearly related to the logarithm of the centroid position error. A least squares fit of the position errors versus centroid estimate error gives the position error variance as

$$\begin{aligned}\sigma_x^2 &= 10^{0.99681 \cdot \log_{10}\left(\sigma_{u_g}^2\right)+0.32095} \\ \sigma_y^2 &= 10^{0.99825 \cdot \log_{10}\left(\sigma_{u_g}^2\right)+0.36738} \\ \sigma_y^2 &= 10^{0.99683 \cdot \log_{10}\left(\sigma_{u_g}^2\right)-0.13370}\end{aligned} \tag{7.10}$$

A least squares fit of the rotation errors versus centroid estimate error gives the rotation error variance as

$$\begin{aligned}\sigma_x^2 &= 10^{0.99739 \cdot \log_{10}\left(\sigma_{u_g}^2\right)-3.0837} \\ \sigma_y^2 &= 10^{0.99794 \cdot \log_{10}\left(\sigma_{u_g}^2\right)-3.08681} \\ \sigma_y^2 &= 10^{0.99614 \cdot \log_{10}\left(\sigma_{u_g}^2\right)--3.55344}\end{aligned} \tag{7.11}$$

The effect of target motion is an increase in the pose error variance. In the case of a moving target, the expected position error variance for the centroid error found during experimentation, of approximately 0.02 pixels, is approximately 0.1 mm. The expected rotation error variance for the centroid error found during experimentation, of approximately 0.02 pixels, is approximately 10^{-5} radians.

7.3.5 Calibration Error

The robot calibration error is directly affected by the error in the pose estimates and control vectors. Monte Carlo simulation of the calibration process was conducted to determine the sensitivity of calibration error to the error sources discussed previously. The most significant error affecting calibration error is the for pose estimate error. Figure 7-6 shows the results of a Monte Carlo simulation of the calibration normalized norm distance error for variable position errors. Each data point is the mean error for 100 independent simulation for 20 pose error values over the range of 0.01 mm to 20 mm. The pose angle errors are fixed at 1e-5 radians. The effect of increasing pose error upon the calibration error is evident. The calibration error increased by a factor 5 for a 20-fold increase in position error.

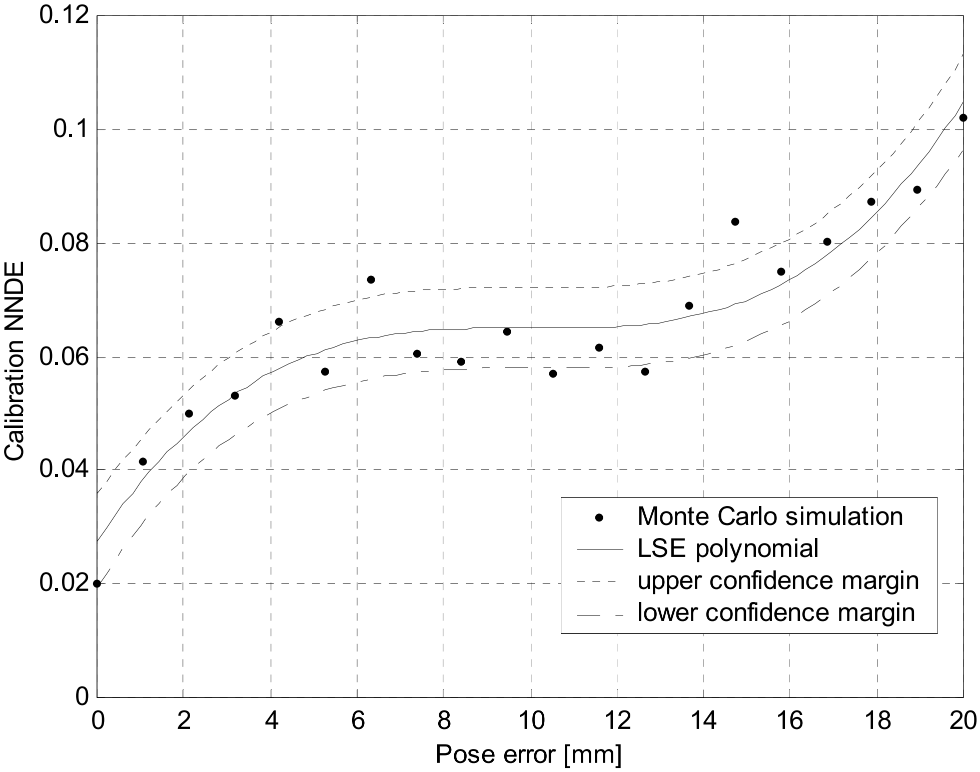


Figure 7-6 Calibration NNDE versus Pose Error

A least squares fit of the data indicates a cubic relationship between the calibration NNDE and pose estimate error. The expected pose error should yield a calibration error of less than 2 percent.

The calibration NDDE of 0.005 determined from experimentation in Chapter 6 is less than the error predicted by the model by a factor of four.

8. Conclusion

This chapter presents a summary of the results of this thesis. Answers are given to the research questions asked in Chapter 1. Candidate areas for future research related to this thesis are identified.

8.1 Summary of Results

In this thesis, a system to calibrate the kinematic model of a wheeled mobile robot is proposed. A methodology to determine the model structure for most wheel mobile robots is proposed. The measurement of robot pose through photogrammetry is established as a viable metrology. Derivative-free forms of the extended Kalman filter are well suited to parameter estimation of non-linear systems such as non-holonomic robots.

The proposed calibration system exhibits several advantages over alternatives. The most significant alternative being sensors installed on the robot that detect targets located throughout the workspace. Many mobile robots have limited computational capability and are often bandwidth limited by low-speed interfaces. The proposed system does not impose any computational load on the robot under calibration. A wheeled mobile robot may be calibrated regardless of kinematic configuration, sensor suite and processing capabilities. Extrinsic calibration requires a single target, composed of multiple fiducial markers, mounted on the robot. This eliminates the requirement to mount multiple camera targets throughout the robot's calibration workspace.

8.2 What accuracy or fidelity is required for a mobile robot kinematic model?

The calibration accuracy requirement for a robot is dependent on the robot application. Chapter 2 cited several existing robot calibration systems that achieved calibration errors of approximately 1 percent. Of note is the work by Roy and Thurn [RT99], where an odometric system is used to calibrate a mobile robot operating in a large indoor environment. The robot is able to navigate throughout the Smithsonian museum with sufficient accuracy to act as a guide by performing periodic calibration. The calibration rate is determined by a navigation error budget. A typical navigation requirement is to maintain a position error less than 10 meters over a 1000 meter run. If odometry is used to measure robot movement, a wheel radius error will directly affect the navigation error. The navigation error is given by $\Delta x = n2\pi\Delta w_R$ where x is robot position, n is the wheel rotation count and w_R is the wheel radius. The wheel radius error is then constrained by $\frac{\Delta w_R}{w_R} \leq \frac{\Delta x}{x}$. Thus the required calibration accuracy is approximately 1 percent.

8.3 Is photogrammetry a suitable measurement scheme for mobile robot calibration?

Calibration experiment conducted yielded a normalized norm distance error of 0.5 percent. This is less then the value identified above. Thus, photogrammetry is a suitable measurement scheme for mobile robot calibration.

8.4 How does calibration accuracy depend on the factors under user control?

Chapter 7 established that calibration error, and hence calibration accuracy, depends on the centroid error position error. The centroid position error is dependent upon apparent marker size in captured images. The apparent marker size in pixels is due to the proximity of the camera to the robot and the intrinsic characteristics of the camera. Specifically, the focal length, and pixel resolution determine the size of the feature in the sampled image. The camera proximity must be such that the entire calibration workspace is within the camera filed of view.

8.5 How can robot calibration accuracy be improved?

The most effective way to improve photogrammetric calibration results is by improving camera capabilities. This can be achieved by increasing the resolution and reducing camera distortion. Camera distortion is reduced by improved optics and sampling electronics.

8.6 What a priori knowledge is necessary for mobile robot calibration?

The user must determine several items in advance of calibration. First, the kinematic model structure must be established as described in Chapter 3. Second a suitable set of control vectors must be determined for the calibration process. Nominal robot parameters are required as an initial point for the divided difference filter. A model of the calibration target is required for the pose estimation algorithm.

8.7 Future Research

The results of this thesis can be improved upon. Several areas of further investigation are suggested.

- Determine the effect of camera quality on calibration performance by experimenting with multiple cameras of variable quality.

- Verify calibration results with other robot configurations such as Ackerman, tricycle and synchro-drive.

- Determine if the introduction of a ground plane constraint into Lowe's pose recovery algorithm improves calibration performance.

- Determine if other forms of derivative-free extended Kalman filters, such as the unscented filter, improve calibration performance.

- Determine if time varying control vectors improve calibration performance.

9. References

[ACB96] H. Araújo, R. L. Carceroni and C. M. Brown. ``A Fully Projective Formulation for Lowe's Tracking Algorithm", TR 641, Comp. Sci. Dept., U. Rochester, Nov 1996.

[ACB97] H. Araújo, R. L. Carceroni and C. M. Brown, ``A Full-Projective Improvement for Lowe's Pose-Estimation Algorithm", in Proc. Darpa Image Understanding Workshop, 1997, Vol. 2, pp. 875-880.

[ACB98] H. Araújo, R. L. Carceroni and C. M. Brown, ``A Fully Projective Formulation to Improve the Accuracy of Lowe's Pose-Estimation Algorithm", in Computer Vision and Image Understanding, Vol. 70, No. 2, pp. 227-238, May 1998 http://www.cs.rochester.edu/u/www/u/carceron/publs/

[ACR02] Acroname Robotics Inc. http://www.acroname.com/robotics/parts/R34-RWPRO.html

[AD98] N.A. Aspragathos and J.K. Dimitros, *A Comparative Study of Three Methods of Robot Kinematics*, IEEE Transactions on Systems, Man, and Cybernetcis – Part B: Cybernetics, Vol 28, No. 2, April 1998.

[Adam93] John A. Adam, "Virtual Reality is for Real", IEEE Spectrum, October, 1993.

[AK71] Y.I. Abdel-Aziz and H.M. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. In Symposium on Close-Range Photogrammetry, pages 1-19, 1971.

[BA93] Bernhardt, Rolf, and Albright, S. L. eds., *Robot Calibration*. London, Chapman. 1993.

[BEF96] Borenstein, J., Everett, B., and Feng, L., 1996, "Navigating Mobile Robots: Systems and Techniques." A. K. Peters, Ltd., Wellesley, MA, ISBN 1-56881-058-X,

[BF94] Borenstein, J. and Feng, L., 1994, "UMBmark A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots." Technical Report, The University of Michigan UM-MEAM-94-22, December 1994. ftp://ftp.eecs.umich.edu/people/johannb/umbmark.pdf

[BF95a] Borenstein, J. and Feng. L., 1995, "Correction of Systematic Odometry Errors in Mobile Robots." Proceedings of the 1995 International Conference on Intelligent Robots and Systems (lROS '95), Pittsburgh, Pennsylvania, August 5-9, 1995, pp. 569-574.

ftp://ftp.eecs.umich.edu/people/johannb/paper59.pdf

[BF95b] Borenstein, J. and Feng. L., 1995, "UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots." Proceedings of the 1995 SPIE Conference on Mobile Robots, Philadelphia, October 22-26, 1995. ftp://ftp.eecs.umich.edu/people/johannb/paper60.pdf

[BF96] Borenstein, J. and Feng. L., 1996, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." IEEE Journal of Robotics and Automation, Vol 12, No 6, December 1996, pp. 869-880. ftp://ftp.eecs.umich.edu/people/johannb/paper58.pdf

[BH96] R.G. Brown and P.Y.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering: with MATLAB Exercises and Solutions* (Third edition): Wiley & Sons, Inc., 1996.

[BLA+99] M. Bak, T. D. Larsen, N. A. Andersen, and O. Ravn. Auto-calibration of systematic odometry errors in mobile robots. In Proceeding of SPIE, Boston, MA, USA, August 1999.

[BLN+98] Martin Bak, Thomas Dall Larsen, Magnus Nørgaard, NilsA. Andersen, Ole Ravn & Niels Kjølstad Poulsen. Location Estimation using Delayed Measurements.International Workshop on Advanced Motion Control, AMC '98.

[BM00] D. Balkcom and M. Mason, Extremal Trajectories for Bounded Velocity Differential Drive Robots, IEEE International Conference on Robotics and Automation (ICRA '00), 2000.

[BMZ92] Paul Beardsley, David Murray, Andrew Zisserman, *Camera Calibration Using Multiple Images*, Computer Vision - ECCV'92, Second Europoean Conference on Computer Vision, Santa Margerita Ligure, Italy, May 19-22, 1992, pp. 312-320.

[CB97] R. L. Carceroni and C. M. Brown, ``Numerical Methods for Model-Based Pose Recovery", TR 659, Comp. Sci. Dept., U. Rochester, Aug 1997. http://www.cs.rochester.edu/u/www/u/carceron/publs/

[Craig89] Craig, John. J., *Introduction to Robotics, Mechanics and Control*. 2ed, Addison-Wesley, 1989.

[JHG00] Bernd Jähne, Horst Haußecker and Peter Geißler, (Editors), *Handbook Of Computer Vision and Applications*, Academic Press, Boston, USA, 2000.

[DD95] D. F. DeMenthon and L. S. Davis. Model based object pose in 25 lines of code. International Journal of Computer Vision, 15:123-141, 1995.

[Deric94] R. Deriche, Z. Zhang, Q.T. Luong, O. Faugeras, *Robust Recovery of the Epipolar Geometry for an Uncalibrated Stereo rig*, Lecture Notes in Computer Science, Vol. 800, Computer Vision – ECCV'94, pp. 567-576, 1994.

[DJ00] Dudek, Gregory and Jenkin, Michael, *Computational Principles of Mobile Robotics*, Cambridge University Press, 2000.

[Ever95] Everett, H.R. (1995). *Sensors for Mobile Robots: Theory and Application*, ISBN 1-56881-048-2, AK Peters, Ltd, Wellesley, MA.

[Everett93] Everett, L.J., *Research Topics in Robot Calibration*, in *Robot Calibration*. Edited by Bernhardt, Rolf, and Albright, S. L. eds., London, Chapman. 1993.

[Everett93] L.J. Everett, *Research Topics in Robot Calibration*, in *Robot Calibration*. Edited by Bernhardt, Rolf, and Albright, S. L. eds., London, Chapman. 1993.

[Eykhoff74] Peter Eykhoff, *System Identification: Parameter and State Estimation*. John Wiley and & Sons, 1974.

[Faig71] W. Faig, *Photogrammetric Measurements of Hydraulic Surfaces*, Proceedings of the 37th Annual Meeting of the American Society of Photogrammetry, March, 1971.

[Faig72] W. Faig, *Single Camera Approaches in Close-Range Photogrammetry*, Proceedings of the 38th Annual Meeting of the American Society of Photogrammetry, March, 1972.

[FB94a] Feng, L. Koren, Y., and Borenstein, J., 1994e, "A Model-Reference Adaptive Motion Controller for a Differential-Drive Mobile Robot." Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, May 8-13, 1994, pp. 3091-3097. ftp://ftp.eecs.umich.edu/people/johannb/paper51.pdf

[FB94b] Feng, L. Koren, Y., and Borenstein, J., 1994e, "A Model-Reference Adaptive Motion Controller for a Differential-Drive Mobile Robot." Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, May 8-13, 1994, pp. 3091-3097. ftp://ftp.eecs.umich.edu/people/johannb/paper51.pdf

[FLM92] O.D. Faugera, Q.T. Luong, S.J. Maybank, *Camera Self-Calibrations Theory and Experiments*, Computer Vision - ECCV'92, Second Europoean Conference on Computer Vision, Santa Margerita

Ligure, Italy, May 19-22, 1992, pp. 321-334.

[Garcia98] Christophe Garcia, *Complete Calibration of the Autonomous Hand-Eye Robot JANUS.* February 1998, 56 pages, GMD Technical Report-002. http://www.gmd.de/publications/report/0002/Text.pdf

[Garcia98] Christophe Garcia, *Complete Calibration of the Autonomous Hand-Eye Robot JANUS.* February 1998, 56 pages, GMD Technical Report-002. http://www.gmdn.de/publications/report/0002/Text.pdf

[Georg99] N.D. Georganas, E. Petriu, M. Cordea, D. Ionescu, "Distributed Virtual Environments for Training and Telecollaboration," Proc. IMTC/99, IEEE Instrum. Meas. Technol. Conf., (to appear) Venice, Italy, 1999. http://www.site.uottawa.ca/~petriu/publication.htm

[Grosky90] W.I. Grosky and L.A. Tamburino. *A unified approach to the linear camera calibration problem.* IEEE Trans. Pattern Analysis and Machine Intelligence, 12(7):663-671, July 1990.

[GW93] R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addision-Wesley, 1993.

[Heik96] Heikkila, J. & Silven O., *Calibration procedure for short focal length off-the-shelf CCD cameras*, In: Proc of The 13th International Conference on Pattern Recognition, Vienna, Austria. pp. 166-170, 1996.

[Heik97] Heikkilä, J. & Silvén, O., *A Four-step Camera Calibration Procedure with Implicit Image Correction,*.IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico, 1997.

[Heik99] Janne Heikkila, *CAMERA CALIBRATION TOOLBOX FOR MATLAB (v2.1b),* University of Oulu, Department of Electrical Engineering, FIN-90570 Oulu, Finland. http://ee.oulu.fi/~jth.

[HS97] Heikkilä, J. & Silvén, O., *A Four-step Camera Calibration Procedure with Implicit Image Correction,* IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), San Juan, Puerto Rico, 1997.

[HTR88] S. Hayati, K. Tso and G. Roston, "Robot geometry calibration", Proceedings of the 1988 IEEE International conference on robotics and automation, pp. 947-951

[IX00] Kazufumi Ito and Kaiqi Xiong, "Gaussian Filters for Nonlinear Filtering Problems," *IEEE Transactions on Automatic Control*, vol. 45, no. 5, pp. 910–927, may 2000.

[JH00] Bernd Jahne and Horst Haussecker *Computer Vision and Applications: A Guide for Students and Practitioners.* Academic Press, 2000.

[JU97] S. J. Julier and J. K. Uhlmann, *A New Extension of the Kalman Filter to Nonlinear Systems*, in *Proc. of AeroSense: The 11th Int. Symp. A.D.S.S.C.*, 1997.

[JUW95] S. Julier, J. Uhlmann, and H. Durrant-Whyte, “A new approach for filtering nonlinear systems,” in *Proceedings of the American Control Conference*, 1995, pp. 1628–1632.

[Kalman60] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D), 35-45, 1960.

[Kara74] H.M. Karara and Y.I. Abdel-Aziz, *Accuracy Aspects of Non-Metric Imageries*. Photogrammetric Engineering, Vol. 40, No. 9, pages 1107-1117, 1974.

[Kara79] H.M. Karara, Ed., *Handbook of Non-Topographic Photogrammetry*, Amer. Soc. of Photogrammetry, 1979.

[Kay93] S.M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall Signal Processing Series, 1993.

[Kelly94] Kelly, Alonzo (1994). *Essential Kinematics for Autonomous Vehicles*. CMU-RI TR-94-14, Carnegie Melon University, Robotics Institute.

[Kelly94] Kelly, Alonzo (1994). *Essential Kinematics for Autonomous Vehicles*. CMU-RI TR-94-14, Carnegie Melon University, Robotics Institute.

[Kelly94] Kelly, Alonzo (1994). *Essential Kinematics for Autonomous Vehicles*. CMU-RI TR-94-14, , Carnegie Melon University, Robotics Institute.

[Kopp98] Sunil K Kopparapu and Peter Corke, *The Effect of Measurement Noise on Camera Calibration Matrix,* IVCNZ 98, Auckland, New Zealand, November, 1998. http://www.cat.csiro.au/automation/staff/nil/csiro/

[Kopp99] Sunil K Kopparapu and Peter Corke, *The Effect of Measurement Noise on Intrinsic Camera Calibration Parameters*, International Conference on Robotics and Automation, ICRA 99. http://www.cat.csiro.au/automation/staff/nil/csiro/

[Larsen99a] T. D. Larsen, K. L. Hansen, N. A. Andersen, and O. Ravn. Design of kalman filters for mobile robots; evaluation of the kinematic and odometric approach. In Proceeding of IEEE Conference on Control Applications, CCA'99, Hawaii, USA, August 1999.

[Larsen99b] T. D. Larsen, N. A. Andersen, and O. Ravn. A new approach for kalman filtering on mobile robots in the presence of uncertainties. In Proceeding of IEEE Conference on Control Applications, CCA'99, Hawaii, USA, August 1999.

[Larsen99c] T. D. Larsen, K. L. Hansen, N. A. Andersen, and O. Ravn. Design of kalman filters for mobile robots; evaluation of the kinematic and odometric approach. In Proceeding of IEEE Conference on Control Applications, CCA’99, Hawaii, USA, August 1999.

[Larsen99d] T. D. Larsen, N. A. Andersen, and O. Ravn. A new approach for kalman filtering on mobile robots in the presence of uncertainties. In Proceeding of IEEE Conference on Control Applications, CCA’99, Hawaii, USA, August 1999.

[Laumond98] Jean-Paul Laumond. Robot Motion Planning and Control, Lecture Notes in Control and Information Sciences, 229, 1998.

[Lee00] Ten-min Lee, "A New Approach to Mobile Robot Simulation By Means Of Acquired Neural Network Models". Ph.D Thesis, The University of Manchester, Manchester, England, U.K., March, 2000. http://www.cs.man.ac.uk/~leet/PhD_thesis.html

[Lee00] Ten-min Lee, "A New Approach to Mobile Robot Simulation By Means Of Acquired Neural Network Models". Ph.D Thesis, The University of Manchester, Manchester, England, U.K., March, 2000. http://www.cs.man.ac.uk/~leet/PhD_thesis.html

[Lowe80] David Lowe. *Solving for the parameters of object models from image descriptions*. In Proc. ARPA Image Understanding Workshop, pages 121{127, College Park, MD, Apr 1980.

[Lowe87] David Lowe. *Three-dimensional object recognition from single two-dimensional images*. Artificial Intelligence, 31(3):355{395, Mar 1987.

[Lowe91] David Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441-450, May 1991.

[Luong94] Q.T. Luong, O. Faugeras, *A Stability Analysis of the Fundamental Matrix*, Lecture Notes in Computer Science, Vol. 800, Computer Vision -

ECCV'94, pp. 577-588, 1994.

[McK91] McKerrow, P.J. (1991). *Introduction to Robotics*. Addison-Wesley.

[Meng94] Mengxiang Li, *Camera Calibration of a Head-Eye System for Active Vision*, Lecture Notes in Computer Science, Vol. 800, Computer Vision - ECCV'94, pp. 543-554, 1994.

[Mettler00] B. Mettler, M. Tischler, and T. Kanade ,System Identification of a Model-Scale Helicopter, tech. report CMU-RI-TR-00-03, Robotics Institute, Carnegie Mellon University, January, 2000. http://www.ri.cmu.edu/cgi-bin/tech_reports.cgi

[MN86] P.F. Muir and C.P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," Robotics Institute Technical Report No. CMU-RI-TR-86-12, Carnegie Mellon University, Pittsburgh, PA 15213, June 1986.

[MN87] P.F. Muir and C.P. Neuman, "Kinematic Modeling of Wheeled Mobile Robots," Journal of Robotic Systems, Vol. 4, No. 2, April 1987, pp.281-340.

[MPS01] L.P. McNamee, *E.M. Petriu*, and H.J.W. Spoelder, "Photogrammetric Calibration of a Mobile Robot Model," *Proc. IMTC/2001, IEEE Instrum. Meas. Technol. Conf.*, pp.245-250, Budapest, Hungary, May 2001.

[Muir90] P. Muir, "A Virtual Sensor Approach to Robot Kinematic Identification," Proceedings of the 1990 IEEE International Conference on Systems Engineering, August, 1990, pp. 440-445.

[MW01] R. van der Merwe and E. A. Wan, "Efficient Derivative-Free Kalman Filters for Online Learning", in *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, Apr, 2001. http://cslu.cse.ogi.edu/publications/ps/merwe01.pdf

[Nehmzow99a] Ulrich Nehmzow, Mobile Robotics: A Practical Introduction, Springer Verlag 1999. ISBN 1-85233-173-9.

[Nehmzow99b] Ulrich Nehmzow, Mobile Robotics: A Practical Introduction, Springer Verlag 1999. ISBN 1-85233-173-9.

[Nor00] M. Nørgaard. "KALMTOOL - State Esimation for Nonlinear Systems" Technical Report. IMM-REP-2000-6, Department of Mathematical Modelling, Technical University of Denmark, 2000.

[NPR00a] M. Nørgaard, N.K. Poulsen, O. Ravn: Advances in Derivative-Free State Estimation for Nonlinear Systems, Technical report IMM-REP-

1998-15, Department of Mathematical Modelling, DTU, 1998 (revised April7, 2000)

[NPR00b] M. Nørgaard, N.K. Poulsen, O. Ravn: *New Developments in State Estimation for Nonlinear Systems,* Automatica, (36:11), Nov. 2000, pp. 1627-1638.

[NPR97a] M. Nørgaard, N.K. Poulsen, O. Ravn: Autonomous Guided Vehicle: Modelling, Technical Report IMM-REP-1997-25, Department of Mathematical Modelling, Technical University of Denmark, 1997.

[NPR97b] M. Nørgaard, N.K. Poulsen, O. Ravn: Autonomous Guided Vehicle: Sensor Analysis, Technical Report IMM-REP-1997-26, Department of Mathematical Modelling, Technical University of Denmark, 1997.

[NPR99] M. Nørgaard, N. K. Poulsen, and O. Ravn. Easy and accurate state estimation for nonlinear systems. In Proceeding of IFAC World Congress, Beijing, P.R.China, July 1999.

[Petriu99] E. Petriu, M. Cordea, D.C. Petriu, Lou McNamee, *Modeling Issues in Virtual Prototyping Environments*, Proc. VIMS'99, IEEE Workshop on Virtual and Intelligent Measurement Systems, Venice, Italy, 1999.

[RT99] N. Roy and S. Thrun, "Online Self-Calibration For Mobile Robots," *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[SBH93] G. Stark, E. Benz and M. Huttenhofer, *Calibration Experiences in Industry*, in *Robot Calibration*. Edited by Bernhardt, Rolf, and Albright, S. L. eds., London, Chapman. 1993.

[SBH93] G. Stark, E. Benz and M. Huttenhofer, *Calibration Experiences in Industry*, in *Robot Calibration*. Edited by Bernhardt, Rolf, and Albright, S. L. eds., London, Chapman. 1993.

[Scheider95] Schneider, Jeff G., *Robot Skill Learning Through Intelligent Experimentation*, TR567 and Ph.D. Thesis, Computer Science Dept., U. Rochester, January 1995. ftp://ftp.cs.rochester.edu/pub/papers/robotics/95.tr567.ps.gz

[Schneider95] Schneider, Jeff G., *Robot Skill Learning Through Intelligent Experimentation*, TR567 and Ph.D. Thesis, Computer Science Dept., U. Rochester, January 1995. ftp://ftp.cs.rochester.edu/pub/papers/robotics/95.tr567.ps.gz

[Shamah99] B. Shamah , Experimental Comparison of Skid Steering Vs.

Explicit Steering for a Wheeled Mobile Robot, Master's thesis, tech. report CMU-RI-TR-99-06, Robotics Institute, Carnegie Mellon University, March, 1999. http://www.ri.cmu.edu/cgi-bin/tech_reports.cgi

[Sobel74] I. Sobel *On Calibrating Computer Controlled Cameras for Perceiving 3-d Scenes,* Artificial Intelligence. An International Journal **5**, 185-198, 1974.

[Stein98] M. C. Steinbach, H. G. Bock, G. V. Kostin, and R. W. Longman: Mathematical Optimization in Robotics: Towards Automated High Speed Motion Planning. Surv. Math. Ind. 7(4), 303-340, 1998. [Preprint]

[SU71] P.N. Sheth and J.J. Uicker. *A Generalized Symbolic Notation for Mechanisms*, Journal of Engineering for Industry, Series B, **93**, 70-Mech-19, pp. 102-112.

[Tani96] Jun Tani, *Model-based learning for mobile robot navigation from the dynamical system*s. IEEE Transactions on Systems, Man, and Cybernetics-Part B Vol.26, No.3, June 1996Special Issue on Learning Autonomous Robots

[Tani96] Jun Tani, *Model-based learning for mobile robot navigation from the dynamical systems***.** IEEE Transactions on Systems, Man, and Cybernetics-Part B Vol.26, No.3, June 1996Special Issue on Learning Autonomous Robots

[Trev96] J. Trevelyan, "Robot Calibration with an Extended Kalman Filter", ", Department of Mechanical and Materials Engineering, University of Western Australia, 1996.

[Tsai87a] Roger Y. Tsai, *A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987, pages 323-344.

[WB01] Greg Welch and Gary Bishop. *An Introduction to the Kalman Filter*. A tutorial presented at SIGGRAPH 2001, Los Angles, CA, August 12-17, 2001.

[Wei94] G.Q. Wei and S.D. Ma, *Implicit and Explicit Camera Calibration - Theory and Experiments. IEEE Trans. PAMI, 16(5): 469-480, May 1994.*

[Weiss99] Eric W. Weisstein. *CRC Concise Encyclopedia of Mathematics*, CRC Press, 1999.

[Weng92] Juyang Weng, Paul Cohen, Marc Herniou, *Camera Calibration with Distortion Models and Accuracy Evaluation*, IEEE Transactions on

Pattern Analysis and Machine Intelligence, Vol.14, No.10, 1992, pp.965-980.

[Wong76] K.W. Wong, *Mathematical Formulation and Digital Analysis in Close-Range Photogrammetry*, Invited Paper, Comm. V, ISP Congress, Helsinki, 1976. (PE&RS, Nov. 1975).

[Wren97] Christopher Wren et. al., *Perceptive Spaces for Performance and Entertainment*, Applied Artificial Intelligence, Vol. 11, No. 4, June 1997.

[Yuan89] Joseph S.C. Yuan, *A General Photogrammetric Method for Determining Object Position and Orientation*, IEEE Transactions on Robotics and Automation, Vol. 5, No. 2, April 1989.