Intelligent Fastening Tool Tracking Systems Using Hybrid Remote Sensing Technologies

by

Peter S. Won

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of

Doctor of Philosophy
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2010

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.
I understand that my thesis may be made electronically available to the public.

Abstract

This research focuses on the development of intelligent fastening tool tracking systems for the automotive industry to identify the fastened bolts. In order to accomplish such a task, the position of the tool tip must be identified because the tool tip position coincides with the head of the fastened bolt while the tool fastens the bolt. The proposed systems utilize an inertial measurement unit (IMU) and another sensor to track the position and orientation of the tool tip.

To minimize the position and orientation calculation error, an IMU needs to be calibrated as accurately as possible. This research presents a novel triaxial accelerometer calibration technique that offers a high accuracy. The simulation and experimental results of the accelerometer calibration are presented.

To identify the fastening action, an expert system is developed based on the sensor measurements. When a fastening action is identified, the system identifies the fastened bolt by using an expert system based on the position and orientation of the tool tip and the position and orientation of the bolt. Since each fastening procedure needs different accuracies and requirements, three different systems are proposed.

The first system utilizes a triaxial magnetometer and an IMU to identify the fastened bolt. This system calculates the position and orientation by using an IMU. An expert system is used to identify the initial position, stationary state, and the fastened bolt. When the tool fastens a bolt, the proposed expert system detects the fastening action by triaxial accelerometer and triaxial magnetometer measurements. When the fastening action is detected, the system corrects the velocity and position error using zero velocity update (ZUPT). By using the corrected tool tip position and orientation, the system can identify the fastened bolts. Then, with the fastened bolt position, the position of the IMU is corrected. When the tool is stationary, the system corrects linear velocity error and reduces the position error. The experimental results demonstrate that the proposed system can identify fastened bolts if the angles of the bolts are different or the bolts are not closely placed. This low cost system does not require a line of sight, but has limited position accuracy.

The second system utilizes an intelligent system that incorporates Kalman filters (KFs) and a fuzzy expert system to track the tip of a fastening tool and to identify the fastened bolt. This system employs one IMU and one encoder-based position sensor to determine the orientation and the centre of mass location of the tool. When the KF is used, the orientation error increases over time due to the integration step. Therefore, a fuzzy expert system is developed to correct the tilt angle error and orientation error. When the tool fastens a bolt, the system identifies the fastened bolt by applying the fuzzy expert system. When the fastened bolt is identified, the 3D orientation error of the tool is corrected by using the location and the orientation of the fastened bolt and the position sensor outputs. This orientation correction method results in improved reliability in determining the tool tip location. The fastening tool tracking system was experimentally tested in a lab environment, and the results indicate that such a system can successfully identify the fastened bolts. This system not only has a low computational cost but also provides good position and orientation accuracy. The system can be used for most applications because it provides a high accuracy.

The third system presents a novel position/orientation tracking methodology by hybridizing one position sensor and one factory calibrated IMU with the combination of a particle filter (PF) and a KF. In addition, an expert system is used to correct the angular velocity measurement errors. The experimental results indicate that the orientation errors of this method are significantly reduced compared to the orientation errors obtained from an EKF approach. The improved orientation estimation using the proposed method leads to a better position estimation accuracy. The experimental results of this system show that the orientation of the proposed method converges to the correct orientation even when the initial orientation is completely unknown. This new method was applied to the fastening tool tracking system. This system provides good orientation accuracy even when the gyroscopes (gyros hereafter) include a small error. In addition, since the orientation error of this system does not grow over time, the tool tip position drift is limited. This system can be applied to the applications where the bolts are closely placed. The position error comparison results of the second system and the third system are presented in this thesis. The comparison results indicate that the position accuracy of the third system is better than that of the second system because the orientation error does not increase over time.

The advantages and limitations of all three systems are compared in this thesis. In addition, possible future work on fastening tool tracking system is described as well as applications that can be expanded by using the KF/PF combination method.

Acknowledgements

First of all, I cannot thank God enough for his step by step guidance. Without his help, I could not even have begun studying. Also, I am very grateful that he sent me many people to help and support me.

Among those people, I would like to express my gratitude to my two supervisors, Prof. William Melek and Prof. Farid Golnaraghi. With their continuous support, guidance, and advice, not only I have improved my research ability, but also I have achieved great improvement on my personality.

I would like to thank all defence committee members, Prof. Glenn Helppler, Prof. Robert Gorbet, Prof. Jan Huissoon, and the external examiner Prof. Guangjun Liu. They read my thesis very carefully and gave me good advice to make my thesis stronger. I also want to thank my colleagues, especially Dr. Orang Vahid and Mr. Ying Kun Peng, for the help, support, and friendship.

Last, but not least, I really want to express my appreciation to my parents, my parents-in-law, and most of all, my wife, I-yung for their prayer, love, and endless support.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Motivation

Traditionally, motion tracking systems have been used for outdoor applications such as vehicle and missile tracking [1]-[3] because traditional inertial sensors were bulky and heavy. Recently, various motion tracking systems have been developed and used for indoor applications such as manufacturing [4], human supporting systems [5], [6], and rehabilitation [7]. This thesis describes developments of tool tracking systems that identify fastened bolts as a quality control system.

Quality control has been a primary focus in the automotive industry. However, many automotive parts are still produced without any quality control process. For instance, the fastening process is necessary in various assembly lines such as engine mount, air bags, and seat assembly. This process requires fastening bolts in the right place with the right amount of torque. Currently, human operators can set the output torque of the tool and monitor the fastening process to ensure that each bolt is fastened through the application of a torque value within the desired range. However, monitoring the torque values alone does not guarantee that each bolt is fastened in the right places because operators can make mistakes by not fastening all the bolts or fastening a bolt in the wrong sequence. Therefore, these assembly flaws in the fastening process should be significantly reduced to produce safer and higher quality automotive parts. To eliminate any potential mistakes and correctly fasten bolts, a quality control system that tracks the location of the tool tip where bolts are placed during fastening is required.

Currently, only two fastening tool position tracking devices are available in the market; namely, SmartArm and UOS-100. SmartArm from PINpoint Information Systems Inc. relies on a passive robotic arm without an actuator [8]. SmartArm consists of various links, and each link has an encoder that tracks the angle of the link. When an operator moves a tool in the workspace, the encoders of the passive robot arm track the motion of each link, and the position and the orientation of the tool are calculated by using kinematics and the geometry of the robot. This passive robot arm can accurately track the position of the tool tip, but it can also easily limit the movement of the operators working on the assembly process.

The other product, UOS-100 from Pepperl+Fuchs, uses ultrasonic transducers to track the position of the tool tip [9]. This system consists of one ultrasonic emitter, at least three receivers, and one control interface unit. The ultrasonic emitter is attached to the tip of a fastening tool, and the ultrasonic receivers are placed on three different fixed locations such as ceilings. Since this system locates the ultrasonic emitter by triangulation method, ultrasonic receivers should be placed some distance apart from each other. By measuring the distance between one emitter and three receivers, the location of the tool tip in 3D space (x, y, and z) is determined. The advantage of this system is that it does not interfere with the movements of the operator because only a small light-weighted ultrasonic emitter is attached to the tool tip. However, an ultrasonic sensor has limited applications due to maximum emitting angle, reflection, occlusions [10], and sound sensitivity [11]. Since the ultrasonic position sensor uses triangulation method to locate the 3D position of the ultrasonic emitter, the lines of sight between the emitter and the receivers are critical. For example, when the receivers are installed on flat ceilings, UOS-100 can only detect the fasteners that have close to zero tilt angles in order to obtain the lines of sight. When at least three lines of sight are not secure, the system fails to locate the position of the tool tip. In addition, UOS-100 is not practical for a noisy factory environment because of its sensitivity to high frequency acoustics such as metal beating.

In summary, the tracking systems that make use of those two products are limited. Therefore, a compact tool tracking system that has high dexterity and robustness is needed for automated tracking and the quality inspection of parts assembled by bolt fastening actions.

1.2 Literature Review

1.2.1 IMU Calibration

Accelerometers and gyros include several errors such as nonlinearity, gain error, and bias, as a result of temperature change, input voltage and aging [12]-[14]. Nonlinearity is not usually compensated for, especially for a low cost IMU, because the process of modeling nonlinearity is complex and the resultant error is often very small. Thus, IMU calibration typically deals with estimating gains and biases. The gain and bias values depend significantly on temperature as shown in [12] and [15]. Even with the temperature error compensation, the IMU error depends on the input voltage. Therefore, the input voltage is usually controlled by a voltage regulator. However, the supplied voltage varies even if a voltage regulator is used [12]. Therefore, when an IMU is powered, it is commonly observed that the current gains and biases differ slightly from the previous values even if the IMU is powered off and immediately powered on. To reduce the gain and bias errors, an IMU should be calibrated whenever it is powered on.

Due to the need for frequent recalibration of an IMU, the calibration method should follow a simple procedure and must provide a high accuracy. Section 1.2.1.1 reviews the calibration techniques of a triaxial accelerometer, and Section 1.2.1.2 provides the background of triaxial gyro calibration techniques.

1.2.1.1 Triaxial Accelerometer Calibration Techniques

The conventional method of triaxial accelerometer calibration involves rotating an accelerometer at known tilt angles [16]-[18] to use the gravitational vector as a reference. To achieve accurate results with this method, the tilt angles must be precisely measured. Calibration methods with an external device such as an actuator [19], [20] or a position sensor [21], [22] can accurately compute the gains and biases, but such a device is not usually available outside of a laboratory. Accelerometer calibration methods that do not require known tilt angles or an external device have been presented in [23]-[28]. The calibration method in [23] continuously estimates gains and biases of a triaxial accelerometer to calculate the tilt angle of human body parts. This method requires prior knowledge of the frequencies of human body movements, and assumes that the acceleration of a body segment has a zero mean to calibrate a triaxial accelerometer. However, the accuracy of this method depends on how true the assumptions are. For example, this method has a higher accuracy when the accelerometer is attached to the pelvis rather than the trunk. Other methods use a least squares estimation to calculate the gains, biases, and misalignment errors [24], [25]. The aforementioned method allows redundant tilt angle measurements to achieve more accurate calibration results. However, Syed *et al.* [25] have stated that the initial estimation of the gains and biases need to be close to the true values to converge to reasonable gains and biases to use the method in [24]. To find the rough estimates of gains and biases, a calibration procedure in [25] roughly aligns the three axes of the sensor with the gravity vector once positively and once negatively. Calibration methods rely on the Taylor series expansion up to the first order term to linearize the nonlinear mathematical model of the gains and biases are presented in [26]-[28]. Lai *et al.* [26] have reported a method to find three gains and three biases of a triaxial accelerometer by placing it in six randomly chosen tilt angles in a stationary state. The experimental results show that this method usually requires five iterative steps to estimate the gains and biases. Lai *et al.* [26] have also stated that this method also requires the initial estimations of the gains and biases so that the true values do not diverge from the correct solution.

1.2.1.2 Gyro Calibration Techniques

The conventional calibration method rotates each gyro about its axis at various constant angular velocities, and then, the relationship between the gyro outputs and the angular velocity measurement are established [18]. To achieve accurate calibration results with this method, the rotated axis must be perfectly parallel to the calibrating gyro axis. Since this method requires correct angular velocity measurements, a turntable which can measure angular velocity is used for calibration in [24], [29]. However, the procedures of these methods are inconvenient and time consuming. The calibration method that utilizes an optical position sensor with three light-emitting-diodes (LEDs) has presented a faster and simpler procedure [21]. This method can calibrate an IMU by randomly moving the sensor for half a minute. Then, by using the orientation measurements calculated from three LED positions, the IMU is calibrated. Although this method is easy to use, the optical sensor is relatively large and expensive. Consequently, this method is not feasible outside the laboratory. Another calibration method [16] places the gyro stationary to estimate the bias and rotates the gyro about its rotation axis 360° to calibrate the gain.

1.2.2 Position and Orientation Computation Using an IMU

An IMU can be used to estimate orientation [30]–[32] and position [33]. Assuming that the local gravity vector is perfectly known at all times, the position and the orientation of an object can be accurately calculated by numerically integrating the linear acceleration and angular velocity measured by an ideal IMU. However, an ideal IMU that has continuous and perfectly accurate measurements does not exist in the real world. Therefore, the position error increases over time due to the numerical integration of corrupted inertial sensor data. Figure 1-1 depicts the position error simulations of different grades of stationary IMUs with small biases [11]. The simulations indicate that the position error reaches 450 mm after 7 seconds with a commercial grade IMU, 10 seconds with a tactical grade IMU, and 90 seconds with a navigation grade IMU. Figure 1-1 shows that position tracking systems using an IMU only are not reliable for an extended period of time. In order to estimate the position, the acceleration needs to be integrated twice and the angular velocity needs to be integrated once. However, to estimate orientation, the angular velocity needs to be integrated once. This means the orientation error grows at a much slower rate than the position error. Therefore, an IMU is often used as an orientation sensor, otherwise is usually hybridized with another sensor to estimate the position [34], [35].

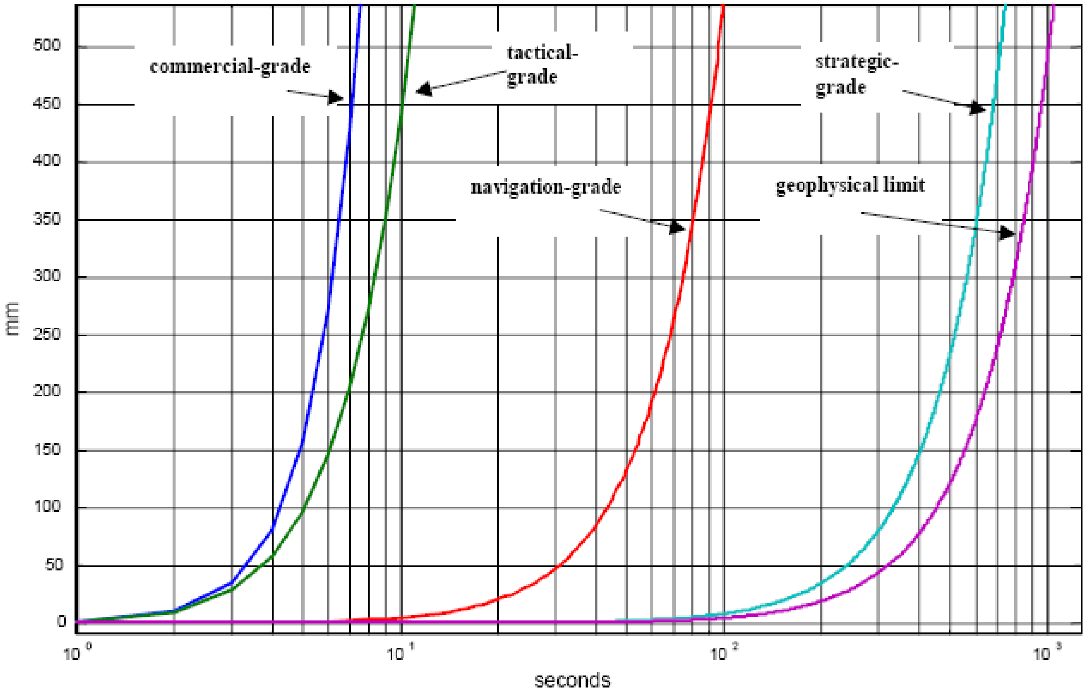


Figure 1-1: Position error simulations of different grades of IMUs [11]

Using an IMU as an orientation sensor is still a challenge because the orientation calculation drifts over time due to the integration of gyro errors [36]. In order to overcome this problem, three accelerometers of an IMU are used as a tilt angle sensor because the accelerometer can find the tilt angle without any integration step. An orientation correction method using a triaxial accelerometer is described in [37], but the experimental results of this method show a high orientation error. Many researchers have employed the KF to combine the tilt angles from accelerometer measurements with the orientation calculation from gyro measurements [38]-[40]. Rehbinder and Hu [40] have utilized gyros to find the angular position of a robot and correct the tilt angle with accelerometers when the robot does not accelerate. Fuzzy expert systems are chosen to detect the static state of an object to correct the tilt angles [41]-[43].

Although these techniques can limit the drift in the roll and pitch angles, they cannot correct the yaw angle which does not depend on the tilt angles. In order to correct the yaw angle, three magnetometers are used to measure the magnetic field of the earth [44], [45]. However, the magnetic field of the earth can be corrupted by ferrous materials [46] and the magnetic field can be generated by electronic devices. Bachmann *at el.* [47] have reported that when the ferrous material is less than two feet from a magnetometer, the error due to the distortion of the magnetic field can be significant. To reduce the distortion of the magnetic field of the earth, methods that combine IMU outputs with magnetometers using a KF have been proposed [48]-[50]. The accuracy of such methods decreases if the disturbance has a similar low bandwidth to that of the angular drift encountered by the gyro error or if the magnetic field is constantly disturbed.

1.2.3 Position Sensors

Various position sensing technologies are available and can be categorized as follows: (i) visual-based, (ii) non-visual-based, and (iii) encoder-based. Each group has its own advantages and limitations.

Visual-based position sensors require a detecting system to be in the line of sight of the object or the marker to obtain measurements (e.g. infrared, camera, and ultrasonic position sensor). An infrared position tracking system can achieve less than 1 mm root mean square (RMS) error when the marker is less than 2 m from the sensor [51]-[54]. However, an accurate infrared position sensor such as Optotrak is very expensive (over $150,000). The accuracy of a camera-based position sensor depends on various factors such as the distance between the object and the camera(s). Thus, the position error range varies but [55] shows that stereo vision system can achieve a 5 mm error. However, the camera-based position estimation techniques still require further improvement [56] and a complex calibration procedure [57]. In addition, the accuracy of an ultrasonic position sensor varies depending on the distance between the marker and the receiver. The ultrasonic-based fastening tool tracking system, UOS-100, has a position error less than 10 mm. However, the ultrasonic position sensors have performance issues related to reflections, occlusions, and sound sensitivity as discussed in Section 1.1.

Non-visual-based position sensors do not require lines of sight, but usually have lower accuracy. Non-visual-based position sensors include IMU, magnetic position sensor [58], radio frequency (RF) [59]-[61], and ultra-wideband (UWB) [62]. Magnetic position sensors can achieve accuracy better than a 10 mm error [63], [64] when no ferrous material is in the measuring range. However, when ferrous material is in the measuring range, the magnetic field is distorted [47], and the position error can be significantly increased depending on the ferrous material and the distance between the material and the sensor [64], [65]. The position estimation using the RF technology does not require lines of sight, but introduces undesirable errors when occlusions and reflections occur [66], [67]. UWB is suitable for in-door tracking applications such as body tracking or shipping container tracking because UWB can penetrate various wall materials [68]. However, when a human body obstructs the line of sight between the tag and the sensor, a UWB position sensor cannot locate the tag due to the signal strength attenuation by human tissue [69]. In addition, the UWB signal can be reflected by metal.

A string-encoder position sensor that can track 3D position has been introduced [70]. This method uses one encoder and three force sensors to estimate the position of the end-point of the wire that is connected to the sensor. However, the friction between the wire and the guidance hole creates a measurement error in the force sensor, and significant noise is observed when the position is calculated from the force sensor measurements. Also, an product that utilizes three encoders to find the 3D position is available [71]. This position sensor has a single wire which is connected to three encoders to find the position of the other end of the wire. This product has a high accuracy and provides noise-free outputs. However, it still needs a line of sight due to the existence of the string.

The summary of the position sensors and their advantages and limitations are described in Table 1-1. Since each position sensor has its own characteristics, selecting a position sensor depends on the required accuracy and the environment of the application.

Table 1-1: Position sensors and their advantages and limitations.

Sensor	Accuracy	Advantages	Limitations
RF	3000 mm	No LOS (line of sight)	Poor accuracy, strength attenuation, reflection
UWB	500 mm	Ability to measure behind walls	Poor accuracy, strength attenuation, reflection
Magnetic	10 mm	No LOS, high accuracy	Ferrous materials limit accuracy
Ultrasonic	10 mm	High accuracy	3 LOS, Noise
Infrared	1 mm	High accuracy	3 LOS, High cost
Camera	5 mm	High accuracy	Complexity, 2 LOS
Encoder	7 mm	High accuracy	LOS, longer length results in poor accuracy

1.2.4 Orientation Detection Using Position Sensors

By attaching multiple position markers on the tracking object, drift-free orientation estimation can be achieved. A multi-antenna GPS receiver has been utilized to find the orientation of a vehicle [72], [73], and multiple ultrasonic markers have been used to find the orientation [74], [75]. Multiple position markers are integrated with one IMU to obtain a drift-free position and orientation estimation [76], [77]. The position and orientation can be estimated more accurately by hybridizing an IMU with a position sensor. In addition, an IMU is used to estimate position and orientation when the position sensor data are not available. When position sensors are utilized to estimate the orientation, the position markers should be attached to a rigid body, and the markers should be located some distance apart from each other to obtain meaningful orientation measurements. However, this requirement is not feasible in many applications due to small object size or the restrictions of applications.

1.3 Thesis Overview

The primary objective of this thesis is to develop framework for intelligent remote position/orientation systems. Since each application has its own needs and limitations, the newly developed framework requires specific design considerations in relation to the application. The immediate goal of this research is to develop tool tracking systems that can identify the fastened bolt for an automotive manufacturing environment. The proposed methods integrate an IMU with an additional sensor to identify the fastened bolt. A MEMS IMU is chosen for this research because it is small and lightweight. Therefore, it can be attached to a tool without limiting the movement of the operator. This thesis consists of seven chapters.

In Chapter 2, various hybridization techniques using variants of the Bayesian filter are described as well as the fundamental concepts of fuzzy expert system.

Chapter 3 describes a novel triaxial accelerometer calibration technique that does not require any external sensor. The calibration parameters of an accelerometer change slightly whenever the sensor is powered on or when the temperature of the sensor is changed. Since the presented calibration method has a very simple calibration procedure and a high accuracy, it can easily be applied by operators in the automotive industry.

Chapter 4 presents a fastened bolt tracking system using an IMU and a triaxial magnetometer. This chapter describes how to calculate the position and orientation by using an IMU. Also, an expert system that identifies stationary state, fastening action, and initial position is presented. The expert system also corrects or reduces position and orientation errors.

Chapter 5 presents a fastening tool tracking system by using an IMU and an encoder-based position sensor. This tracking system hybridizes one position sensor and one IMU to estimate position and orientation. In addition, a fuzzy expert system is developed to identify the stationary state and the fastened bolts.

Chapter 6 presents a hybridization technique which combines a PF and a KF. This novel technique estimates orientation using a PF and estimates the position using a KF. An extensive analysis of the experiments is conducted. This novel method is tested and the results are compared with those using the KF-based method in Chapter 5.

1.4 Contributions

The contribution of this research is as follows.

- Development of a novel triaxial accelerometer calibration.
- Development of a tool tracking system by using an IMU and a triaxial magnetometer.
- Design of an expert system that identifies the fastening action, steady state, and orientation correction.
- Design a fuzzy expert system that identifies the stationary state and fastened bolt identification.
- Development of a tool tracking system using an IMU and encoder-based position sensor.
- Development of the framework of a position/orientation tracking system that combines the PF and the KF.
- Development of a tool tracking system that utilizes the KF-PF combination and an expert system.

Chapter 2
Theoretical Preliminaries

2.1 Hybridization Techniques

Even for applications that require only position estimation, many recent tracking systems depend on a hybrid hardware consisting of one IMU and one position sensor to improve the position estimation accuracy rather than using a standalone position sensor. When an IMU is integrated with a position sensor, the drawbacks of each sensor are accounted for, and more accurate state estimations can be achieved. To integrate one position sensor with one IMU, the variants of the Bayes filter [78], [79] such as the KF [80], [81] or the PF [82], [83] are widely used. A KF is an optimum observer that estimates the states of linear Gaussian state space models. KF and its variants such as the extended Kalman filter (EKF) [84], unscented KF [85], and complementary KF [86], [87] are the most commonly used filtering techniques to integrate an IMU with a position sensor. When the model is highly nonlinear or the noise distribution is non-Gaussian, a PF is more suitable because a PF does not require the state space model to be linear nor assume the noise is zero mean Gaussian. In addition, even when the initial states are unknown, the states typically converge to the correct values if enough number of particles is used. A PF approximates the posterior with a set of state samples, called particles, instead of assuming that the posteriors are Gaussian at every time step. As the number of particles increases, the approximated posteriors get closer to the true posterior, which offers more accurate estimation at the cost of higher computational complexity [88], [89]. To reduce the computational complexity, the linear Gaussian part of the system can be solved by using a KF while the remaining part is solved by using a PF [90]. This combination provides improved results even with a smaller number of particles.

2.1.1 Bayes Filter

For a system identification purpose, the following dynamic state space model is considered:

$$x_k = f_k(x_{k-1}, u_{k-1}, b_{k-1}) \tag{2-1}$$

$$z_k = h_k(x_k, d_k)\,, \tag{2-2}$$

where subscript k represents the iteration number or time t_k, x_k is the state, f_k is a state transition function from time t_{k-1} to time t_k, u_{k-1} is the deterministic input, b_{k-1} is the process noise, z_k is the measurement, d_k is the measurement noise, and h_k is a measurement function. The complete solution of the current state (x_k) is presented through the posterior probability density function (PDF) when all the measurements up to the current time instant ($z_{1:k}$) and all the inputs up to the previous time instant ($u_{0:k-1}$) are given. Bayes filter calculates the posterior PDF, $p(x_k \mid z_{1:k}, u_{0:k-1})$, in two steps: (i) prediction and (ii) update. Using the Bayes rule and the Markov property, that is if the current state is known, the future state is independent of the past states, the prediction and update step can be formulated as

Prediction:

$$\begin{aligned} & p(x_k \mid z_{1:k-1}, u_{0:k-1}) \\ & = \int p(x_k \mid x_{k-1}, u_{k-1}) \cdot p(x_{k-1} \mid z_{1:k-1}, u_{0:k-2}) \cdot dx_{k-1} \end{aligned} \tag{2-3}$$

Update:

$$p(x_k \mid z_{1:k}, u_{0:k-1}) = \frac{p(z_k \mid x_k) \cdot p(x_k \mid z_{1:k-1}, u_{0:k-1})}{p(z_k \mid z_{1:k-1}, u_{0:k-1})}. \tag{2-4}$$

$p(z_k \mid x_k)$ is the likelihood, $p(x_k \mid z_{1:k-1}, u_{0:k-1})$ is the prior, and $p(z_k \mid z_{1:k-1}, u_{0:k-1})$ is a normalizing factor. In order to construct the posterior PDF, the prior must be available including the initial PDF, $p(x_0)$.

A Bayes filter requires integration over the state space, which is often impossible to calculate analytically. In some cases, the posterior distribution can be analytically calculated such as the linear Gaussian state space model (i.e., KF). When the analytical computation is not feasible, the posterior density is approximated by using estimators such as a PF.

2.1.2 Kalman Filter [91]

The KF presents an optimal solution of a Bayes filter by assuming that the posterior density is Gaussian. In order for the posteriors to be Gaussian at every time step, the following conditions must be satisfied [92], [93]:

- The initial PDF is Gaussian.
- $f_k(x_{k-1}, u_{k-1}, b_{k-1})$ is a linear function of x_{k-1} and u_{k-1} with added Gaussian noise.
- $h_k(x_k, d_k)$ is a linear function of x_k with added Gaussian noise.

By using these assumptions, (2-1) and (2-2) become:

$$x_k = \Phi_k \cdot x_{k-1} + \Gamma_k \cdot u_{k-1} + b_{k-1}, \tag{2-5}$$

$$z_k = H_k \cdot x_k + d_k, \tag{2-6}$$

where Φ_k is the system transition matrix from time t_{k-1} to time t_k, Γ_k is the input matrix, H_k is the measurement matrix. It is assumed that the process noise and the measurement noise have zero-mean Gaussian distributions and they are uncorrelated. In addition, the covariance of process noise (Q_k) and the covariance of measurement noise (R_k) at each time step are known as well as the initial state (x_0) and initial covariance ($\hat{P}_0$).

The probability density distributions of both the predicted state and the measurement are Gaussian. Since the estimated state ($\hat{x}_k$) is the combination of the two probability density distributions, the estimated state is also Gaussian and has a linear relation of predicted state $\tilde{x}_k$ and measurement z_k as follows:

$$\hat{x}_k = L_k \tilde{x}_k + K_k z_k. \tag{2-7}$$

The objective of the KF is to find the weights, L_k and K_k, that minimize the error covariance. Since the KF is a special case of the Bayesian filter, the KF also estimates states by using the prediction step and the update step. When the previous estimated state and input are known, the current state can be predicted as

$$\tilde{x}_k = \Phi_k \cdot \hat{x}_{k-1} + \Gamma_k \cdot u_{k-1}. \tag{2-8}$$

In order to predict the error covariance, the error must be predicted. The predicted error is expressed as

$$\begin{aligned}\tilde{e}_k = \tilde{x}_k - x_k &= \Phi_k \hat{x}_{k-1} + \Gamma_k u_{k-1} - x_k \\ &= \Phi_k (x_{k-1} + \hat{e}_{k-1}) + \Gamma_k u_{k-1} - x_k \\ &= (\Phi_k x_{k-1} + \Gamma_k u_{k-1} + b_{k-1}) - b_{k-1} + \Phi_k \hat{e}_{k-1} - x_k \\ &= \Phi_k \hat{e}_{k-1} - b_{k-1}\end{aligned}, \tag{2-9}$$

where $\hat{e}_{k-1}$ is the estimated error at time t_{k-1}. Then, the prediction covariance is calculated as

$$\begin{aligned}\tilde{P}_k &= E[\tilde{e}_k \tilde{e}_k{}^T] \\ &= E[(\Phi_k \hat{e}_{k-1} - b_{k-1})(\Phi_k \hat{e}_{k-1} - b_{k-1})^T] \\ &= \Phi_k E[\hat{e}_{k-1}\hat{e}_{k-1}{}^T]\Phi_k{}^T + E[b_{k-1}b_{k-1}{}^T] - \Phi_k E[\hat{e}_{k-1}b_{k-1}{}^T] - E[b_{k-1}\hat{e}_{k-1}{}^T]\Phi_k{}^T\end{aligned}. \tag{2-10}$$

Since the estimated error and process noise are uncorrelated, $E[\hat{e}_{k-1}b_{k-1}{}^T] = E[b_{k-1}\hat{e}_{k-1}{}^T] = 0$. Therefore, the prediction error covariance becomes

$$\tilde{P}_k = \Phi_k \hat{P}_{k-1} \Phi_k{}^T + Q_{k-1}, \tag{2-11}$$

where $\hat{P}_{k-1}$ is the estimated error covariance at time t_{k-1}, which is defined as

$$\hat{P}_{k-1} = E(\hat{e}_{k-1}\hat{e}_{k-1}{}^T). \tag{2-12}$$

When the current measurement is available, the predicted state and error covariance is updated. From (2-7), the estimated error ($\hat{e}_k$) is calculated as:

$$\begin{aligned}\hat{e}_k = \hat{x}_k - x_k &= L_k \tilde{x} + K_k z_k - x_k \\ &= L_k (x_k + \tilde{e}_k) + K_k z_k - x_k \\ &= L_k \tilde{e}_k + L_k x_k + K_k (H_k x_k + d_k) - x_k \\ &= L_k \tilde{e}_k + (L_k + K_k H_k - I)x_k + K_k d_k\end{aligned}. \tag{2-13}$$

In order to minimize the estimated error covariance, L_k should be defined as

$$L_k = I - K_k H_k. \tag{2-14}$$

Then, the estimated state becomes

$$\begin{aligned}\hat{x}_k &= L_k \tilde{x}_k + K_k z_k \\ &= (I - K_k H_k)\tilde{x}_k + K_k z_k \\ &= \tilde{x}_k + K_k (z_k - H_k \tilde{x}_k)\end{aligned}. \tag{2-15}$$

In order to estimate the estimated error covariance, the estimated error should be calculated. From (2-13) and (2-14), the estimated error becomes

$$\begin{aligned}\hat{e}_k &= L_k\tilde{e}_k + (L_k + K_k H_k - I)x_k + K_k d_k \\ &= (I - K_k H_k)\tilde{e}_k + K_k d_k\end{aligned} . \qquad (2\text{-}16)$$

Therefore, the estimated error covariance at time t_k becomes

$$\begin{aligned}\hat{P}_k &= E(\hat{e}_k \hat{e}_k{}^T) \\ &= E\left[\{(1 - K_k H_k)\tilde{e}_k + K_k d_k\}\{(1 - K_k H_k)\tilde{e}_k + K_k d_k\}^T\right] \\ &= E\left[(1 - K_k H_k)\tilde{e}_k\tilde{e}_k{}^T(1 - K_k H_k)^T + K_k d_k d_k{}^T K_k{}^T\right] \\ &+ E\left[(1 - K_k H_k)\tilde{e}_k d_k{}^T K_k{}^T + K_k d_k \tilde{e}_k{}^T(1 - K_k H_k)^T\right]\end{aligned} . \qquad (2\text{-}17)$$

Since the estimated error and measurement noise are uncorrelated, both $E[\tilde{e}_k d_k{}^T]$ and $E[d_k \tilde{e}_k{}^T]$ are zero. Then, (2-17) can be simplified as

$$\hat{P}_k = (I - K_k H_k)\tilde{P}_k (I - K_k H_k)^T + K_k R_k K_k{}^T . \qquad (2\text{-}18)$$

Now, the weight factor, K_k, should be calculated. This weight factor should be chosen so that it minimizes the estimated error covariance. Eq. (2-18) can be expanded as:

$$\begin{aligned}\hat{P}_k &= [(I - K_k H_k)\tilde{P}_k (I - K_k H_k)^T + K_k R_k K_k{}^{-1} \\ &= \tilde{P}_k + K_k (H_k \tilde{P}_k H_k^T + R_k) K_k^T - K_k H_k \tilde{P}_k - (K_k H_k \tilde{P}_k)^T \\ &= \tilde{P}_k + K_k (H_k \tilde{P}_k H_k^T) K_k^T + K_k R_k K_k^T - K_k H_k \tilde{P}_k - (K_k H_k \tilde{P}_k)^T\end{aligned} . \qquad (2\text{-}19)$$

Since $\hat{P}_k$ is a covariance matrix, it should be symmetric and non-negative. Thus, K_k should be chosen so that the trace of the estimated error covariance is minimized. The trace of $\hat{P}_k$ is calculated as

$$Trace[\hat{P}_k] = Trace[\tilde{P}_k] + Trace[K_k (H_k \tilde{P}_k H_k^T) K_k^T] + Trace[K_k R_k K_k^T] - 2Trace[K_k H_k \tilde{P}_k] . \qquad (2\text{-}20)$$

To find the weight factor K_k that minimizes $Trace[\hat{P}]$, the derivative of $Trace[\hat{P}]$ with respect to K_k should be zero as

$$\frac{\partial Trace[\hat{P}_k]}{\partial K_k}=\frac{\partial Trace[\tilde{P}_k]}{\partial K_k}+\frac{\partial Trace[K_k(H_k\tilde{P}_kH_k^T)K_k^T]}{\partial K_k}+\frac{\partial Trace[K_kR_kK_k^T]}{\partial K_k}-\frac{2\partial Trace[K_kB_k]}{\partial K_k}=0\,.$$

(2-21)

The first term in (2-21) is

$$\frac{\partial Trace[\tilde{P}_k]}{\partial K_k}=\frac{\partial Trace[\Phi_{k-1}\hat{P}_{k-1}\Phi_{k-1}+Q_{k-1}]}{\partial K_k}=0\,.$$

Since the error covariance matrix and the measurement noise covariance matrix are symmetric, the second term and third term in (2-21) is

$$\frac{\partial Trace{\scriptstyle r}[K_k(H_k\tilde{P}_kH_k^T)K_k{}^T]}{\partial K_k}=2K_k(H_k\tilde{P}_kH_k^T)$$

$$\frac{\partial Trace[K_kR_kK_k{}^T]}{\partial K_k}=2K_kR_k\,.$$

The last term in (2-21) is

$$\frac{2\partial Trace[K_kH_k\tilde{P}_k]}{\partial K_k}=2(H_k\tilde{P}_k)^T\,.$$

Then, (2-21) can be simplified as follows:

$$2K_k(H_k\tilde{P}_kH_k^T+R_k)-2(H_k\tilde{P}_k)^T=0\,. \qquad (2\text{-}22)$$

Then, the weight factor becomes:

$$K_k=\tilde{P}_k^TH_k^T(H_k\tilde{P}_kH_k^T+R_k)^{-1}\,. \qquad (2\text{-}23)$$

Since the predicted error covariance matrix is symmetric, the weight factor can be written as

$$K_k=\tilde{P}_kH_k{}^T(H_k\tilde{P}_kH_k{}^T+R_k)^{-1}\,. \qquad (2\text{-}24)$$

Then, (2-18) can be further simplified as follows:

$$
\begin{aligned}
\hat{P}_k &= [(I - K_k H_k)\tilde{P}_k (I - K_k H_k)^T + K_k R_k K_k^T] \\
&= \tilde{P}_k + K_k (H_k \tilde{P}_k H_k^T + R_k) K_k^T - K_k H_k \tilde{P}_k - (K_k H_k \tilde{P}_k)^T \\
&= \tilde{P}_k + [\tilde{P}_k H^T (H_k \tilde{P}_k H_k^T + R_k)^{-1} (H_k \tilde{P}_k H_k^T + R_k) K_k^T] - K_k H_k \tilde{P}_k - (K_k H_k \tilde{P}_k)^T \\
&= \tilde{P}_k + \tilde{P}_k H_k^T K_k^T - K_k H_k \tilde{P}_k - (K_k H_k \tilde{P}_k)^T \\
&= \tilde{P}_k - K_k H_k \tilde{P}_k \\
&= \left[I - K_k \cdot H_k\right] \cdot \tilde{P}_k
\end{aligned}
\tag{2-25}
$$

In summary, the prediction and update equations of the KF are as follows.

Prediction:

Predicted state: $$\tilde{x}_k = \Phi_k \cdot \hat{x}_{k-1} + \Gamma_k \cdot u_{k-1} \tag{2-8}$$

Predicted covariance: $$\tilde{P}_k = \Phi_k \cdot \hat{P}_{k-1} \cdot \Phi_k{}^T + Q_{k-1} \tag{2-11}$$

Update:

Kalman gain: $$K_k = \tilde{P}_k \cdot H_k{}^T \cdot \left[H_k \cdot \tilde{P}_k \cdot H_k{}^T + R_k\right]^{-1} \tag{2-24}$$

Estimated covariance: $$\hat{P}_k = \left[I - K_k \cdot H_k\right] \cdot \tilde{P}_k \tag{2-25}$$

Estimated state: $$\hat{x}_k = \tilde{x}_k + K_k \cdot (z_k - H_k \cdot \tilde{x}_k) \tag{2-15}$$

2.1.3 Sampling Importance Resampling Particle Filter [93]

The PF is a suboptimal solution that approximates the true posterior with a finite number of random state samples with the corresponding normalized weights. Then, the posterior density approximation at time t_k is

$$p(x_k \mid z_{1:k}, u_{0:k-1}) \approx \sum_{t=1}^{N} w_k^i \delta(x_k - x_k^i), \tag{2-26}$$

where $\delta(\cdot)$ is the Dirac delta function, *N* is the number of samples, w_k^i is the normalized weight of the i^{th} particle at time t_k, and x_k^i is the i^{th} state particle at time t_k.

In order to derive the PF state estimation algorithm, consider the posterior density up to time t_k such that

$$p(x_{0:k} \mid z_{1:k}, u_{0:k-1}) = \frac{p(z_k \mid x_k) \cdot p(x_k \mid x_{k-1}, u_{k-1}) \cdot p(x_{0:k-1} \mid z_{1:k-1}, u_{0:k-2})}{p(z_k \mid z_{1:k-1}, u_{0:k-1})}. \tag{2-27}$$

Since it is usually difficult to sample from the posterior density, the importance sampling technique [94] is used to sample in the PF. When the target density (posterior density in this case) can be evaluated at any point but is difficult to sample from, samples can be drawn from a known normalized probability density [$r(x)$], the so-called importance density. To compensate for the difference between the target density and the importance density, normalized weights, which are the ratios of the two densities, are assigned to all the particles [92]. The discrete posterior density approximation up to time t_k is expressed as

$$p(x_{0:k} \mid z_{1:k}, u_{0:k-1}) \approx \sum_{t=1}^{N} w_k^i \delta(x_{0:k} - x_{0:k}^i). \tag{2-28}$$

The normalized weight has the following relationship with the target density, $p(x)$, and the importance density, $r(x)$:

$$\begin{aligned} w_k^i &\propto \frac{p(x_{0:k}^i \mid z_{1:k}, u_{0:k-1})}{r(x_{0:k}^i \mid z_{1:k}, u_{0:k-1})} \\ &\propto \frac{p(z_k \mid x_k^i) \cdot p(x_k^i \mid x_{k-1}^i, u_{k-1}) \cdot p(x_{0:k-1}^i \mid z_{1:k-1}, u_{0:k-2})}{r(x_{0:k}^i \mid z_{1:k}, u_{0:k-1})}. \end{aligned} \tag{2-29}$$

The importance density should be chosen so that it can be determined recursively as

$$r(x_{0:k} \mid z_{1:k}, u_{0:k-1}) = r(x_k \mid x_{0:k-1}, z_{1:k}, u_{0:k-1}) \cdot r(x_{0:k-1} \mid z_{1:k-1}, u_{0:k-2}). \tag{2-30}$$

When the importance density also satisfies the Markov property same as the target density, (2-29) is rewritten as

$$\begin{aligned} w_k^i &\propto \frac{p(z_k \mid x_k^i) \cdot p(x_k^i \mid x_{k-1}^i, u_{k-1}) \cdot p(x_{0:k-1}^i \mid z_{1:k-1}, u_{0:k-2})}{r(x_k^i \mid x_{k-1}^i, z_k, u_{k-1}) \cdot r(x_{0:k-1}^i \mid z_{1:k-1}, u_{0:k-2})} \\ &\propto \frac{p(z_k \mid x_k^i) \cdot p(x_k^i \mid x_{k-1}^i, u_{k-1})}{r(x_k^i \mid x_{k-1}^i, z_k, u_{k-1})} \cdot w_{k-1}^i \end{aligned}. \tag{2-31}$$

To simplify (2-31), the importance density can be chosen from prior as

$$r(x_k \mid x_{k-1}, z_k, u_{k-1}) = p(x_k \mid x_{k-1}, u_{k-1})\,. \tag{2-32}$$

Then, (2-31) can be rewritten as

$$w_k^i \propto w_{k-1}^i \cdot p(z_k \mid x_k^i)\,. \tag{2-33}$$

The problem with this type of PF is that only one particle will have high weight (close to unity) and the remaining particles will have negligible weights (almost zero) when k is high. This phenomenon, called a degeneracy problem, is undesirable because the weighted particles do not represent the true posterior density. In order to avoid this problem, particles can be resampled based on their weights. Resampling draws more samples from the higher weights and reduces the number of samples from the lower weights. After resampling, all the particles are assigned the same weight; thus, the weights at time t_{k-1} are the same ($w_{k-1}^i = 1/N$). Then, (2-33) becomes

$$w_k^i \propto p(z_k \mid x_k^i)\,. \tag{2-34}$$

After resampling at time t_k , (2-26) can be written as

$$p(x_k \mid z_{1:k}, u_{0:k-1}) \approx \frac{1}{N} \sum_{t=1}^{N} \delta(x_k - x_k^i)\,. \tag{2-35}$$

Some approaches in the literature propose to calculate the weights based on the "fitness" value [95], [96] or the "evidence" value [97], [98] of each particle to represent the likelihood in (2-4).

2.2 Fuzzy Expert Systems

Fuzzy expert systems, a branch of artificial intelligence, are very powerful decision-making tools and are used for quality control algorithms in various industries including textile companies [99], steel companies [100], and PCB manufacturing [101]. The main components of an expert system are a knowledge base, reasoning mechanism, and user interface [102]. The knowledge base in a classical expert system is constructed with facts and rules that are expressed in Boolean logic. Rules are often expressed in a form of 'IF A, THEN C' where A is a set of antecedent conditions, and C is a set of consequences. In classical crisp logic, the consequence, C, is true when the antecedent, A, is perfectly satisfied. As discussed in section 1.2, position sensors do not always output reliable measurement. As a result, fuzzy expert systems are proposed in various position tracking applications including a GPS/IMU hybrid system [106]-[108] and an Ultrasonic/IMU hybrid sensor [109] due to the ability to estimate outputs when sensor measurements have uncertainties.

2.2.1 Fuzzy Set

A fuzzy expert system utilizes fuzzy sets [103] and fuzzy logic [104]-[105] and allows for the computation of a partially true consequence based on how much the antecedent is satisfied. A fuzzy set, *F*, is defined on a universe discourse and can be described in terms of the membership function. The fuzzy membership function can take any value between 0 and 1, where 0 indicates false (complete non-membership) and 1 indicates true (complete membership). The value between 0 and 1 represents a partial membership to the fuzzy set. The fuzzy set is a generalization of crisp sets where the governing axioms are relaxed to allow for partial membership. A fuzzy set of generic elements, x , and its membership function, $\mu_F(x)$, can be represented as follows:

$$F = \begin{cases} \int_U \mu_F(x) & x \text{ if U is continuous} \\ \sum_U \mu_F(x) & x \text{ if U is discrete} \end{cases}$$

DeMorgan’s law, laws of contradiction and the excluded middle are also investigated for these operations. Motivated by their crisp counterparts, for any two fuzzy sets, *A* and *B*, the fuzzy union, intersection and complement is defined as:

$$\begin{aligned} \mu_{A\cup B}(x) &= \max[\mu_A(x), \mu_B(x)] \\ \mu_{A\cap B}(x) &= \min[\mu_A(x), \mu_B(x)] \\ \mu_{\overline{A}}(x) &= 1 - \mu_A(x) \end{aligned} \tag{2-36}$$

The fuzzy set theory can be viewed as an extension of the classic crisp set theory where DeMorgan’s Law holds. However, the laws of contradiction and excluded middle in a crisp set may not be true for fuzzy operations due to a partial membership. For example, the following can be true in a fuzzy set:

$$\begin{aligned} A \cup \overline{A} &\neq U \\ A \cap \overline{A} &\neq \varnothing \end{aligned} \tag{2-37}$$

2.2.2 Fuzzy Logic

By applying the fuzzy set to the classical crisp logic, fuzzy logic is derived. The fundamental axioms of the fuzzy logic are: (i) each fuzzy proposition has a membership degree between 0 and 1, (ii) each fuzzy proposition is a collection of linguistic terms and fuzzy operations, (iii) the terms of fuzzy proposition are defined within the fuzzy set domain, and (iv) the fuzzy logic operators combining fuzzy propositions are conjunction, disjunction, negation and implication. In fuzzy logic, a proposition is a combination of terms that are defined within the boundaries of the fuzzy set theory. A fuzzy rule is also expressed in the form of 'IF A THEN C' like the classical crisp logic, but both A and C have their own membership function. The membership function measures the degree of truth of the implication.

The fuzzy logic system provides a method of mapping an input space to an output space. The mapping is achieved by transforming the crisp inputs to the membership values of a fuzzy set. This process is called fuzzification, and an example is shown in Figure 2-1. When the input value is X1 or X3, the membership degree becomes 0.5, and when the input value is X2, the membership value is 1. The membership degrees of the fuzzy set are processed by using the fuzzy inference mechanism and fuzzy rules incorporated in the rule-base and yield a fuzzy output. This fuzzy output is transformed back into crisp outputs. This inverse process of fuzzification is called defuzzification. Many defuzzifiers have been proposed and the following five difuzzifiers are widely used: maximum defuzzifier, mean of maxima defuzzifier, centroid defuzzifier, height defuzzifier, and modified height defuzzifier.

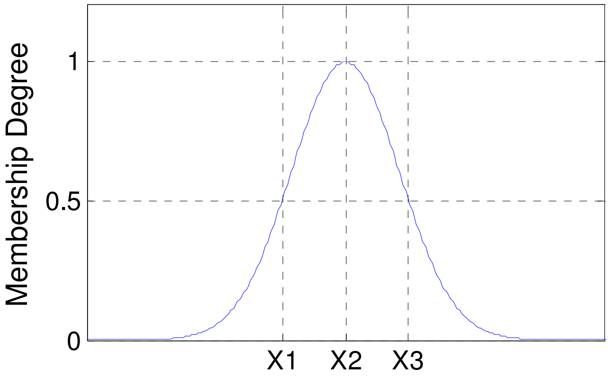


Figure 2-1: Fuzzification example.

In a fuzzy expert system, fuzzy rule-base is a collection of rules based on expert knowledge to identify the state (or output) of the given system by using the inputs. A fuzzy rule with multiple antecedents and a single conclusion in Mamdani's style is described as follows:

IF x_l is $A_{l,1}$ ✖ … x_2 is $A_{2,j}$ …✖ x_d is $A_{j,d}$ THEN y is C_l

where x_1,…, x_d are the input space, and y the output, subscript *j* is the rule number, *j=1,…,d* is the index for the inputs, and $A_{j,d}$ and $C_{1,}$ are fuzzy sets, and ✖ denotes a fuzzy operation such as 'AND', 'OR'. In Mamdani's rule, the consequence of the rule itself is a fuzzy set. Although each set of antecedent can have multiple inputs, each set has a single membership degree value. The membership degree derived from each antecedent is used to reshape the rule's output. An example of an inference mechanism is given in Figure 2-2. For example, The rules are given as follows:

Rule 1: IF x_1 is $A_{l,1}$ AND x_2 is $A_{l,2}$ THEN y1 is max[u_1(X1), u_2(X2)] and
Rule 2: IF x_1 is $A_{2,1}$ AND x_2 is $A_{2,2}$ THEN y2 is min[u_1(X1), u_2(X2)].

The membership degree based on Input 1 in Rule 1 is 0.3 and the membership degree based on Input 2 in Rule 1 is 0.5. Since Rule 1 states to find the maximum of the two inputs, the membership degree of the output of Rule 1 (y1) is 0.5, and the membership function of y1 becomes the shaded shape of the output of Rule 1. For Rule 2, the membership based on the output (y2) is 0.4, and the membership function is shown as the shaded shape of the output in Rule 2. The results of each rule are added to get the crisp output value through defuzzification. When the centroid defuzzifier is used, output y1 and output y2 are added and the centre of gravity value is chosen as the output.

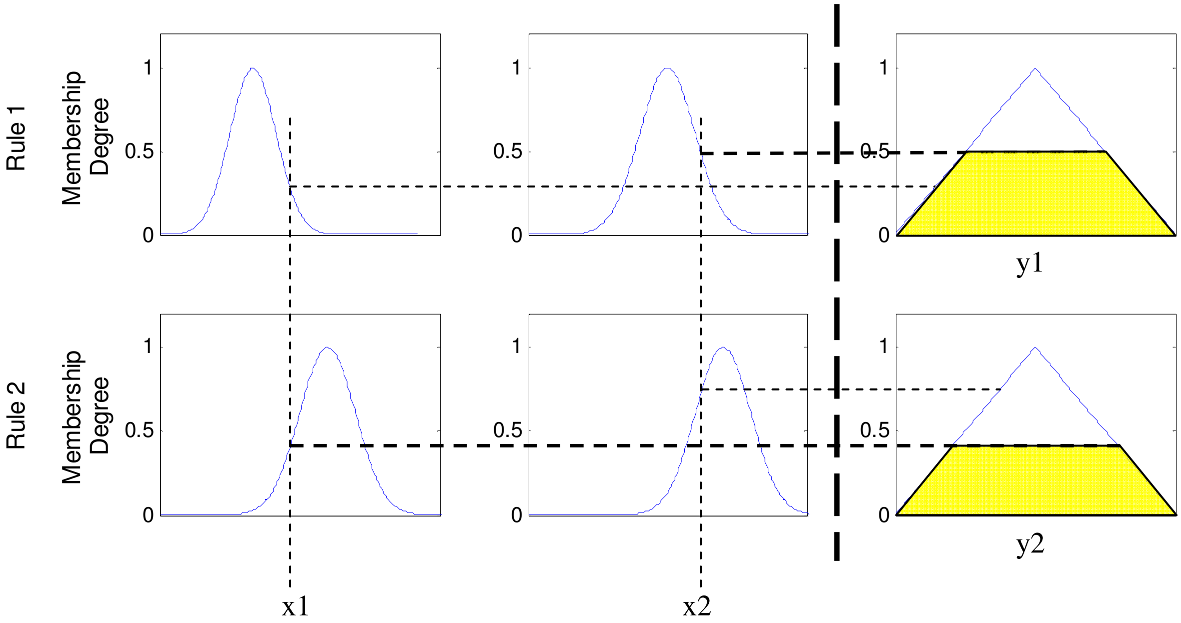


Figure 2-2: Inference mechanism.

Chapter 3
Accelerometer Calibration Technique

This chapter presents a novel triaxial accelerometer calibration method that does not require any additional sensor but guarantees a high accuracy. The proposed triaxial accelerometer calibration method utilizes the gravity vector and the mathematical model of the triaxial accelerometer calibration parameters; three gains and three biases. This method only requires the triaxial accelerometer to be stationary in six different tilt angles to estimate six calibration parameters and does not require any knowledge of the tilt angles. Since this calibration method does not require any additional sensor but offers a high accuracy, it is suitable for a variety of applications including industrial applications.

3.1 Overview

Figure 3-1 summarizes the calibration parameter estimation steps and the process of conversion from the triaxial accelerometer outputs to accelerations. If the calibration parameters are available, the accelerometer data are converted from voltage to acceleration, but if the calibration parameters are not available, the proposed method outputs the accelerometer measurements. The fundamental basis of the proposed triaxial accelerometer calibration method is that the vector sum of the acceleration measurement by using a triaxial accelerometer is equal to the gravity vector when the sensor is stationary. Since a triaxial accelerometer has six unknown calibration parameters, three gains and three biases, it has to be placed in six different tilt angles to obtain six equations as shown in Figure 3-2. The stationary state is identified by using an expert system. When the six tilt angles are measured for calibration, the sensor should be rotated in at least two different axes. If the sensor is rotated in only one axis, one of the three accelerometers will not be calibrated. When the accelerometer output readings in six different tilt angles are collected, the six calibration parameters can be estimated from the derived equations.

When the sensor is stationary, the relationship between the local gravity vector (g_l) and the accelerations in X, Y, and Z axis (A_X, A_Y, and A_Z, respectively) is

$$A_X{}^2 + A_Y{}^2 + A_Z{}^2 = (g_l)^2 \,. \tag{3-1}$$

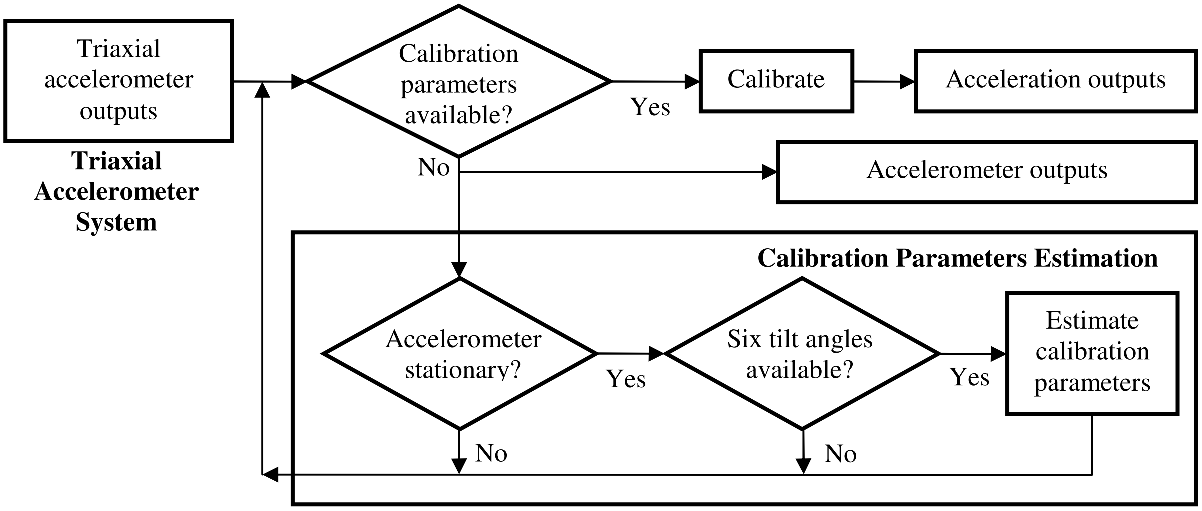

Triaxial accelerometer outputs
Triaxial Accelerometer System
Calibration parameters available?
Yes
Calibrate
Acceleration outputs
No
Accelerometer outputs
Calibration Parameters Estimation
Accelerometer stationary?
Yes
Six tilt angles available?
Yes
Estimate calibration parameters
No
No

Figure 3-1: Flow chart diagram of the proposed triaxial accelerometer calibration method.

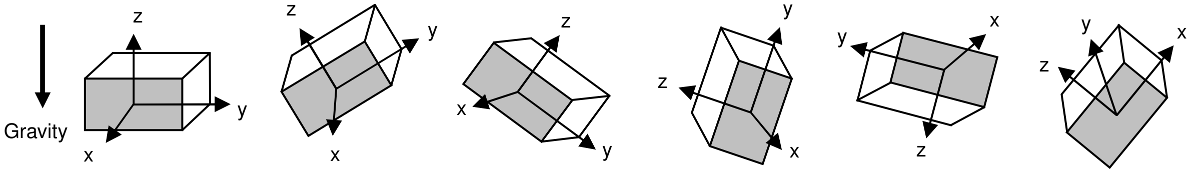


Figure 3-2: Calibration procedure: the proposed calibration method requires six different tilt angle measurements to determine six calibration parameters.

The relationship between the accelerometer outputs in each axis (S_{axis}) and the true acceleration in each axis (A_{axis}) can be written as

$$S_{axis} = G_{axis} \cdot A_{axis} + B_{axis} , \tag{3-2}$$

where G_{axis} is the true gain of each axis and B_{axis} is the true bias of each axis. Therefore, the squares of the triaxial accelerometer system outputs are described as

$$S_x{}^2 + S_y{}^2 + S_z{}^2 = (G_x \cdot A_x + B_x)^2 + (G_y \cdot A_y + B_y)^2 + (G_z \cdot A_z + B_z)^2 . \tag{3-3}$$

When the accelerometer is stationary, (3-1) holds, and (3-3) can be expanded as

$$\begin{aligned} S_x{}^2 + S_y{}^2 + S_z{}^2 = (G_x{}^2 - 1) \cdot A_x{}^2 + 2 \cdot G_x \cdot A_x \cdot B_x + B_x{}^2 \\ + (G_y{}^2 - 1) \cdot A_y{}^2 + 2 \cdot G_y \cdot A_y \cdot B_y + B_y{}^2 \\ + (G_z{}^2 - 1) \cdot A_z{}^2 + 2 \cdot G_z \cdot A_z \cdot B_z + B_z{}^2 + (g_l)^2. \end{aligned} \tag{3-4}$$

When all three true gains are unity and all three true biases are zero, the accelerometer system output of each axis is equal to the acceleration. Otherwise, there should be an error term (*Error*) due to the gain error and the bias error where the gain error is defined as the gain minus unity, and the bias error is the bias minus zero. Thus, (3-4) is represented as

$$\begin{aligned} S_x{}^2 + S_y{}^2 + S_z{}^2 = g_l{}^2 + Error \\ Error = S_x{}^2 + S_y{}^2 + S_z{}^2 - g_l{}^2. \end{aligned} \tag{3-5}$$

By substituting (3-5) into (3-4), the error term can be described as

$$\begin{aligned} Error = (G_x{}^2 - 1) \cdot A_x{}^2 + 2 \cdot G_x \cdot A_x \cdot B_x + B_x{}^2 \\ + (G_y{}^2 - 1) \cdot A_y{}^2 + 2 \cdot G_y \cdot A_y \cdot B_y + B_y{}^2 \\ + (G_z{}^2 - 1) \cdot A_z{}^2 + 2 \cdot G_z \cdot A_z \cdot B_z + B_z{}^2. \end{aligned} \tag{3-6}$$

The error term of (3-6) can be determined from (3-5). However, since the acceleration terms are unknown, the acceleration terms of (3-6) should be replaced with the known accelerometer system output terms. Then, (3-6) is rewritten as

$$\begin{aligned} Error = (1 - 1/G_x{}^2) \cdot S_x{}^2 + 2 \cdot S_x \cdot B_x / G_x{}^2 + (1 - 1/G_y{}^2) \cdot S_y{}^2 + 2 \cdot S_y \cdot B_y / G_y{}^2 \\ + (1 - 1/G_z{}^2) \cdot S_z{}^2 + 2 \cdot S_z \cdot B_z / G_z{}^2 - (B_x{}^2 / G_x{}^2 + B_y{}^2 / G_y{}^2 + B_z{}^2 / G_z{}^2). \end{aligned} \tag{3-7}$$

Eq. (3-7) has six unknowns, and the last term, $(B_x{}^2 / G_x{}^2 + B_y{}^2 / G_y{}^2 + B_z{}^2 / G_z{}^2)$, is non-linear and has six unknowns. Thus, to determine all six unknowns, this problem should be solved by using an iterative method.

3.2 Iteration Method to Calculate Axes Gains and Biases

The proposed calibration method uses an iterative method to calculate the gains and biases of each axis of a triaxial accelerometer. To implement an iterative method, (3-2) should be rewritten so that it fits the method. Thus, (3-2) is rewritten as

$$\begin{aligned} S_{axis} &= G_{axis} \cdot A_{axis} + B_{axis} = \left(\prod_{i=0}^{k} \tilde{G}_{axis,i} \right) \cdot \hat{A}_{axis,k} + \sum_{i=0}^{k} \tilde{B}_{axis,i} \\ &= \hat{G}_{axis,k} \cdot \hat{A}_{axis,k} + \hat{B}_{axis,k}, \end{aligned} \quad , \tag{3-8}$$

where $\tilde{G}_{axis,i}$ represents the calculated gain of each axis at the i^{th} iteration, $\tilde{B}_{axis,i}$ represents the calculated bias of each axis at the i^{th} iteration, $\hat{A}_{axis,k}$ is the estimated acceleration of each axis at the k^{th} iteration, $\hat{G}_{axis,k}$ is the estimated gain of each axis at the k^{th} iteration, and $\hat{B}_{axis,k}$ is the estimated bias of each axis at the k^{th} iteration. Given that the true gains are real positive numbers and the true biases are real numbers, the initial estimated gain of each axis ($\hat{G}_{axis,0}$) can be chosen from any positive real number, and the initial estimated bias of each axis ($\hat{B}_{axis,0}$) can be chosen from any real number.

When the iterative method converges, the estimated gains and biases at the k^{th} iteration should match their true counterparts. Thus, the objective of the iterative method is to determine the calculated gains of each axis ($\tilde{G}_{axis,k}$) and the calculated biases of each axis ($\tilde{B}_{axis,k}$) that satisfy

$$\begin{aligned} G_{axis} &= \hat{G}_{axis,k} = \hat{G}_{axis,k-1} \cdot \tilde{G}_{axis,k} \\ B_{axis} &= \hat{B}_{axis,k} = \hat{B}_{axis,k-1} + \tilde{B}_{axis,k} \end{aligned} . \tag{3-9}$$

From (3-8), the estimated acceleration at the (k-1)th iteration is

$$\hat{A}_{axis,k-1} = (S_{axis} - \hat{B}_{axis,k-1}) / \hat{G}_{axis,k-1} . \tag{3-10}$$

Since the accelerometer outputs are known as well as the previous estimated gains and biases, the acceleration of each axis at the (k-1)th iteration can be calculated by using (3-10). When the accelerometer is stationary, (3-1) holds. If $\hat{A}_{axis,k-1}$ does not match the true acceleration of each axis (A_{axis}), an error is encountered. The error term at the (k-1)th iteration is

$$\begin{aligned} E_{k-1} &= (\hat{A}_{X,k-1})^2 + (\hat{A}_{Y,k-1})^2 + (\hat{A}_{Z,k-1})^2 - g_l{}^2 \\ &= ((S_X - \hat{B}_{X,k-1}) / \hat{G}_{X,k-1})^2 + ((S_Y - \hat{B}_{Y,k-1}) / \hat{G}_{Y,k-1})^2 + ((S_Z - \hat{B}_{Z,k-1}) / \hat{G}_{Z,k-1})^2 - g_l{}^2. \end{aligned} \tag{3-11}$$

Since all the terms in (3-11) are known, E_{k-1} can be calculated. By using (3-1), (3-11) is written as

$$\begin{aligned} E_{k-1} = {} & ((S_X - \hat{B}_{X,k-1}) / \hat{G}_{X,k-1})^2 + ((S_Y - \hat{B}_{Y,k-1}) / \hat{G}_{Y,k-1})^2 \\ & + ((S_Z - \hat{B}_{Z,k-1}) / \hat{G}_{Z,k-1})^2 - (A_X{}^2 + A_Y{}^2 + A_Z{}^2). \end{aligned} \tag{3-12}$$

The true acceleration terms in (3-12) are replaced with the known accelerometer terms from (3-8) and (3-9). Then, (3-12) can be expanded as

$$\begin{aligned} E_{k-1} = {} & (1 - 1/\tilde{G}_{X,k}{}^2) \cdot \hat{A}_{X,k-1}{}^2 + 2 \cdot \tilde{B}_{X,k} \cdot \hat{A}_{X,k-1} / (\hat{G}_{X,k-1} \cdot \tilde{G}_{X,k}{}^2) \\ & + (1 - 1/\tilde{G}_{Y,k}{}^2) \cdot \hat{A}_{Y,k-1}{}^2 + 2 \cdot \tilde{B}_{Y,k} \cdot \hat{A}_{Y,k-1} / (\hat{G}_{Y,k-1} \cdot \tilde{G}_{Y,k}{}^2) \qquad , \\ & + (1 - 1/\tilde{G}_{Z,k}{}^2) \cdot \hat{A}_{Z,k-1}{}^2 + 2 \cdot \tilde{B}_{Z,k} \cdot \hat{A}_{Z,k-1} / (\hat{G}_{Z,k-1} \cdot \tilde{G}_{Z,k}{}^2) - \varepsilon_{k-1} \end{aligned} \tag{3-13}$$

where $\varepsilon_{k-1} = (\dfrac{\tilde{B}_{X,k}{}^2}{(\hat{G}_{X,k-1} \cdot \tilde{G}_{X,k})^2} + \dfrac{\tilde{B}_{Y,k}{}^2}{(\hat{G}_{Y,k-1} \cdot \tilde{G}_{Y,k})^2} + \dfrac{\tilde{B}_{Z,k}{}^2}{(\hat{G}_{Z,k-1} \cdot \tilde{G}_{Z,k})^2})$. Eq. (3-13) has six unknowns and the last term, ε_{k-1}, is nonlinear with all six unknowns. However, when the iterations converge, ε_{k-1} becomes almost zero because the calculated bias terms ($\tilde{B}_{X,k}$, $\tilde{B}_{Y,k}$, and $\tilde{B}_{Z,k}$) are expected to converge to zero. By setting ε_{k-1} zero, (3-13) can be rewritten as

$$\begin{aligned} E_{k-1} = {} & (1 - 1/\tilde{G}_{X,k}{}^2) \cdot \hat{A}_{X,k-1}{}^2 + 2 \cdot \tilde{B}_{X,k} \cdot \hat{A}_{X,k-1} / (\hat{G}_{X,k-1} \cdot \tilde{G}_{X,k}{}^2) \\ & + (1 - 1/\tilde{G}_{Y,k}{}^2) \cdot \hat{A}_{Y,k-1}{}^2 + 2 \cdot \tilde{B}_{Y,k} \cdot \hat{A}_{Y,k-1} / (\hat{G}_{Y,k-1} \cdot \tilde{G}_{Y,k}{}^2) \quad . \\ & + (1 - 1/\tilde{G}_{Z,k}{}^2) \cdot \hat{A}_{Z,k-1}{}^2 + 2 \cdot \tilde{B}_{Z,k} \cdot \hat{A}_{Z,k-1} / (\hat{G}_{Z,k-1} \cdot \tilde{G}_{Z,k}{}^2) \end{aligned} \tag{3-14}$$

To solve for the six calibration parameters, a triaxial accelerometer should be placed in six different tilt angles. Then, (3-14) becomes a 6 by 1 matrix such that

$$[Error_{k-1}] = [Accel_{k-1}] \cdot [Cal_k]$$

$$\begin{bmatrix} E_{k-1}1 \\ E_{k-1}2 \\ E_{k-1}3 \\ E_{k-1}4 \\ E_{k-1}5 \\ E_{k-1}6 \end{bmatrix} = \begin{bmatrix} \hat{A}_{X,k-1}1^2 & \hat{A}_{Y,k-1}1^2 & \hat{A}_{Z,k-1}1^2 & \hat{A}_{X,k-1}1 & \hat{A}_{Y,k-1}1 & \hat{A}_{Z,k-1}1 \\ \hat{A}_{X,k-1}2^2 & \hat{A}_{Y,k-1}2^2 & \hat{A}_{Z,k-1}2^2 & \hat{A}_{X,k-1}2 & \hat{A}_{Y,k-1}2 & \hat{A}_{Z,k-1}2 \\ \hat{A}_{X,k-1}3^2 & \hat{A}_{Y,k-1}3^2 & \hat{A}_{Z,k-1}3^2 & \hat{A}_{X,k-1}3 & \hat{A}_{Y,k-1}3 & \hat{A}_{Z,k-1}3 \\ \hat{A}_{X,k-1}4^2 & \hat{A}_{Y,k-1}4^2 & \hat{A}_{Z,k-1}4^2 & \hat{A}_{X,k-1}4 & \hat{A}_{Y,k-1}4 & \hat{A}_{Z,k-1}4 \\ \hat{A}_{X,k-1}5^2 & \hat{A}_{Y,k-1}5^2 & \hat{A}_{Z,k-1}5^2 & \hat{A}_{X,k-1}5 & \hat{A}_{Y,k-1}5 & \hat{A}_{Z,k-1}5 \\ \hat{A}_{X,k-1}6^2 & \hat{A}_{Y,k-1}6^2 & \hat{A}_{Z,k-1}6^2 & \hat{A}_{X,k-1}6 & \hat{A}_{Y,k-1}6 & \hat{A}_{Z,k-1}6 \end{bmatrix} \cdot \begin{bmatrix} 1 - 1/\tilde{G}_{X,k}{}^2 \\ 1 - 1/\tilde{G}_{Y,k}{}^2 \\ 1 - 1/\tilde{G}_{Z,k}{}^2 \\ 2 \cdot \tilde{B}_{X,k} / (\hat{G}_{X,k-1} \cdot \tilde{G}_{X,k}{}^2) \\ 2 \cdot \tilde{B}_{Y,k} / (\hat{G}_{Y,k-1} \cdot \tilde{G}_{Y,k}{}^2) \\ 2 \cdot \tilde{B}_{Z,k} / (\hat{G}_{Z,k-1} \cdot \tilde{G}_{Z,k}{}^2) \end{bmatrix} . \tag{3-15}$$

Matrix $[Error_{k-1}]$ can be calculated from (3-11), and Matrix $[Accel_{k-1}]$ can be calculated from (3-10). Since Matrix $[Error_{k-1}]$ and Matrix $[Accel_{k-1}]$ are known, Matrix $[Cal_k]$ is calculated as

$$[Cal_k] = [Accel_{k-1}]^{-1} \cdot [Error_{k-1}]\,. \tag{3-16}$$

From Matrix $[Cal_k]$, three calculated gains at the k^{th} iteration should be determined first from the first three rows. Then, with the square of the calculated gains at the k^{th} iteration and the estimated gains at the $(k\text{-}1)^{\text{th}}$ iteration, three calculated biases at the k^{th} iteration can be determined from the last three rows. The gains, however, must be real positive numbers. To ensure that the calculated gains are positive real numbers, the calculated gains are calculated as

$$\begin{aligned}
\tilde{G}_{X,k} &= \left|1 - [Accel_{k-1}{}^{-1} \cdot Error_{k-1}]_1\right|^{-0.5}, \\
\tilde{G}_{Y,k} &= \left|1 - [Accel_{k-1}{}^{-1} \cdot Error_{k-1}]_2\right|^{-0.5}, \\
\tilde{G}_{Z,k} &= \left|1 - [Accel_{k-1}{}^{-1} \cdot Error_{k-1}]_3\right|^{-0.5},
\end{aligned} \tag{3-17}$$

where subscript 1, 2, and 3 represent the row number of $[Accel_{k-1}]^{-1} \cdot [Error_{k-1}]$ matrix.

After the six unknowns are determined, the estimated gains and biases at the k^{th} iteration are obtained from (3-9). This iterative method terminates when the three calculated gains converge to unity and the three calculated biases converge to zero.

3.3 Numerical Analysis

In experiments, it is difficult to validate the calibration results because the true gains and biases are unknown. In addition, accelerometers have many sources of error such as non-linearity, misalignments, and cross-axis sensitivity. In simulations, however, the true gains and biases can be defined to allow for validating the iterative calibration technique proposed in subsection 3.1.2 without the sensor errors. The simulation procedure is depicted in Figure 3-3.

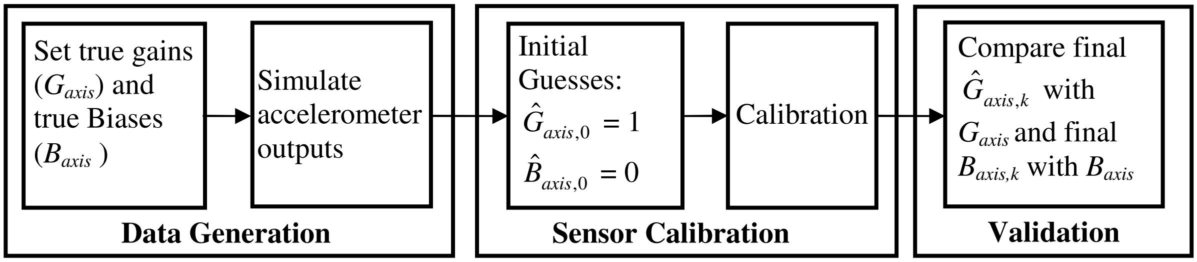

Set true gains (G_{axis}) and true Biases (B_{axis})
Simulate accelerometer outputs
Data Generation
Initial Guesses:
$\hat{G}_{axis,0} = 1$
$\hat{B}_{axis,0} = 0$
Calibration
Sensor Calibration
Compare final $\hat{G}_{axis,k}$ with G_{axis} and final $B_{axis,k}$ with B_{axis}
Validation

Figure 3-3: Procedure for the simulation calibration and parameter validation.

The true gains and biases are defined for the simulation, and the accelerometer data are generated with MATLAB® based on the true gains, true biases, gravity, and the movement of the accelerometer. Then, by using the proposed calibration method, the gains and biases are estimated. Unity gains and zero biases are chosen to initialize the iterative algorithm. When the final estimated gains and biases are calculated with the proposed calibration method, they are compared with the true gains and biases.

One hundred simulations were performed with gains between 0.001 and 1000 [V/(m/s^2)] and biases between ±100V. The six stationary tilt angles had at least 2° difference for the simulations because when the tilt angle differences are less than 1.5°, the gain and bias errors start to increase due to the computer software precision limitations. Five distinct results of the hundred simulations are presented in Table 3-1 and the tilt angle data for the five simulations are shown in Figure 3-4. For the first simulation (Simulation 1 of Table 3-1), all the biases are set to zero. The second simulation consists of high gains and high biases, and the third simulation contains low gains and low biases. Simulation 4 consists of a mixture of high and low gains and biases, and Simulation 5 has low gains and high biases. Simulation 1 converges on the first iteration because the initial estimated biases match the true biases. In this case, ε_0 in (3-13) becomes zero, and the gains and biases can be accurately calculated by using (3-14). Simulation 2 to 4 indicate that the biases converge on the first iteration, and the gains converge on the second iteration. Simulation 5 has high biases and low gains that create a high ε_0 in (3-13). Due to high ε_0, Simulation 5 requires more iteration steps than the other four simulations. For all one hundred simulations, the gains and biases converge to the correct values within three iteration steps.

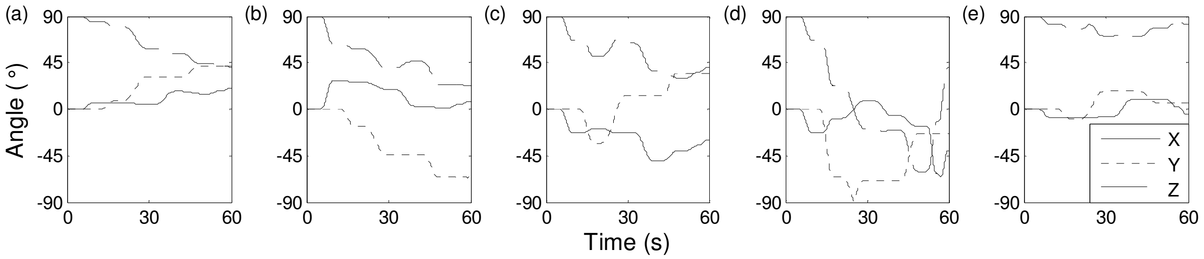


Figure 3-4: Tilt angle contents of the simulations: tilt angles for (a) Simulation 1, (b) Simulation 2, (c) Simulation 3, (d) Simulation 4, and (e) Simulation 5.

Table 3-1: Simulation results with different gains and biases.

Number of iterations	G_X [V/(m/s^2)]	G_Y [V/(m/s^2)]	G_Z [V/(m/s^2)]	B_X (V)	B_Y (V)	B_Z (V)
Simulation 1 Result						
True value	0.24	712	32	0	0	0
1 - converged	0.24000	712.00	32.000	0.00000	0.00000	0.00000
Simulation 2 Result						
True value	1000	500.0	400	100.0	-100	-25
1	999.72	499.86	399.89	100.00	-100.00	-25.000
2 - converged	1000.0	500.00	400.00	100.00	-100.00	-25.000
Simulation 3 Result						
True value	0.001	0.5	0.4	-0.1	0.5	-1
1	0.010148	5.0741	4.0593	-0.10000	0.50000	-1.0000
2 - converged	0.0010000	0.50000	0.40000	-0.10000	0.50000	-1.0000
Simulation 4 Result						
True value	0.06	300	1.4	-11	-1.5	80
1	1.1730	5864.9	27.369	-11.000	-1.5000	80.000
2 - converged	0.060000	300.00	1.4000	-11.000	-1.5000	80.000
Simulation 5 Result						
True value	0.001	0.002	0.001	-100	100	100
1	15.191	30.489	15.409	-99.155	100.77	100.17
2	0.096219	0.19244	0.096219	-100.00	100.00	100.00
3 - converged	0.0010000	0.0020000	0.0010000	-100.00	100.00	100.00

3.4 Experiments

The proposed calibration method was tested with two different triaxial accelerometers. The first triaxial accelerometer consists of three identical single-axis accelerometers, and the second triaxial accelerometer is assembled with two different biaxial accelerometers that have different gains. The experimental system setup is shown in Figure 3-5. One triaxial accelerometer is connected to a data acquisition card (DAQ) which is connected to a computer. Since both the triaxial accelerometer and the data acquisition card have their own gains and biases, the proposed calibration method determines the gains and biases of the combined system (hereafter accelerometer system).

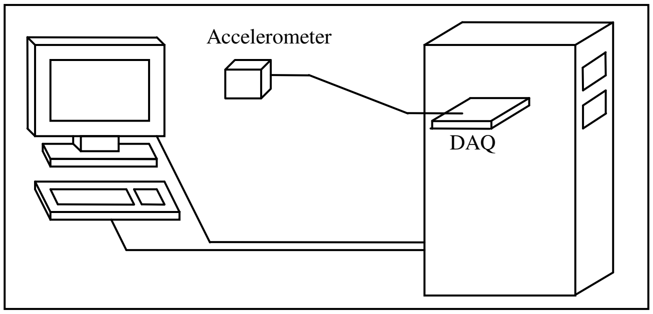


Figure 3-5: Experimental system setup.

The six sides of the accelerometer were placed on a table for calibration because such a procedure is the easiest way to make a sensor stationary in six different tilt angles and also minimizes the effects of the sensor errors such as nonlinearity on the calibration parameter calculation. In order to identify the stationary state and collect the six stationary state sensor data, an expert system is used. When the accelerometer is stationary, the acceleration in each axis should be constant. Thus, when the accelerometer measurements of each axis do not fluctuate with a magnitude that exceeds the maximum noise level of the accelerometer system for a certain period of time, it is assumed that the accelerometer is stationary. When the sensor is calibrated by a person, two seconds is a sufficient time period to identify the stationary state because it is unlikely for a person to maintain the same acceleration for two seconds unless the sensor is stationary. Instead of using instantaneous accelerometer outputs in each axis (S_{axis}), the average accelerometer outputs in each axis are used to minimize the effect of noise in the experiment.

The accurate magnitude of the local gravity vector must be known to solve (3-11). In this experiment, 9.8036 m/s^2 is used for the magnitude of the local gravity vector (Kitchener-Waterloo, Ontario, Canada) [110]. Unity gains are used for the initial estimated gains because a unit gain is far from the specs of both triaxial accelerometers. For the initial estimated biases, two values are selected: the bias voltages from the accelerometer specs, which are close to the true biases and 10V which is far from the biases of both triaxial accelerometers. Since the true gains and biases are not available in the experiments, the estimated gains and biases cannot be compared with their true counterparts. However, if the estimated gains and biases match their true counterparts, the accelerations measured with an accelerometer at stationary state should match the true accelerations that are calculated from the tilt angle measurements. Thus, to verify if the final estimated gains and biases match their true values, the acceleration measurements after calibration are compared with the accelerations calculated from the gravity vector. The simulations were generated based on the final estimated gains and biases that are obtained from the experiments with 10V initial estimated biases for comparison. The experimental test procedure is summarized in Figure 3-6.

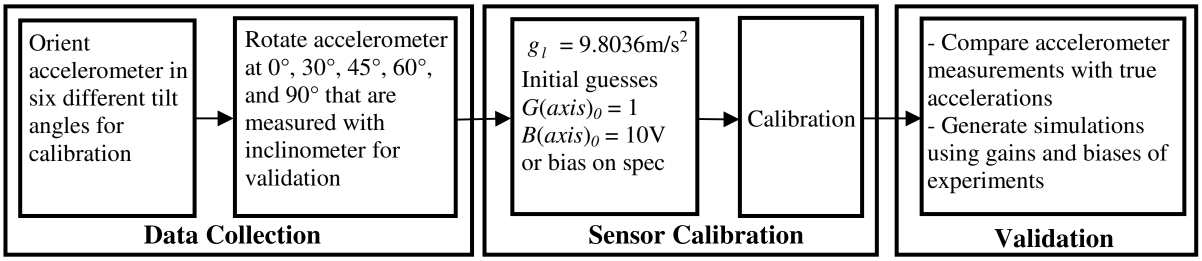

Orient accelerometer in six different tilt angles for calibration
Rotate accelerometer at 0°, 30°, 45°, 60°, and 90° that are measured with inclinometer for validation
Data Collection
g_l = 9.8036m/s^2
Initial guesses
$G(axis)_0$ = 1
$B(axis)_0$ = 10V
or bias on spec
Calibration
Sensor Calibration
- Compare accelerometer measurements with true accelerations
- Generate simulations using gains and biases of experiments
Validation

Figure 3-6: Procedure for the calibration experiment and parameter validation.

3.4.1 Experiments Using a Triaxial Accelerometer with Three Identical Single-Axis Accelerometers

A commercially available triaxial accelerometer (Colibrys, SF3000L) is selected for the first set of experiments. This accelerometer consists of three almost perfectly perpendicular single-axis accelerometers that have almost the same gains and biases. After the accelerometer was calibrated by rotating the sensor in six different tilt angles, the sensor was placed on a milling vise as shown in Figure 3-7. Then, the sensor was held stationary for a while at 0°, 30°, 45°, 60°, and 90° with respect to the gravity vector as presented in Figure 3-8 to validate the estimated gains and biases. These reference tilt angles were measured with a mechanical inclinometer (Hilger & Watts, TB121-1).

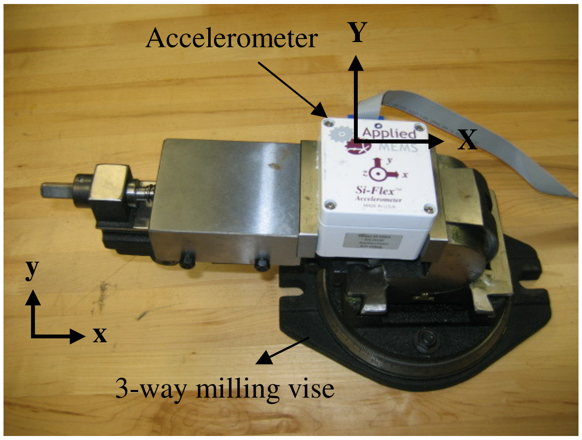
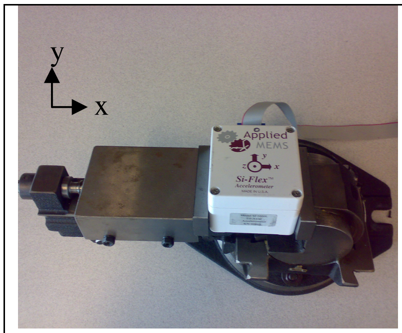
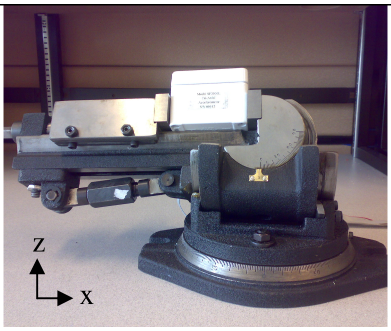
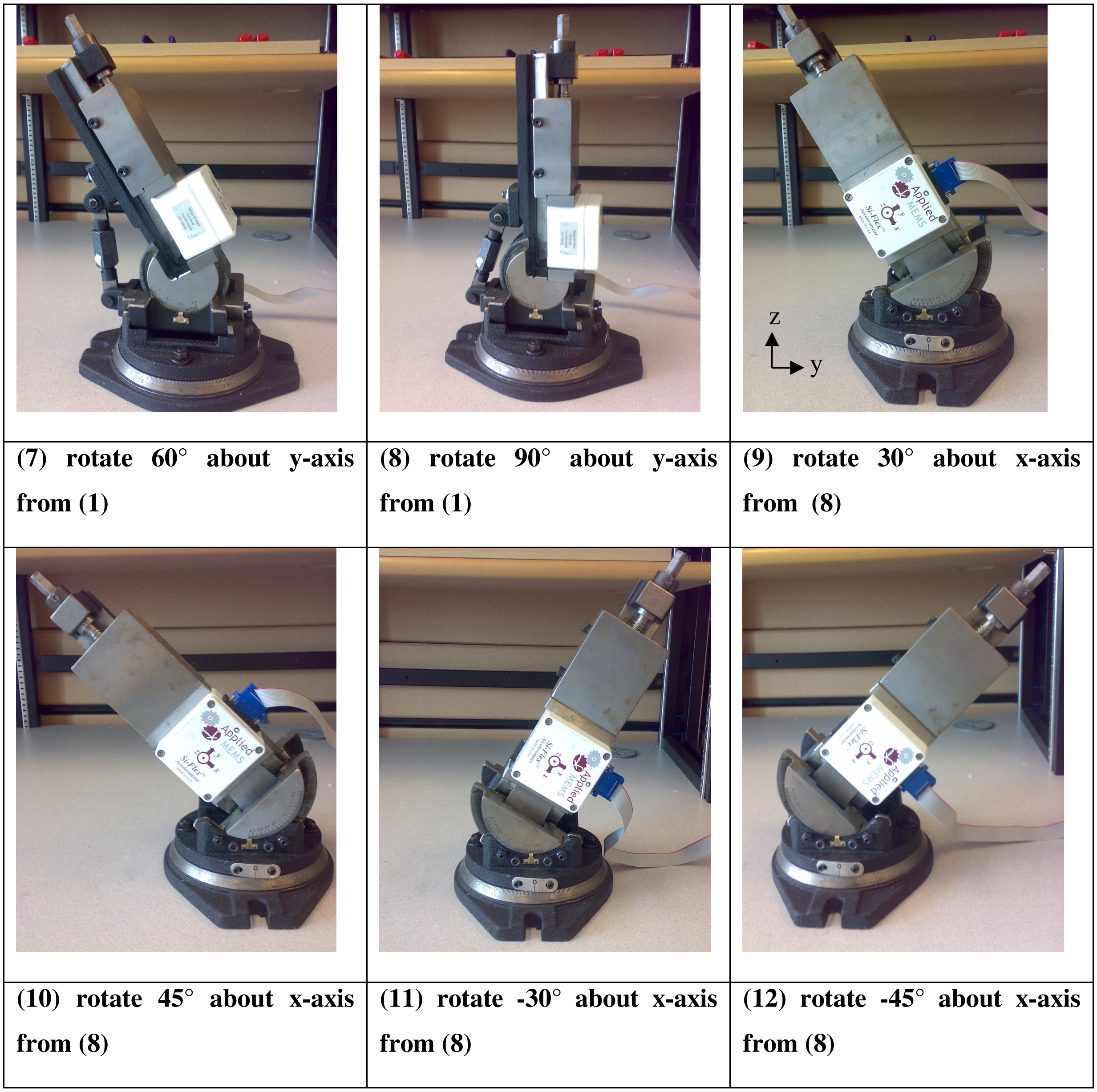


Figure 3-7: Colibrys triaxial accelerometer on a 3-way milling vise: the sensor axes (XY) and the fixed frame axes (xy).

		
(1) Initial orientation	**(2) rotate -30° about x-axis from (1)**	**(3) rotate -45° about x-axis from (1)**
(4) rotate 30° about x-axis from (1)	**(5) rotate 45° about x-axis from (1)**	**(6) rotate 30° about y-axis from (1)**



(7) rotate 60° about y-axis from (1)

(8) rotate 90° about y-axis from (1)

(9) rotate 30° about x-axis from (8)

(10) rotate 45° about x-axis from (8)

(11) rotate -30° about x-axis from (8)

(12) rotate -45° about x-axis from (8)

Figure 3-8: Rotation sequence of a triaxial accelerometer on a 3-way milling vise after calibration and the fixed frame (xyz).

Since the calibration parameters change whenever an accelerometer is powered on, three calibration tests are conducted to check if the proposed method can accurately estimate the calibration parameters in all three cases. Each test used unity initial estimated gains and two different sets of initial estimated biases: 0V and 10V. Figure 3-9 illustrates the simulation and experiment convergence steps of the calibration when the initial estimated biases are close to the true biases of the accelerometer system ($\hat{B}_{axis,0}$ = 0V). This figure indicates that the experiment convergence steps almost agree with the simulation steps. Figure 3-10 displays the simulation and experiment convergence steps of calibration when the initial estimated biases are far from the true biases ($\hat{B}_{axis,0}$ = 10V). Figure 3-10 shows noticeable mismatches between the simulation steps and the experiment steps on the first iteration while Figure 3-9 exhibits small mismatches. These mismatches could be caused by the triaxial accelerometer system error such as nonlinearity and misalignment. The sensor errors create error in Matrix $[Accel_{k-1}]$. When ε_{k-1} in (3-13) is small, the experiment steps almost match the simulation steps (Figure 3-9) because the multiplication of the inverse of Matrix $[Accel_{k-1}]$ and ε_{k-1} is small. However, when ε_{k-1} is large, the multiplication of the inverse of Matrix $[Accel_{k-1}]$ and ε_{k-1} is large. Thus, the mismatch between the simulations and experiments becomes significant as shown in Figure 3-10. Due to these sensor errors, the experiments of both Figure 3-9 and Figure 3-10 converge on the third iteration while the simulations converge on the second iteration.

The gains and biases of the simulations were generated based on the estimated gains and biases of the experiments with 10V initial estimated biases. Therefore, Figure 3-10 shows that the gains and biases of simulations (dotted lines) perfectly match the experimental results (solid lines) after convergence. Also, Figure 3-9 shows that the experimental results perfectly match the simulation results when the initial estimated biases are zeros. These results demonstrate that regardless of the initial estimated bias values, the converged calibration parameters are the same.

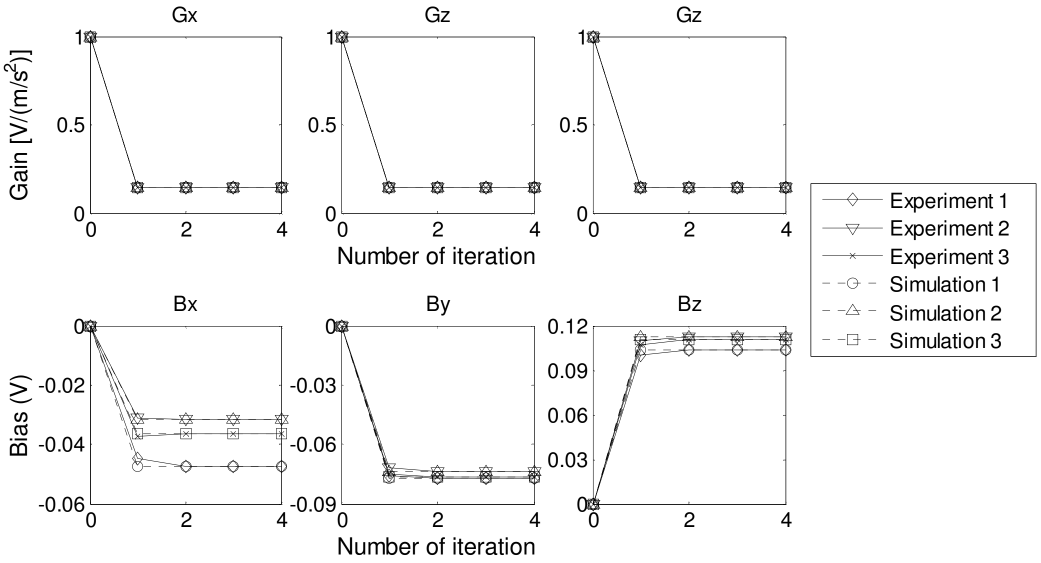


Figure 3-9: Calibration of SF3000L accelerometer system when the initial estimations of biases are 0V: experiment and simulation results.

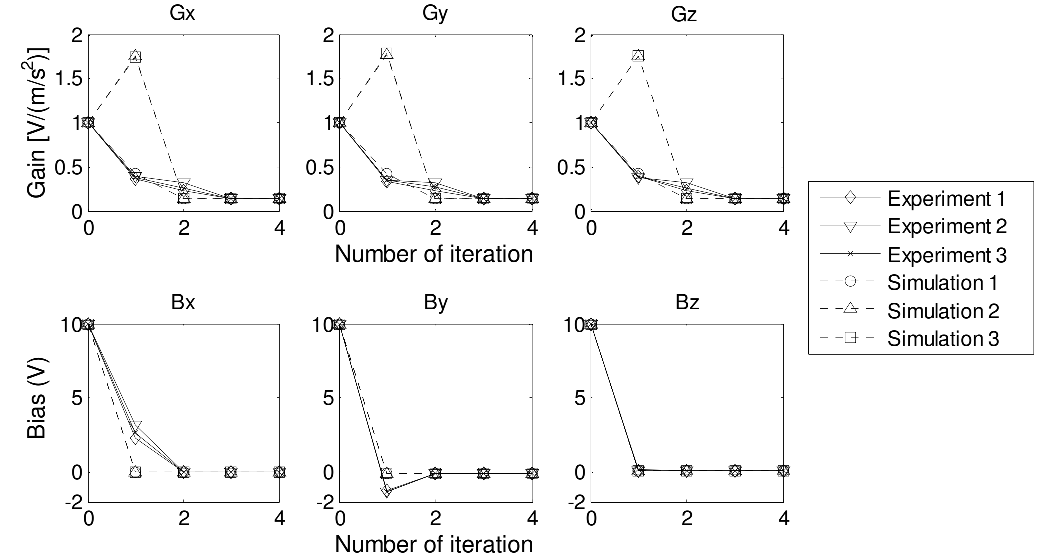


Figure 3-10: Calibration of SF3000L accelerometer system when the initial estimations of biases are 10V: experiment and simulation results.

The result of Experiment 3 in Figure 3-9 is analyzed in Figure 3-11. Figure 3-11 (a) illustrates the acceleration measurements before and after calibration, and Figure 3-11 (b) displays the tilt angle measurements by the following calculation.

$$\begin{aligned}\theta_X &= \arcsin(A_X/({A_X}^2+{A_Y}^2+{A_Z}^2)^{0.5})\times 180/\pi \\ \theta_Y &= \arcsin(A_Y/({A_X}^2+{A_Y}^2+{A_Z}^2)^{0.5})\times 180/\pi \ , \\ \theta_Z &= \arcsin(A_Z/({A_X}^2+{A_Y}^2+{A_Z}^2)^{0.5})\times 180/\pi\end{aligned} \qquad (3\text{-}18)$$

where θ_{axis} represents the tilt angle of each axis in degree. The horizontal grid lines represent the accelerations and tilt angles induced by the rotations illustrated in Figure 3-8.

For the first 35 seconds, the accelerometer is calibrated by placing it in six different tilt angles. During this period, Figure 3-11 (a) demonstrates that the magnitudes of the accelerations are not close to the magnitude of the gravity vector because the initial estimated gains and biases do not match the true values. However, after 35 seconds, the accelerometer is calibrated, and the magnitude of the acceleration almost matches the magnitude of the gravity vector. Figure 3-11 (b) shows that the tilt angle measurements with the accelerometer after calibration are stationary at 0°, 30°, 45°, 60°, and 90° with the same sequence of rotation displayed in Figure 3-8.

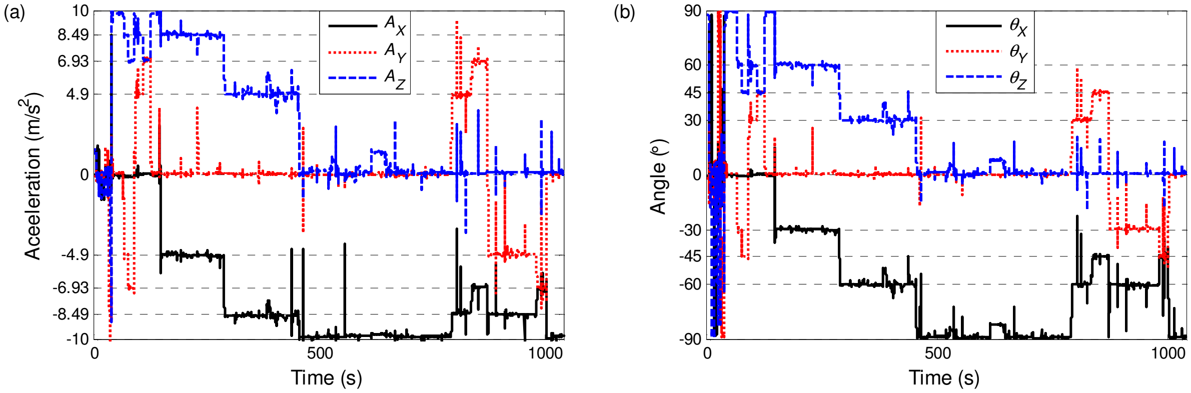


Figure 3-11: Measurements with the SF3000L triaxial accelerometer system: (a) acceleration measurements and (b) tilt angle measurements using (3-18).

A small acceleration error exists due to the errors of the accelerometer and the data acquisition card. The sources of errors are investigated and compared with the acceleration error after calibration. SF3000L has a 0.1% nonlinearity error, 0.5° misalignment error, and 0.5% cross-axis sensitivity. Also, the accelerometer is connected to a computer through a DAQ (Measurement Computing, PCI-DAS-1602/16) that contains a nonlinearity error and cross-talk. The specifications of the data acquisition card indicate that the nonlinearity error is ±5 least significant bits (LSB), and the maximum cross-talk error is ±2 LSB.

The acceleration errors of all three experiments and the error margin of the accelerometer system are analyzed in Figure 3-12. The figure denotes that the acceleration RMS error is 0.09 m/s^2 at 0° and decreases as the tilt angle increases. This occurs because the effects of the misalignment error and the cross-axis sensitivity error are the maximum at 0°. The maximum acceleration error is 0.14 m/s^2 at 0° and tends to decrease as the tilt angle increases. Figure 3-12 shows that the maximum acceleration error of each axis is below the error margin of the accelerometer system.

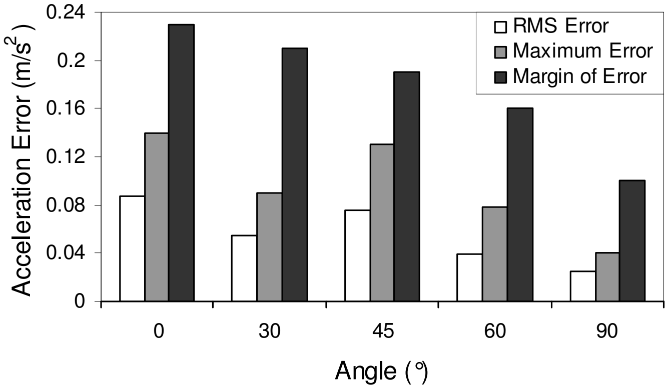


Figure 3-12: Error comparison among the acceleration RMS error, the maximum acceleration error, and the margin of error of the SF3000L triaxial accelerometer system.

The experimental results demonstrate that the proposed calibration method accurately estimates the calibration parameters when three single-axis accelerometers have almost the same gains and biases. In addition, the experimental results demonstrate that even when the initial estimated biases are different, the proposed calibration method generates accurate results.

3.4.2 Experiments Using a Triaxial Accelerometer with Different Gains

For the second set of experiments, a triaxial accelerometer is assembled with two biaxial accelerometers (MA-A202 and MA-A210, Mechworks Systems Inc.) that have different gains. MA-A202 has a gain three times higher than MA-A210. This triaxial accelerometer (hereafter referred to as a Mechworks triaxial accelerometer) is assembled with two axes (X, Y-axis) of MA-A202 and one axis (Z-axis) of MA-A210 (Figure 3-13).

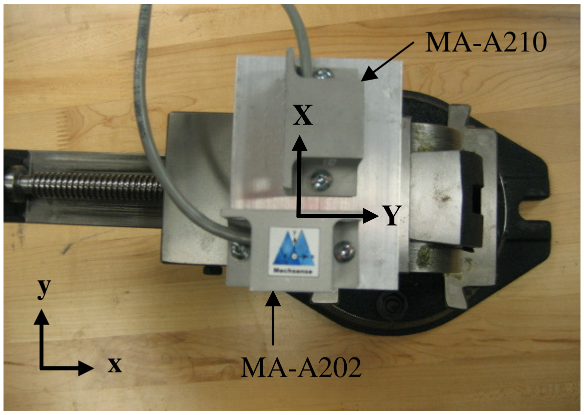


Figure 3-13: Triaxial accelerometer which consists of two biaxial accelerometers, the sensor axes (XY), and the fixed frame axes (xy).

After the sensor is calibrated, it was held stationary on a milling vise at 0°, 30°, 45°, 60°, and 90° with respect to the gravity vector as shown in Figure 3-8 for validation. Three calibration tests were conducted. Each test comprised two different sets of initial estimated biases: 2.5V and 10V. Figure 3-14 shows the simulation and experiment results when the initial estimated biases are close to the true biases of the accelerometer system ($\hat{B}_{axis,0}$= 2.5V), and Figure 3-15 shows the simulation and experiment results of the calibration when the initial estimated biases are far from the true biases ($\hat{B}_{axis,0}$= 10V). The experimental results of the Mechworks triaxial accelerometer have the same trend as the experimental results of SF3000L. Figure 3-14 demonstrates that the experiment steps agree with the simulation steps while Figure 3-15 does not. In both cases, the experimental results reveal that the calibration parameters converge to the same values within three iterations.

The experimental results of Experiment 3 in Figure 3-14 are expanded in Figure 3-16. Figure 3-16 (a) shows the acceleration measurements, and Figure 3-16 (b) shows the tilt angle measurements converted from the acceleration measurements by using (3-18). The first 36 seconds of Figure 3-16 reflect the calibration period where the initial estimated gains of each axis are unity and the initial estimated biases of each axis are 2.5 V. Since the Mechworks accelerometers have true gains less than 0.04 V/(m/s^2) and the initial estimated biases almost match the true biases, the first 36 seconds of Figure 3-16 (a) shows the acceleration signal around 0 V with very small fluctuations. However, after calibration, the two figures in Figure 3-16 illustrate that the acceleration measurements match the accelerations induced by the rotation sequence illustrated in Figure 3-8.

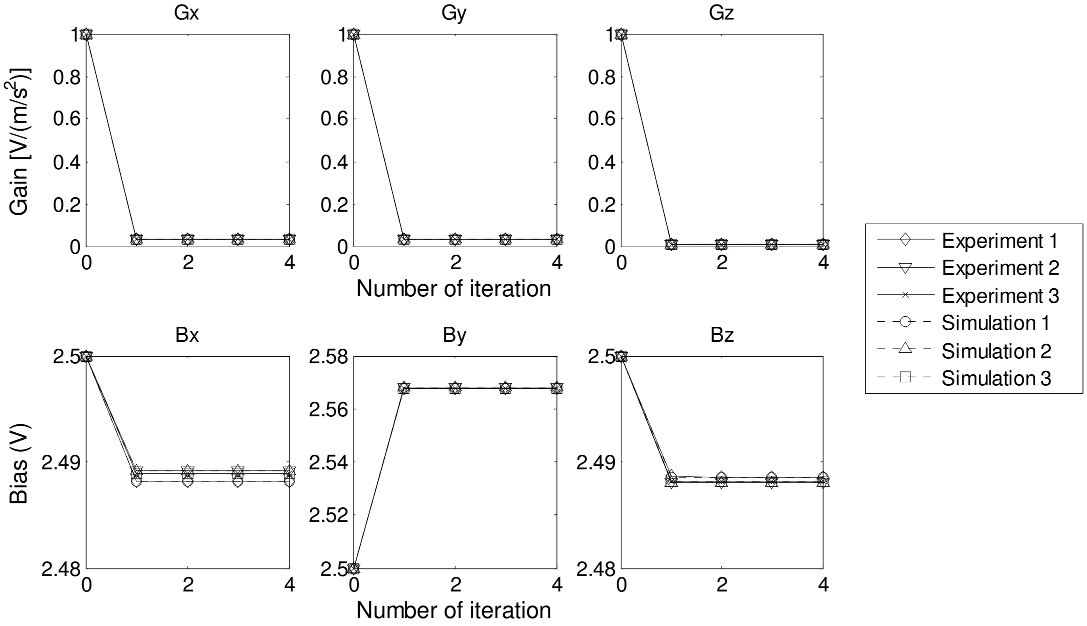


Figure 3-14: Calibration of the Mechworks triaxial accelerometer system when the initial estimations of biases are 2.5V: experiment and simulation results.

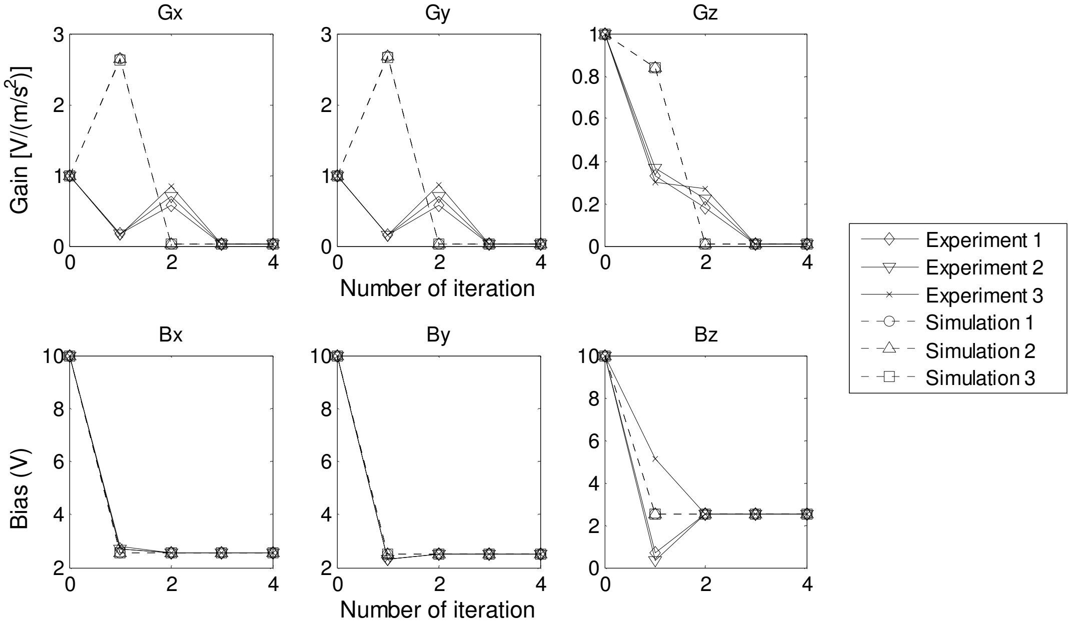


Figure 3-15: Calibration of the Mechworks triaxial accelerometer system when the initial estimations of biases are 10V: experiment and simulation results.

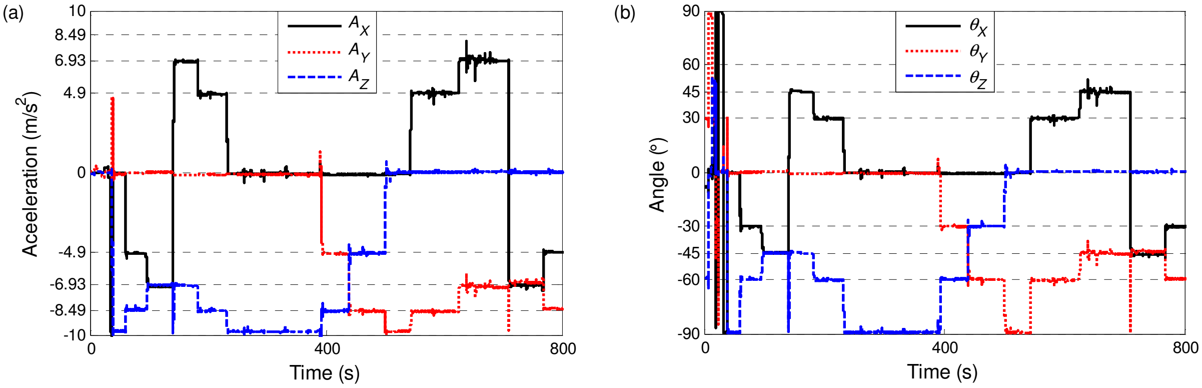


Figure 3-16: Measurements with the Mechworks triaxial accelerometer: (a) acceleration measurements and (b) tilt angle measurements using (3-18).

The specifications of the Mechworks accelerometers denote that they have a 0.2% nonlinearity error, 1° misalignment error, and 2% cross-axis sensitivity. The accelerometers were connected to a computer through the same data acquisition card that is used to connect the Colibrys triaxial accelerometer. Figure 3-17 relates the acceleration RMS error, the maximum error, and the margin of error of MA-A202 and MA-A210 with the data acquisition card. Figure 3-17 shows that the acceleration RMS error is 0.10 m/s^2 at 0° and tends to decrease as the tilt angle increases. The maximum acceleration error is 0.16 m/s^2 at 0° and tends to decrease as the tilt angle increases. The maximum error of the system is within the margin of error of both accelerometer systems. The experimental results of the Mechworks triaxial accelerometer demonstrate that the proposed calibration method can accurately estimate the gains and biases of a triaxial accelerometer with different gains.

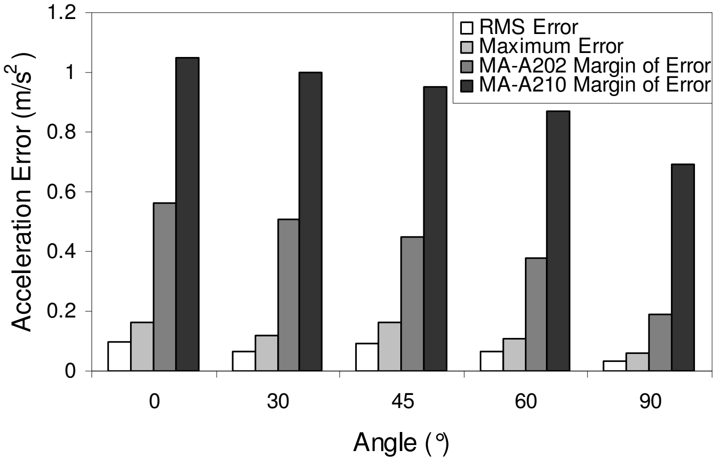


Figure 3-17: Error comparison among the acceleration RMS error, the maximum acceleration error, and the margins of error of the MA-A202 and MA-A210 accelerometer system.

3.5 Conclusion

This chapter presented a novel calibration method that determines the gains and biases of a triaxial accelerometer by placing the sensor in six different tilt angles. The presented method was developed by using the mathematical model of gains and biases. To validate the proposed calibration method, simulations and experiments were performed.

The simulation results demonstrated that the proposed calibration method accurately estimated the gains and biases within three iterations. Since the true gains and biases were not available in the experiments, the accelerations measured with an accelerometer after calibration were compared with the true accelerations calculated from the gravity vector to validate the estimated gains and biases. The first set of experiments was performed by using a triaxial accelerometer consisting of three single-axis accelerometers that have almost the same gains and biases. The acceleration RMS error of the accelerometer after calibration was 0.09 m/s^2, and the maximum error was within the margin of error of the accelerometer system. The second set of experiments was conducted by employing a triaxial accelerometer that has two different gains and almost the same biases. The acceleration RMS error of the Mechworks triaxial accelerometer after calibration was 0.10 m/s^2, and the maximum error was within the margin of error of the accelerometer system.

From the simulations and experiments, it is concluded that the proposed calibration method can be adapted to estimate the calibration parameters accurately. The proposed calibration method follows a very simple procedure and has a low computational cost. In addition, this method does not require any prior knowledge of the accelerometer calibration parameters nor the six tilt angles that are needed to implement the iterative approach for the proposed calibration method.

The proposed method is especially useful for a low cost triaxial accelerometer whose initial estimated gains and biases highly vary from power-on to power-on because the simulation and experiment results indicate that the gains and biases can be accurately estimated even when the initial estimated gains and biases are not close to the true value.

Chapter 4
Fastened Bolt Tracking System Using an IMU and a Triaxial Magnetometer with an Expert System

4.1 Proposed Tracking System

In some automotive parts, the bolts are not closely placed or the bolt orientations are different. In such cases, a cost-effective position/orientation tracking solution with a sufficient accuracy is desirable. This chapter proposes a cost-effective fastened bolt tracking system by using an IMU and a triaxial magnetometer.

In order to calculate position from IMU measurements, the accelerations need to be integrated twice and the angular velocities need to be integrated once. However, since the sensor errors are also factored in the integration steps, the position errors increase over time as shown in Figure 1-1. In order to reduce the position error of a down-hole drill, ZUPT was utilized in [111]. ZUPT zeros velocity whenever the object is stationary, and the position error is reduced by subtracting the position accumulation error caused by the average velocity error. To achieve high accuracy with ZUPT, a short time span between the stationary states is desirable. Thus, ZUPT is widely used to track the location of a person by attaching a low cost IMU on a shoe [112]-[114] because the time span between each foot step (stationary state) is very short.

For a fastening procedure of the manufacturing environment, an operator picks up a tool from the tool holder and fastens all the bolts, and then places the tool back to the tool holder. This entire procedure usually takes less than one minute. The tracking system should be able to detect when the tool is placed on the tool holder to correct the position and orientation error. In addition, the system should be able to detect the stationary state and the fastening action to use ZUPT.

The method to accomplish the aforementioned is described in Figure 4-1. First, the IMU must be calibrated to estimate the position and orientation as accurately as possible. By using the calibrated IMU measurements, the position and orientation of the tool are calculated. Even if an IMU is well-calibrated, the position error grows over time due to the integration steps. Therefore, an expert system is proposed to correct the position, velocity, and orientation errors. By using the corrected position and orientation of the IMU, the tool tip position and orientation are calculated.

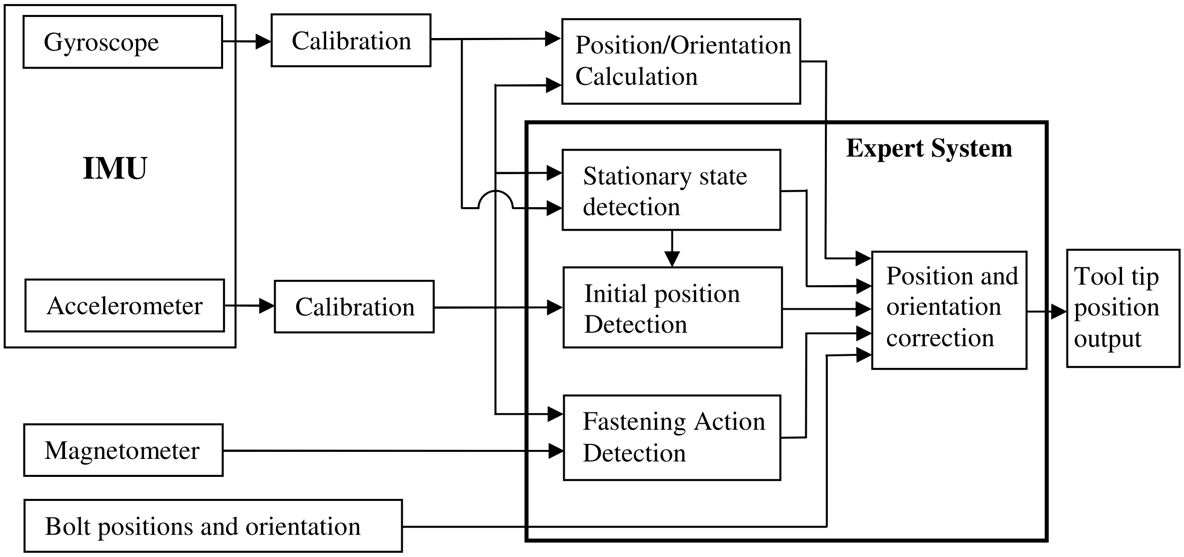

IMU
Gyroscope
Accelerometer
Calibration
Calibration
Magnetometer
Bolt positions and orientation
Position/Orientation Calculation
Expert System
Stationary state detection
Initial position Detection
Fastening Action Detection
Position and orientation correction
Tool tip position output

Figure 4-1: Overview of the position/orientation sensing system.

4.2 Orientation Representations

An IMU is rigidly strapped down to an object being tracked to measure the linear accelerations and angular velocities in terms of the object body frame. To define the orientation of the object with respect to a reference frame, a rotation matrix that describes the transformation from the body frame to the reference frame is required. In this section, two rotation matrix representations are described: the direction cosine matrix and the quaternion [115].

4.2.1 Direction Cosine Matrix

A direction cosine matrix consists of three unit vectors that represent three body axes projected on three reference axes. These three unit vectors form three columns of a direction cosine matrix. The direction cosine matrix from the tool frame to the local fixed frame is described as follows:

$$ {}_{t}^{f}C = \begin{bmatrix} c_{xX} & c_{yX} & c_{zX} \\ c_{xY} & c_{yY} & c_{zY} \\ c_{xZ} & c_{yZ} & c_{zZ} \end{bmatrix}, \tag{4-1} $$

where subscript x, y, z are the orthogonal unit vectors of the tool frame and X, Y, Z are the corresponding vectors of the reference frame. c_{mM} is the cosine of the angle between vector m in the body frame and vector M in the body frame.

4.2.2 Quaternion

The quaternion orientation representation is widely used to calculate the orientation of the body frame with respect to a reference frame. The quaternion representation obtains a rotation matrix with a single rotation about one axis in space. This method uses a concept of the hyper-complex number of rank 4: one real number and three imaginary numbers. A quaternion q is written as

$$q = q_0 + q_v = q_0 + (q_1 i + q_2 j + q_3 k) . \qquad (4\text{-}2)$$

The imaginary numbers follow the right hand rule such that

$ij = k$, $ji = -k$, $jk = i$, $kj = -i$, $ki = j$, $ik = -j$, and $i^2 = j^2 = k^2 = -1$.

In order to use quaternion to determine the orientation in a 3D vector space, the real part of a quaternion v needs to be zero ($v = 0 + v_x i + v_y j + v_z k$). This also means that the real part of a 3D vector v should be zero once rotated by quaternion q. To satisfy this condition, the 3D vector is multiplied by quaternion q and the conjugate of quaternion q, denoted as q^*, as follows:

$$v' = q \cdot v \cdot q^* , \qquad (4\text{-}3)$$

where vector v' is the rotated 3D vector. The conjugate of the quaternion q is defined as

$$q^* = q_0 - q_v = q_0 - q_1 i - q_2 j - q_3 k . \qquad (4\text{-}4)$$

The resultant multiplication of (4-3) should form a cosine matrix multiplication by the 3D vector v expressed as

$${}^f_tC \cdot v = q \cdot v \cdot q^* . \qquad (4\text{-}5)$$

From (4-5), the rotation matrix becomes

$${}^f_tC = \begin{bmatrix} {q_0}^2 + {q_1}^2 - {q_2}^2 - {q_3}^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & {q_0}^2 - {q_1}^2 + {q_2}^2 - {q_3}^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & {q_0}^2 - {q_1}^2 - {q_2}^2 + {q_3}^2 \end{bmatrix} . \qquad (4\text{-}6)$$

When a unit quaternion is used, (4-2) should satisfy

$$q_0^{\,2} + (q_1^{\,2} + q_2^{\,2} + q_3^{\,2}) = 1\,. \qquad (4\text{-}7)$$

Then, (4-2) can be written as

$$q = q_0 + q_v = \cos\theta + u\sin\theta\,. \qquad (4\text{-}8)$$

where u is a unit vector ($u = u_x i + u_y j + u_z k$). Then, when the vector v is rotated by angle θ by applying (4-3), the quaternion components of q become

$$q_0 = \cos\frac{\theta}{2},\ q_1 = u_x \cdot \sin\frac{\theta}{2},\ q_2 = u_y \cdot \sin\frac{\theta}{2},\ \text{and}\ q_3 = u_z \cdot \sin\frac{\theta}{2}\,. \qquad (4\text{-}9)$$

The rotation angle is divided by a factor of 2 in (4-9) because the vector v was rotated by θ twice: about both q and q*. The detailed derivation of quaternion orientation representation is given in Appendix A.

Since the quaternion orientation representation and the direction cosine matrix should match, the following equality is true:

$${}^f_t C = \begin{bmatrix} c_{xX} & c_{yX} & c_{zX} \\ c_{xY} & c_{yY} & c_{zY} \\ c_{xZ} & c_{yZ} & c_{zZ} \end{bmatrix} = \begin{bmatrix} q_0^{\,2} + q_1^{\,2} - q_2^{\,2} - q_3^{\,2} & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^{\,2} - q_1^{\,2} + q_2^{\,2} - q_3^{\,2} & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^{\,2} - q_1^{\,2} - q_2^{\,2} + q_3^{\,2} \end{bmatrix}. \qquad (4\text{-}10)$$

4.3 Position Estimation

Since accelerometers measure accelerations with respect to the inertial frame, the motion equations using accelerometers are derived from the inertial frame. However, the local fixed frame is attached to a stationary building which, in fact, moves with respect to the inertial frame in this application. Therefore, the equations of motion should be expressed with respect to the local fixed frame. The acceleration of an object with respect to the local fixed frame can be expressed as [116]

$${}^f\dot{V} = {}^fA - 2\cdot\omega_e \times {}^f V - {}^f[\omega_e \times \omega_e \times R] + {}^f g\,, \qquad (4\text{-}11)$$

where ω_e is the angular velocity of the earth with respect to the inertial frame and R is the location of the tool from the centre of the earth. In this application, the centripetal force term, $\omega_e \times [\omega_e \times R]$, is caused by the angular velocity of the earth and changes as the location of the tool with respect to the centre of the earth, R, changes. However, the change of R is small for the tool tracking applications (maximum 4 m), and the multiplication of the square of angular velocity of the earth makes the change in value insignificant. Therefore, the centripetal force due to the rotation of the earth can be considered constant. Combining the centripetal force with the gravitational force results in different local gravity values which can be written as:

$$^{f}g_{l}={}^{f}g-\omega_{e}\times{}^{f}[\omega_{e}\times R]. \tag{4-12}$$

The Coriolis force, $2\cdot\omega_{e}\times{}^{f}V$, can be omitted when the velocity of the object is small because the angular velocity of the earth is very small (7.3×10^{-5} rad/sec). Since the velocity of the tool is small (usually less than 1 m/s) in the tool tracking application, the Coriolis term is much lower than the noise level of the accelerometers; hence this term is ignored. Then, (4-11) can be simplified as

$$^{f}\dot{V}={}^{f}A+{}^{f}g_{l}. \tag{4-13}$$

The specific force measurements are represented with respect to the tool frame because they are measured with accelerometers. To change the specific force measurement terms from the tool frame to the local fixed frame, a rotation matrix is required. Thus, (4-13) is rewritten as

$$^{f}\dot{V}={}^{f}_{t}C\cdot{}^{t}A+{}^{f}g_{l}. \tag{4-14}$$

The rotation matrix from the tool frame to the local fixed frame can be determined from (4-10). Since the Z-axis of the local fixed frame is chosen in the opposite direction of the local downwards, the local gravity vector is expressed as

$$^{f}g_{l}=\begin{bmatrix}0 & 0 & |g_{l}|\end{bmatrix} \tag{4-15}$$

The velocity and the position of the object can be calculated by integrating over time as follows:

$$^{f}V=\int{}^{f}\dot{V}\cdot dt, \tag{4-16}$$

$$^{f}P=\int{}^{f}V\cdot dt, \tag{4-17}$$

where *P* is the position of the object.

4.4 Expert System

4.4.1 Stationary State Identification

When the tool is stationary, its velocity is zero, but the calculated velocity using an IMU is usually not zero. Thus, the calculated velocity should be corrected to zero when the tool is stationary. To identify the stationary state, the IMU measurements are utilized. When an object is stationary, the angular velocity is zero, the acceleration in each axis is constant, and the magnitude of the acceleration is equal to the magnitude of the gravity vector. However, the angular velocity measurements are typically not zero because a gyro has several sources of error such as noise. The acceleration measurements also contain noise, and the magnitudes of the acceleration measurements due to gravity vary depending on the tilt angles because of the non-linearity. To identify the stationary state from these inaccurate measurements, an expert system is incorporated by using acceleration measurements and angular velocity measurements. Let the magnitude of gravity-free acceleration (*Acc*) and the magnitude of angular velocity (*Ang_vel*) be expressed as

$$Acc = \left| ({A_x}^2 + {A_y}^2 + {A_z}^2)^{0.5} - |g_l| \right|, \tag{4-14}$$

$$Ang\_vel = ({\omega_x}^2 + {\omega_y}^2 + {\omega_z}^2)^{0.5}. \tag{4-15}$$

When the IMU is stationary, the accelerations of each axis should be constants. Thus, the fluctuation of the acceleration measurements in each axis should be less than the maximum acceleration noise. The acceleration fluctuation (*Acc_fluc*) is defined as

$$Acc\_fluc = ((A_x - Avg\_A_x)^2 + (A_y - Avg\_A_y)^2 + (A_z - Avg\_A_z)^2)^{0.5}, \tag{4-16}$$

where the average accelerations are expressed as

$$\begin{aligned} Avg\_A_x &= \sum_{i=1}^{n} A_x(i)/n, \\ Avg\_A_y &= \sum_{i=1}^{n} A_y(i)/n, \\ Avg\_A_z &= \sum_{i=1}^{n} A_z(i)/n, \end{aligned} \tag{4-17}$$

where *n* is the number of acceleration samples. While the IMU is stationary, *n* increases to obtain more steady average accelerations.

When the IMU is stationary, the gravity-free acceleration (*Acc*) should be lower than the maximum error of the accelerometers which includes non-linearity error and the maximum noise. In addition, the magnitude of the angular velocity (*Ang_vel*) should be lower than the maximum error of the gyro which includes random walk and the maximum noise. Since the tool is manually moved, the IMU measurements cannot maintain almost a constant acceleration in each axis and almost a zero angular velocity for 1 second. Therefore, the IMU is considered stationary when the following conditions are met:

Rule 4-1:

IF $Acc < \sqrt{(A_{x\_\max\_error})^2 + (A_{y\_\max\_error})^2 + (A_{z\_\max\_error})^2}$ for the last 1 second data AND

$Ang\_vel < \sqrt{(\omega_{x\_\max\_error})^2 + (\omega_{y\_\max\_error})^2 + (\omega_{z\_\max\_error})^2}$ for the last 1 second data AND

$Acc\_fluc < \sqrt{(A_{x\_\max\_noise})^2 + (A_{y\_\max\_noise})^2 + (A_{z\_\max\_noise})^2}$ for the last 1 second data,

THEN stationary = 1 (IMU is stationary)

ELSE stationary = 0 (IMU is moving)

where $A_{axis\_\max\_error}$ is the maximum acceleration error in each axis, $\omega_{axis\_\max\_error}$ is the maximum angular velocity error in each axis, and $A_{axis\_\max\_noise}$ is the maximum acceleration error in each axis.

When the IMU is stationary, the three accelerometers measure the tilt angles which are the angles between the gravity vector and the IMU body frame axes. To correct the tilt angles, the relationship between the tilt angles and the direction cosine matrix needs to be established. When the Z-axis of the reference frame is chosen to be in the opposite direction of the gravity vector, the tilt angle components of the direction cosine matrix from (4-1) are c_{xZ}, c_{yZ}, and c_{zZ}. The tilt angle components can be derived by normalizing the three acceleration components. To reduce the effects of the acceleration measurement noise, the average values are used to calculate the tilt angles instead of the instantaneous accelerations. The tilt angle components in terms of the average acceleration in each axis are

$$c_{xZ} = \frac{Avg\_A_x}{(Avg\_A_x{}^2 + Avg\_A_y{}^2 + Avg\_A_z{}^2)^{0.5}},$$
$$c_{yZ} = \frac{Avg\_A_y}{(Avg\_A_x{}^2 + Avg\_A_y{}^2 + Avg\_A_z{}^2)^{0.5}}, \tag{4-18}$$
$$c_{zZ} = \frac{Avg\_A_z}{(Avg\_A_x{}^2 + Avg\_A_y{}^2 + Avg\_A_z{}^2)^{0.5}}.$$

From (4-10), the quaternion terms in the tilt angles are

$$2(q_1q_3 - q_0q_2) = c_{xZ}, \tag{4-19}$$

$$2(q_2q_3 + q_0q_1) = c_{yZ}, \tag{4-20}$$

$$q_0{}^2 - q_1{}^2 - q_2{}^2 + q_3{}^2 = c_{zZ}. \tag{4-21}$$

There are four unknowns and four equations (4-7) and (4-19) to (4-21). However, the equations cannot be solved analytically because all four equations are in non-linear forms. To find the four quaternion terms, one of them is fixed, and the other three terms are corrected. From (4-7) and (4-21),

$$1 - 2q_1{}^2 - 2q_2{}^2 = 2q_0{}^2 + 2q_3{}^2 - 1 = c_{zZ}. \tag{4-22}$$

When q_0 is fixed, the corrected quaternion terms become

$$q_{0\_cor\_0} = q_0, \tag{4-23}$$

$$q_{3\_cor\_0} = sign(q_3) \times (\left|c_{zZ} + 1 - 2 \times q_{0\_cor\_0}^2\right| / 2)^{0.5}, \tag{4-24}$$

$$q_{1\_cor\_0} = \frac{c_{xZ} \times q_{3\_cor\_0} + c_{yZ} \times q_{0\_cor\_0}}{2 \times (q_{0\_cor\_0}^2 + q_{3\_cor\_0}^2)}, \tag{4-25}$$

$$q_{2\_cor\_0} = \frac{-c_{xZ} \times q_{0\_cor\_0} + c_{yZ} \times q_{3\_cor\_0}}{2 \times (q_{0\_cor\_0}^2 + q_{3\_cor\_0}^2)}, \tag{4-26}$$

where $q_{0\_cor\_0}$, $q_{1\_cor\_0}$, $q_{2\_cor\_0}$, and $q_{3\_cor\_0}$ are the corrected quaternion terms. Eq. (4-25) and (4-26) should be calculated after (4-23) and (4-24) are calculated because (4-25) and (4-26) use the corrected quaternion terms. Since a quaternion has four components, four correction possibilities are evaluated. The corrected terms are fed to (4-22) and the corrected terms that do not satisfy (4-22) are discarded. From all the possible quaternion corrections, the correction with the minimum RMS error is chosen as the best correction possibility. To prevent from further orientation drift, the angular velocity is set to zero when the IMU is stationary.

The calculated non-zero velocity at the stationary state represents the velocity error. The position of the IMU can be corrected by utilizing the velocity error at the stationary state [111], [117]. The velocity error is partially caused by the integral of accelerometer error such as bias, gain error, non-linearity, and noise. In addition, when the calculated orientation of the tool is not correct, the acceleration calculation in (4-14) produces an error, which results in the accumulation of the velocity error. When an IMU is accurately calibrated, it can be assumed that a large portion of the velocity error is caused by orientation errors. Since the tool traveling time spans between fastening bolts are short, it can be assumed that the velocity error is caused by a constant acceleration error which is caused by constant orientation errors. Then, position estimation can be corrected by using the following equation:

$$P_{axis} = P_{axis} - V_{axis} \cdot \Delta t / 2 \tag{4-27}$$

where P_{axis} is the current IMU position component in each axis, V_{axis} is the calculated velocity error in each axis at stationary state, and Δt is the time span between the current time and the previous velocity correction. Then, an expert rule is derived to correct the position and tilt angles as follows.

Rule 4-2: Position and orientation correction at stationary state.

```
IF      stationary = 1
THEN    $P_{axis} = P_{axis} - V_{axis} \cdot \Delta t / 2$ AND $\omega_{axis} = 0$, AND $V_{axis} = 0$ AND $\omega_x = 0$, $\omega_y = 0$, $\omega_z = 0$ AND
        correct the quaternion terms using (4-23) to (4-26) AND $n = n+1$
ELSE    $n = 1$
```

4.4.2 Initial Position Detection

The position and orientation of the tool can be corrected when the tool is placed in the tool holder. Such error correction is possible because the position and orientation of the tool in the tool holder is known. Therefore, the system should be able to identify if the tool is in the tool holder to correct the position and orientation constantly. In order to detect the initial position, the tool holder was designed so that the tool is placed at a specific tilt angle (θ) as shown in Figure 4-2. The top of the tool holder is shaped so that the bottom of the tool almost perfectly fits on it. However, there is a small gap between the top of the holder and the tool bottom to ensure that the tool can easily be placed on the holder. Therefore, it is possible for the tool to be rotated in the z-axis of the tool frame. In order to calculate the rotation matrix from the tool frame to the local fixed frame (${}_{t}^{f}C$), this misalignment angle in the z-axis, φ, should be calculated as well. The tilt angle, θ, and the misalignment in z-axis, φ, have the following relationship with the measured acceleration components and the gravity vector:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(-\theta) & 0 & -\sin(-\theta) \\ 0 & 1 & 0 \\ \sin(-\theta) & 0 & \cos(-\theta) \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -g_l \end{bmatrix}. \qquad (4\text{-}28)$$

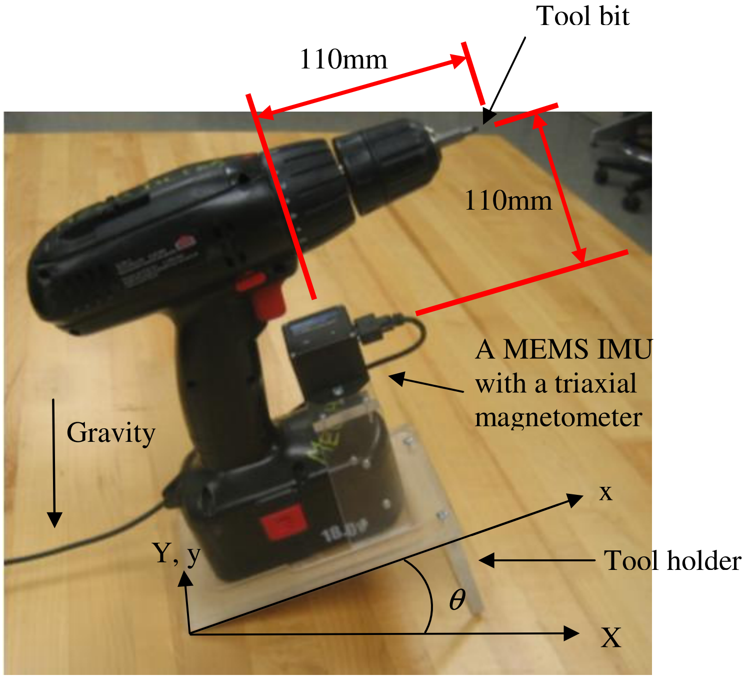


Figure 4-2: Fastening tool, the tool holder, and a sensor that consists of a triaxial magnetometer and an IMU. The tool frame is labeled xyz and the fixed frame is labeled XYZ.

From (4-28), φ and θ are calculated with the accelerometer measurements as follows:

$$\begin{aligned}\varphi &= \arctan(-Avg\_A_y / Avg\_A_x),\\ \theta &= -\arccos(Avg\_A_z / g).\end{aligned} \tag{4-29}$$

The fixed frame (XYZ) is chosen so that the Z-axis is the opposite direction of the gravity vector, and the x-axis of the tool is chosen so that it is parallel to the tool bit. Since (4-29) shows that A_z depends only on *θ*, A_z can be used to identify the tool holder angle, *θ*. When $\left|A_z - g \cdot \cos\theta\right|$ is less than the threshold (the maximum error of the accelerometer) and the tool is stationary, the system concludes that the tool is placed on the tool holder. Then, the quaternion terms of the rotation matrix from the tool frame to the fixed frame can be corrected by using *φ* and *θ* as:

$$\begin{aligned}&\begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \\ &= \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(-\theta) & 0 & -\sin(-\theta) \\ 0 & 1 & 0 \\ \sin(-\theta) & 0 & \cos(-\theta) \end{bmatrix}\end{aligned}. \tag{4-30}$$

There are ten equations {nine from (4-30) and one from (4-7)} and four unknowns. From (4-7) and three equations that only have square terms, the square of each quaternion terms are identified. Then, one of the four quaternion terms can be assumed either positive or negative, and the signs of the rest three quaternion terms can be identified using the rest of the equations.

The expert rule to identify the initial position and to estimate the orientation is derived as follows.

Rule 4-3: Position and angle correction using the tool holder.

IF stationary = 1 AND $\left|A_z - g \cdot \cos\theta\right| < Threshold$

THEN $P_{axis} = 0$ AND $\varphi = \arctan(-Avg\_A_y / Avg\_A_x)$ AND correct the quaternion terms by using (4-7) and (4-30)

where *Threshold* is the maximum tilt angle measurement error.

4.4.3 Fastening Action Detection

The fastening action needs to be detected to identify the fastened bolt. However, the high magnitude of vibration can cause high acceleration fluctuations. This may result in high velocity error in discrete time domain. In order to reduce the magnitude of acceleration measurements due to vibration, an IMU (3DM-GX2, Microstrain) is attached to the bottom of the tool as shown in Figure 4-2. This point has lower vibration than the top part of the tool because the gripping hand of an operator acts as a damper. Due to the reduced magnitudes of vibration measurements, acceleration measurements alone do not provide sufficient vibration to identify the fastening action. Thus, the proposed system uses both acceleration and magnetic field measurements to detect the fastening action.

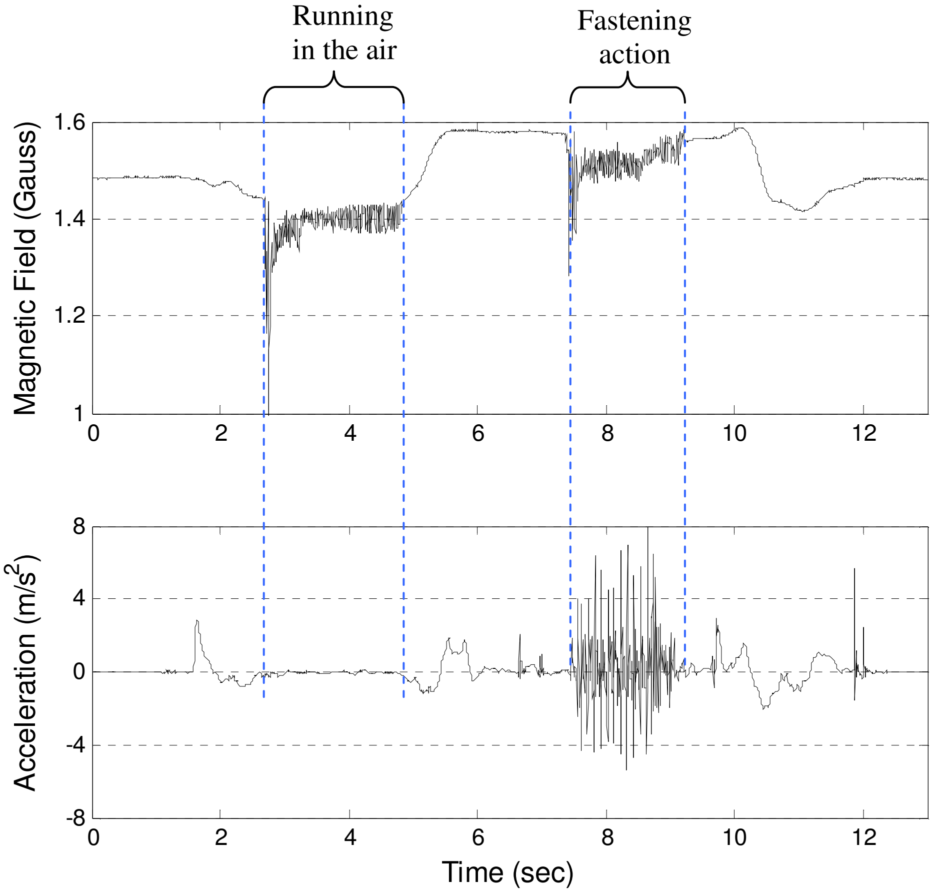


Figure 4-3: Comparison study between the fastening action and running the tool in the air: (a) magnitude of magnetic field and (b) magnitude of acceleration.

Figure 4-3 shows the magnetic field and the acceleration plots with 200 Hz sampling frequency when the tool is running in the air (from 3 second to 5 second) and is fastening a bolt (from 7 second to 9 second). The magnetic fields fluctuate in both scenarios, but the accelerations have high frequency vibration only when the tool fastens a bolt. By using these two distinct measurement signatures, the expert system can accurately detect a fastening action. From several test results, it has become expert knowledge that the tool is fastening a bolt if (i) there are at least 3 peaks with at least 0.02 Gauss difference between the magnitude of the current magnetic field measurement (*mag*) and the previous magnetic field measurement (*pre_mag*) in the last 0.1 second and (ii) there are at least two magnitude of gravity-free acceleration (*Acc*) peaks that are greater than 1 m/s^2 in the past 0.1 second. To identify this 0.1 second of time span, time count (fasten_time), which increases by one every time step, is used. When the system detects the fastening action, the time count is changed to 0.05 second so that continuous fastening action detection is possible without reaching the 0.1 second time limitation. The rules to identify the fastening action are described in Rule 4-3 to Rule 4-10.

Rule 4-3 to Rule 4-10: Fastening action detection

```
Rule 4-3:
IF      $Acc$ > 1 m/s$^2$ AND fasten_time ≥ (0.1 second × 200 Hz)
THEN    fasten_time = 0 AND Acc_peak = 0 AND Mag_peak = 0 AND Acc_peak_high = 1 AND
        Mag_peak_high = 0 AND Fasten_ Accel =0 AND Fasten_Mag = 0
Rule 4-4:
IF      $Acc$ < 1 m/s$^2$ AND Acc_peak_high = 1
THEN    Acc_peak = Acc_peak + 1 AND Acc_peak_high = 0
Rule 4-5:
IF      $Acc$ > 1 m/s$^2$ AND Acc_peak_high = 0
THEN    Acc_peak_high = 1
Rule 4-6:
IF      Acc_peak > 1 AND fasten _time < (0.1 second × 200 Hz)
THEN    Fasten_Accel = 1
Rule 4-7:
IF      $mag$ - $pre\_mag$ > 0.02 AND fasten_time < (0.1 second × 200 Hz)
THEN    Mag_peak_high = 1
Rule 4-8:
IF      $mag$ - $pre\_mag$ < -0.02 AND Mag_peak_high = 1
```

THEN Mag_peak = Mag_peak + 1 AND Mag_peak_high = 0

Rule 4-9:

IF Mag_peak > 3 AND fasten _time < (0.1 second × 200 Hz)

THEN Fasten_Mag = 1

Rule 4-10:

IF Fasten_Accel = 1 AND Fasten_Mag = 1 AND fasten_time < (0.1 second × 200 Hz)

THEN *Fasten* = 1 AND fasten_time = (0.05 second × 200 Hz) AND Acc_peak = 1 AND
Mag_peak = 3 AND Fasten_Accel = 0 AND Fasten_Mag = 0

where Acc_peak_high is an indicator that shows the acceleration is higher than 1 m/s^2, Acc_peak is the number of the gravity-free acceleration (*Acc*) peaks higher than 1 m/s^2, Mag_peak_high is an indicator that shows *mag* - *pre_mag* is greater than 0.02 Gauss, Mag_peak is the number of magnetic field peaks more than 0.02 Gauss difference, Fasten_Accel is an indicator that the tool is fastening a bolt based on acceleration measurements, Fasten_Mag is an indicator that the tool is fastening based on magnetic field measurements, and *Fasten* is the fastening action indicator. The specific numbers vary depending on the tool type and the location of the IMU.

The fastened bolt should be identified when the fastening action is detected. When a bolt is being fastened, the position of the tool tip and the position of the fastened bolt head coincide, and the fastened bolt angle is almost parallel to the tool bit angle. To identify the fastened bolt, the position and orientation of the tool tip was compared with the position and orientation of each bolt. When the gyros of 3DM-GX2 are calibrated, the orientation does not usually drift more than 3°/min. Since one cycle of fastening process (fastening all the bolts in one workpiece) usually takes less than one minute, the maximum orientation error of the system is 3°. On the other hand, position error can drift over 500 mm in 10 second when a commercial grade IMU is used as shown in Figure 1-1. Thus, it is the best to identify the fastened bolt using the orientation of the tool rather than the position of the tool tip.

While a tool fastens a bolt, the orientation of the fastened bolt is parallel to the tool bit. Since the x-axis of the body frame is parallel to the tool bit, the angle of the x-axis of the body frame is compared with the angles of all the bolts in the workpiece to identify the fastened bolt. However, a small gap between the tool bit and the bolt head can introduce a small orientation error. Therefore, the fastened bolt may not be parallel to the threaded hole at the beginning of the fastening. In addition, the orientation calculation drift due to the integration of angular velocity measurements should be accounted for. By considering all the error margins, the expert system concludes that the tool fastens one of the bolts which has less than a 15° error between the x-axis of the tool and the bolt angles as follows.

$$\begin{aligned} c_{xX} - O_{nx} &< 15° \\ c_{xY} - O_{ny} &< 15°, \\ c_{xZ} - O_{ny} &< 15° \end{aligned} \tag{4-31}$$

where O_{nx}, O_{ny}, and O_{nz} are the orientation of the n^{th} bolt with respect to the fixed frame.

If more than one bolt satisfies (4-31), the position information must be utilized to identify the fastened bolt. While the tool fastens a bolt, the tool tip can be assumed stationary. In addition, the angular velocity of the tool is almost zero because the tool bit fits in the bolt head and does not allow the tool to rotate very much. Therefore, the linear velocity of the IMU is set to zero while the tool fastens a bolt. Then, the position of the IMU can be corrected by using (4-27). After the position is corrected, the position error between the corrected tool tip position and the n^{th} bolt position (Err_n) is calculated as

$$Err_n = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + {}_t^f C \begin{bmatrix} 0.11 \\ 0 \\ 0.11 \end{bmatrix} - \begin{bmatrix} P_{nx} \\ P_{ny} \\ P_{nz} \end{bmatrix}, \tag{4-32}$$

where P_{nx}, P_{ny}, and P_{nz} are the position of the n^{th} bolt. The bolt with the minimum position error is chosen as the fastened bolt. Since the fastened bolt is identified, the position of the IMU (P_{axis}) can be corrected by using the position of the fastened bolt (P_{f-axis}) and the rotation matrix as:

$$\begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} = \begin{bmatrix} P_{fx} \\ P_{fy} \\ P_{fz} \end{bmatrix} - {}_t^f C \begin{bmatrix} 0.11 \\ 0 \\ 0.11 \end{bmatrix}. \tag{4-33}$$

Then, the fastened bolt identification expert rule is as follows.

Rule 4-11: fastened bolt identification

IF $\quad$ *Fasten* = 1

THEN $P_{axis} = P_{axis} - V_{axis} \cdot t_s / 2$, AND *Fastened_bolt* = bolt with minimum $(Err_n{}^T \cdot Err_n)^{0.5}$ AND

$V_{axis} = 0$ AND correct the IMU position using (4-33) AND *Fasten* = 0

ELSE fasten _time = fasten _time+1

where *Fastened_bolt* represents the fastened bolt number.

4.5 Experiments

The proposed fastened bolt tracking system was tested with a system consisting of a MEMS IMU and a triaxial magnetometer shown in Figure 4-2. In order to obtain the true position and velocity, an ultrasonic position sensor (CMS10, Zebris) is attached to the IMU. The tool fastens four bolts on an aluminum tube shown in Figure 4-4. There are two sets of bolts with the same orientation, and these bolts are 410 mm apart from each other. The tool fastened from Bolt 1 to Bolt 4 in sequence and then placed back on the tool holder. The IMU sensors were sampled at 200 Hz.

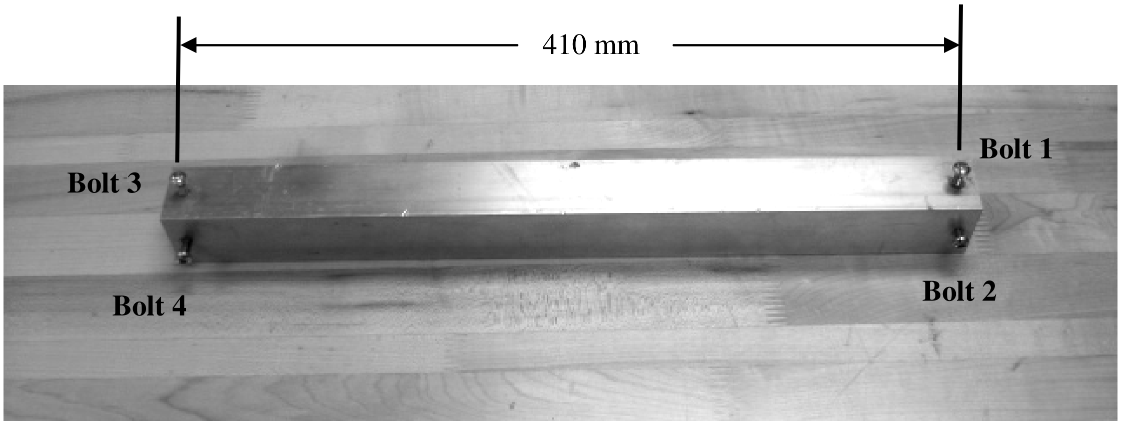


Figure 4-4: Workpiece with four bolts. Bolt 1 and Bolt 3 have the same orientation and Bolt 2 and Bolt 4 have the same orientation.

Figure 4-5 depicts the magnitude of magnetic field, the magnitude of acceleration, and the identified fastened bolt number during the experiment. The figures indicate that the proposed system identified the fastened bolt in the correct sequence.

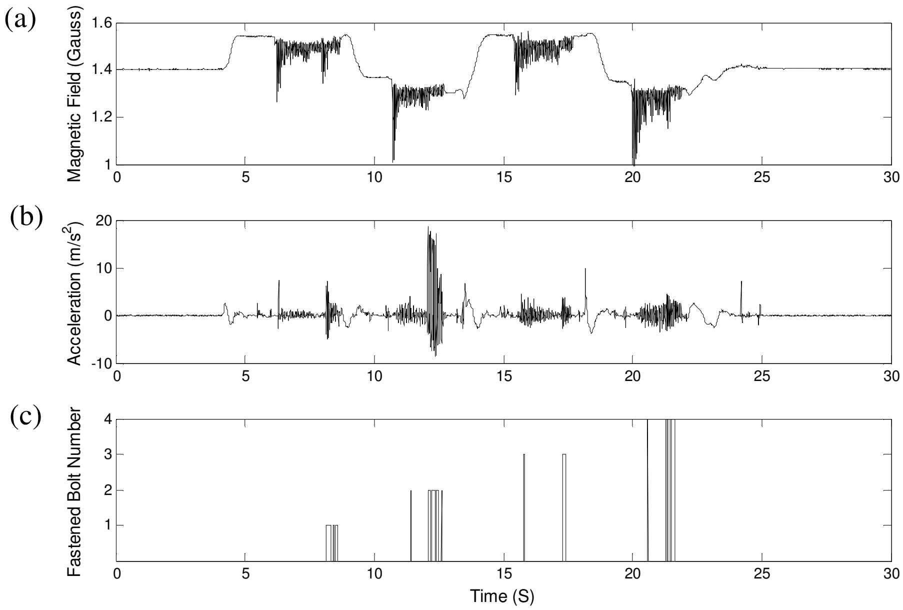


Figure 4-5: (a) Magnitude of magnetic field, (b) magnitude of acceleration, and (c) identified fastened bolt number by the proposed method.

Figure 4-6 shows the velocity comparison among the true values, using the traditional navigation system, and using the proposed method. Figure 4-6 illustrates that the true velocity almost matches the velocity using the proposed method because the velocity error is corrected whenever the fastened bolts are detected. In addition, the velocity is zeroed at 26 seconds when the tool was placed on the tool holder. However, the velocity calculation by using the conventional navigation equations drifts over time.

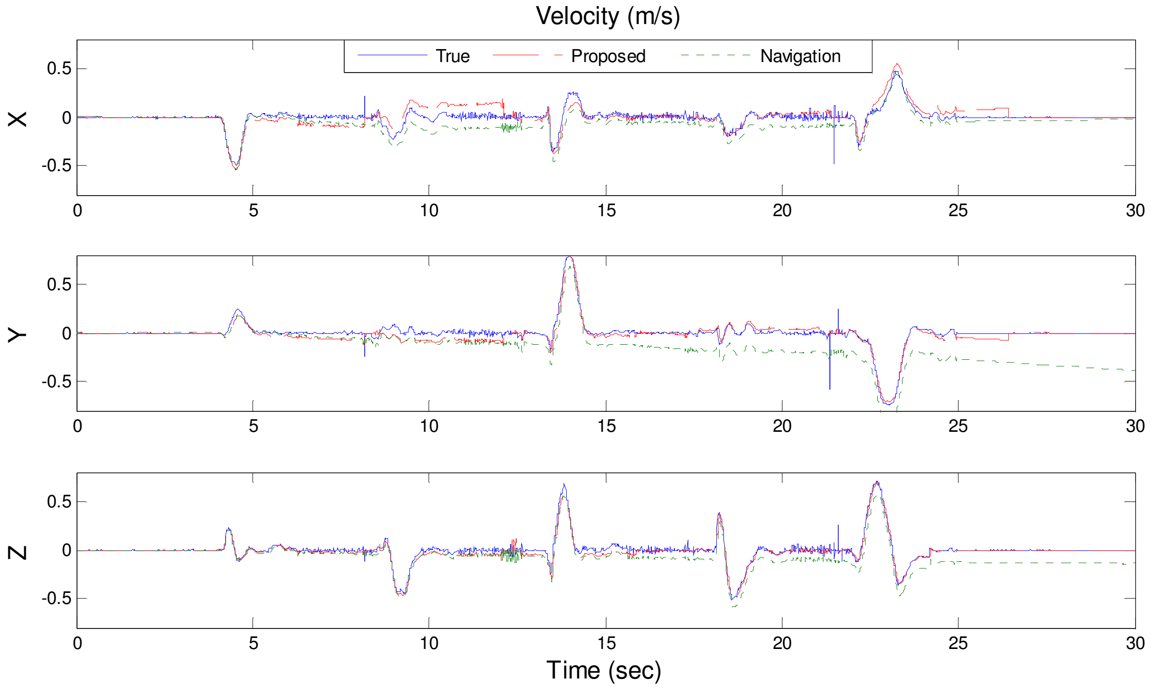


Figure 4-6: Velocity comparison in each axis among the true values measured with an ultrasonic sensor, the calculated values using the conventional navigation equations, and the calculated values using the proposed method.

Figure 4-7 displays the position comparison results between the true values measured with the ultrasonic position sensor and the calculated values by using the proposed system. Figure 4-7 shows the position comparison between the calculated values by using the proposed method and the true values. When the fastening action is detected, the expert system corrects the position by using (4-27), which is labeled "After position correction using ZUPT" (at 8.155 s) in Figure 4-7 (b). Then, by comparing the orientation and the corrected position of the tool tip with the possible bolt positions and orientations, the fastened bolt is identified. After the fastened bolt is identified, the expert system calculates the position of the IMU using (4-33), which is labeled "After position correction using bolt position" in Figure 4-7 (b). As a result, the position error is almost zero at 8.16 seconds as shown in Figure 4-7 (b). The position error comparison before and after the position corrections using ZUPT are shown in Table 4-1. This table shows that the fastened bolt can be identified more accurately by using ZUPT.

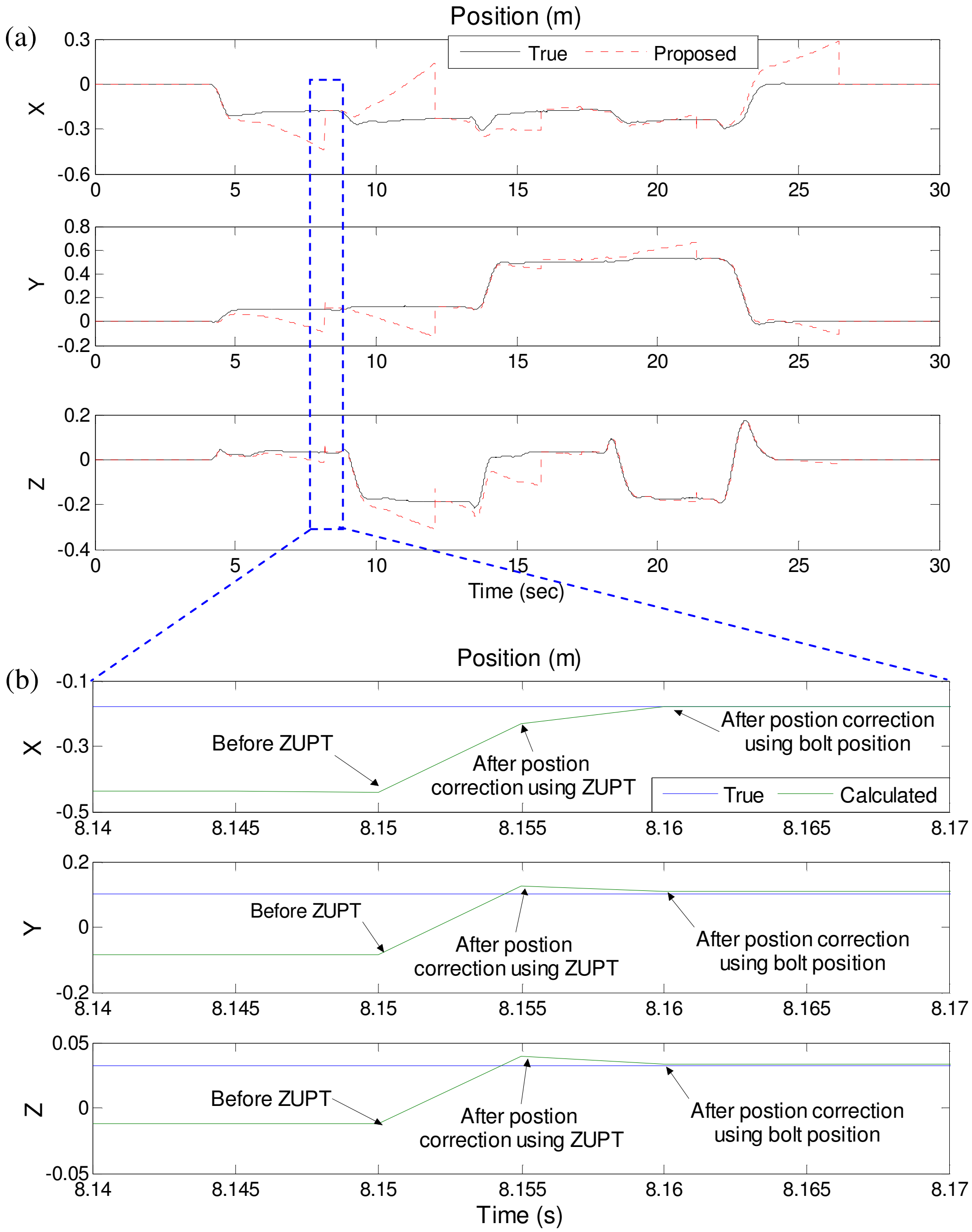


Figure 4-7: Position comparison in each axis between the true values measured with an ultrasonic position sensor and the calculated values using the proposed method (a) entire time span and (b) magnified when Bolt 1 is fastened – before and after ZUPT, and after the position correction using (4-33).

Table 4-1: Position Error using the proposed method before and after position correction.

	Position error before position correction (mm)				Position error after position correction using ZUPT (mm)			
Bolt	X	Y	Z	Total	X	Y	Z	Total
1	-261	-198	-46	331	-52	14	6	54
2	370	-249	-125	463	100	72	-31	127
3	-138	-77	-153	220	-30	-6	69	75
4	34	128	-12	132	-66	-6	-22	70

4.6 Conclusion

This chapter presented a novel cost-effective fastened bolt tracking system that does not require a line of sight. Such a system is intended for a workpiece that has different bolt orientations or long distances between the bolts. The position and orientation of the tool are estimated by the use of an IMU, and the fastening action is detected utilizing a triaxial accelerometer and a triaxial magnetometer. The inaccuracies in position, velocity, and orientation are corrected by using an expert system.

The expert system identifies the stationary state, initial position, and fastening action. When the tool is stationary, the system corrects the velocity errors, tilt angle errors, and reduces the position error. When the tool is on the tool holder, both the calculated position and orientation of the tool are corrected. When the tool fastens a bolt, the system identifies the fastened bolt and corrects the velocity and position error.

The experimental results indicate that the proposed system can identify the fastened bolt when the time span between the fastening actions is short. When the bolts with the same orientation are closely placed or the time span between the fastening actions is longer, a more accurate IMU must be used. Therefore, the tradeoff between the required accuracy and the sensor cost should be decided depending on the application.

Chapter 5
Fastening Tool Tracking System Using an IMU and a Position Sensor with Kalman Filters and a Fuzzy Expert System

In the previous chapter, a fastened bolt tracking system using an IMU and a triaxial magnetometer was presented. This system is low-cost and does not require a line of sight, but the time span between the fastening bolts must be short or the bolt orientation must be different to correctly identify the fastened bolts. In this chapter, a fastening tool tracking system consisting of an IMU and a position sensor is presented. The presented system uses KFs and a fuzzy expert system to track the position of the tool tip and to identify the fastened bolt.

5.1 Position Sensor Selection for Tool Tracking System

Selecting a position sensor depends on the required accuracy and the environment of the application. Table 1-1 describes the accuracy, advantages and limitations of various position sensors. The error of RF position sensors is too high for precision applications such as fastening tool tracking. When the UWB position sensor is hybridized with an IMU, the accuracy might be improved enough to be used for the fastening tool tracking system. However, the signal strength can be attenuated by human tissue, and signals can be reflected by metal as discussed in subsection 2.1.3. These drawbacks make UWB position sensors unsuitable for a tool tracking system because a fastening tool is always used in the proximity of an automotive metal part and an operator can easily block the lines of sight between the emitter and the receivers. An electromagnetic position sensor is inappropriate for an automotive manufacturing environment because magnetic fields are distorted by ferrous parts as discussed in Chapter 1.

The ultrasonic position sensor requires three lines of sight and has performance issues related to sound reflections and sound noise sensitivity. Since most automotive factories are subjected to sound noise, an ultrasonic position sensor is not a reliable solution for a fastening tool tracking system. The camera-based position estimation techniques require a complex calibration procedure and the procedure can be time consuming. Since the whole assembly line can be stopped during calibration, a time consuming calibration procedure is not desirable. Infrared position sensors are too expensive to be installed for all fastening operations.

A string-encoder position sensor can track 3D position with a high accuracy and provides noise-free outputs. The string-encoder position sensor is a feasible option for a fastening tool tracking system because fastening tools are often hung on a balancer in automotive factories as shown in Figure 5-1. Since the position sensor acts as a balancer and support a tool, the end of the wire must be connected to the centre of mass of the tool. This enables the position sensor to locate the centre of mass of the tool instead of the tool tip. To determine the position of the tool tip with a string-encoder position sensor, an IMU is used as an orientation sensor.

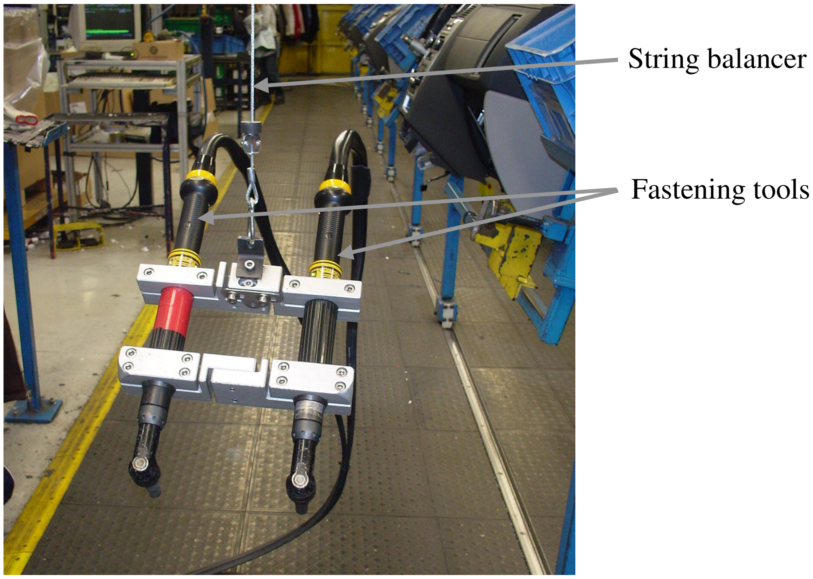


Figure 5-1: Two fastening tools are attached to a string balancer on the centre of mass of the tools

5.2 Tool Tracking System Design with an IMU and an Encoder-Position Sensor

A string-encoder position sensor provides the location of the centre of mass of a tool, and an IMU provides the orientation of the tool. With this sensor configuration, the position of the tool tip can be calculated as

$$\begin{bmatrix} T_x & T_y & T_z \end{bmatrix} = \begin{bmatrix} P_x & P_y & P_z \end{bmatrix} + {}^{f}_{t}C \begin{bmatrix} L_x & L_y & L_z \end{bmatrix}^T . \tag{5-1}$$

The location from the tool tip to the centre of mass of the tool with respect to the tool frame, [L_x L_y L_z], is fixed and can be predetermined. The location of the centre of mass of the tool is obtained by the position sensor. The rotation matrix from the tool frame to the local fixed frame, ${}_{t}^{f}C$, is obtained by integrating angular velocity measurements from the gyros of the IMU.

In order to compensate for the orientation drift, an intelligent system which utilizes KFs and a fuzzy expert system is proposed. Figure 5-2 depicts the structure of the proposed system. An IMU consists of accelerometers and gyros that have biases and gains. Thus, accelerometers and gyros should be calibrated before use. The position sensor consists of encoders which always start from zero whenever the sensor is turned on. Therefore, the position sensor should be initialized by locating the sensor to a known position. A KF is used to estimate the orientation and the angular velocities of the tool with gyros, and another KF is employed to estimate the positions and accelerations of the centre of mass of the tool with an IMU and an encoder-based position sensor. Although two KFs can be combined, they are separated because the position KF requires orientation information, and using two KFs is computationally cheaper. The workpiece information provides the locations and the orientations of all bolts. All the information is processed with an expert system to identify the fastened bolt and to correct the orientation error. When the expert system detects a fastening action, the fastened bolt is identified. Then, the complete 3D orientation of the tool can be determined by using the location and the orientation of the fastened bolt and the location of the position sensor; thus, the orientation error can be corrected. When the tool is stationary, the tilt angles of the tool are corrected. The fuzzy expert system outputs the orientation and position of the centre of mass of the tool. Then, the location of the tool tip is estimated by using (5-1).

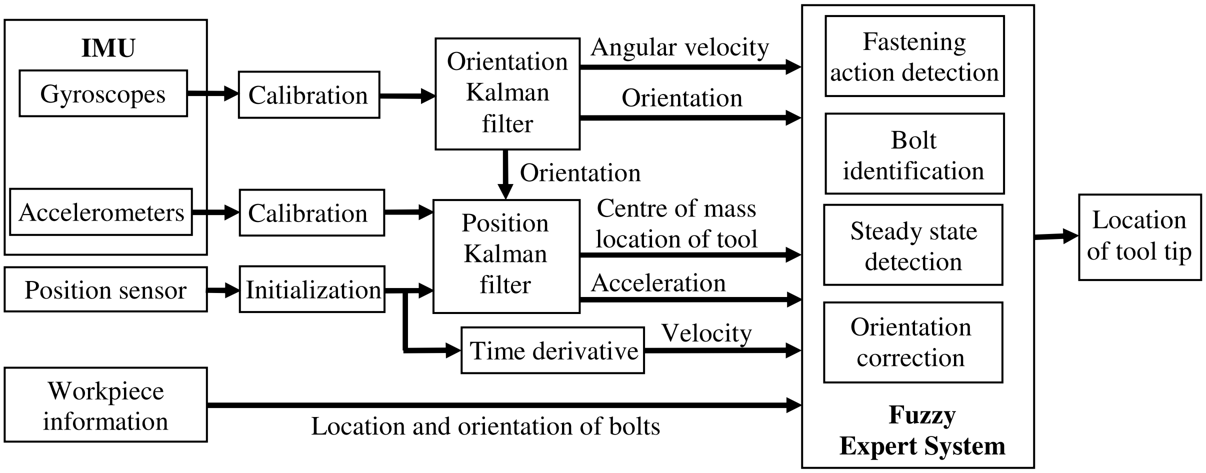


Figure 5-2: Overview of the fastening tool tracking system

5.2.1 Orientation Calculation Using Kalman Filtering

Gyros measure angular velocities with respect to the inertial frame whose origin is at the centre of the earth and its axes are stationary with respect to the fixed stars [116]. However, the tool tracking system is used in a small area, and the local fixed frame is attached to a stationary building which rotates 360°/day with the earth. Thus, the position and orientation changes due to the rotation of the earth must be discarded. In order to discard the angular velocity of the earth, gyros should be calibrated so that the angular velocity is zero when the IMU is stationary with respect to the local fixed frame. Even when the rotation rate of the earth is not accounted for the calibration, the earth rotation rate does not make any difference because the rotation matrix is corrected often and the rotation rate of the earth is only 7.27×10^{-5} rad/sec, which is often smaller than the noise level of a MEMS gyro. Therefore, the rotation rate of the earth can be discarded. Figure 5-3 shows the tool frame and the local fixed frame which is attached to the lab building. The Z-axis of the local fixed frame is chosen in the opposite direction of the local gravity vector, and the z-axis of the tool frame is chosen so that it is parallel to the bolt socket axis, which is parallel to the fastened bolt.

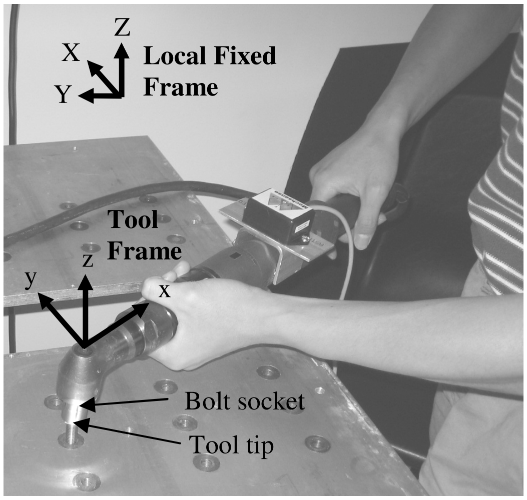


Figure 5-3: Local fixed frame and tool frame: the Z-axis of the local fixed frame is the opposite direction of the gravity vector and the z-axis of the tool frame is along the bolt socket axis.

To define the orientation of the tool with respect to the reference frame, the quaternion representation is used. To utilize the KF, the quaternion equation needs to be written in the form of (2-5). The differential equation of quaternion q with respect to time has the following matrix form:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} . \tag{5-2}$$

Since there are four quaternion states and three angular velocity measurements, the state of orientation, x_{ori}, becomes

$$x_{ori} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 & \omega_x & \omega_y & \omega_z \end{bmatrix}^T . \tag{5-3}$$

From (5-2) and (5-3), the system matrix of orientation in the form of (2-5) becomes

$$\Phi_{ori} = \frac{1}{2} \begin{bmatrix} 2 & 0 & 0 & 0 & -q_1 \cdot t & -q_2 \cdot t & -q_3 \cdot t \\ 0 & 2 & 0 & 0 & q_0 \cdot t & -q_3 \cdot t & q_2 \cdot t \\ 0 & 0 & 2 & 0 & q_3 \cdot t & q_0 \cdot t & -q_1 \cdot t \\ 0 & 0 & 0 & 2 & -q_2 \cdot t & q_1 \cdot t & q_0 \cdot t \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} . \tag{5-4}$$

The system input matrix of orientation, Γ_{ori}, is a zero matrix because the tool is rotated by a human operator. Also, since the quaternion states are estimated from the angular velocities, the process noise of the system is

$$b_{ori} = \begin{bmatrix} 0 & 0 & 0 & 0 & b_{ang} & b_{ang} & b_{ang} \end{bmatrix}^T , \tag{5-5}$$

where b_{ang} is the process noise of the angular velocity of the tool.

The angular velocities are measured by gyros, but there is no sensor that measures quaternion components. Since the gyros are calibrated, the measurement matrix for the orientation in the form of (2-6) becomes

$$H_{ori} = \begin{bmatrix} 0_{3x4} & I_{3x3} \end{bmatrix} . \tag{5-6}$$

To satisfy (4-7), all the quaternion components should be normalized after they are calculated.

The states of a KF converge on meaningful estimations when the states are observable. The observability matrix (*OM*) is calculated as

$$OM = \begin{bmatrix} H \\ H \cdot F \\ \vdots \\ H \cdot F^{m-1} \end{bmatrix}, \tag{5-7}$$

where *m* is the dimension of the state vector *x*. If the observability matrix has a rank of *m*, then the states are completely observable.

For the orientation KF, the rank of the observability matrix is three. In other words, only three angular velocities are observable, and the four quaternion terms may not converge to the correct values. Therefore, a quaternion term correction method needs to be implemented. The details of this correction method will be discussed in Section 5.3.

5.2.2 Position Kalman Filter Using an IMU and a Position Sensor

For the position estimating system, six measurements are available: three position components in the local fixed frame from the position sensor and three acceleration components in the tool frame from the IMU. From (4-14) and (4-15), the three acceleration measurements in the tool frame can be expressed in the local fixed frame in the form of (2-5) as

$$\begin{aligned} \dot{V}_x &= c_{xX} \cdot A_x + c_{yX} \cdot A_y + c_{zX} \cdot A_z \\ \dot{V}_y &= c_{xY} \cdot A_x + c_{yY} \cdot A_y + c_{zY} \cdot A_z \qquad . \\ \dot{V}_z &= c_{xZ} \cdot A_x + c_{yZ} \cdot A_y + c_{zZ} \cdot A_z + |g_l| \end{aligned} \tag{5-8}$$

By applying a constant acceleration motion model, the equations of velocity in the local fixed frame are

$$\begin{aligned} \dot{P}_x &= \dot{V}_x \cdot t + V_x \\ \dot{P}_y &= \dot{V}_y \cdot t + V_y \; . \\ \dot{P}_z &= \dot{V}_z \cdot t + V_z \end{aligned} \tag{5-9}$$

According to (5-8) and (5-9), the states of the position KF, x_{pos}, are defined as follows:

$$x_{pos} = \begin{bmatrix} P_x & V_x & A_x & P_y & V_y & A_y & P_z & V_z & A_z \end{bmatrix}^T . \tag{5-10}$$

The position and velocity components are represented in the local fixed frame, and the acceleration components are represented in the tool frame. From (5-8) and (5-9), the system dynamic matrix of object position in discrete time, Φ_{pos}, becomes

$$\Phi_{pos} = \begin{bmatrix} 1 & t & c_{xX} \cdot t^2/2 & 0 & 0 & c_{yX} \cdot t^2/2 & 0 & 0 & c_{zX} \cdot t^2/2 \\ 0 & 1 & c_{xX} \cdot t & 0 & 0 & c_{yX} \cdot t & 0 & 0 & c_{zX} \cdot t \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{xY} \cdot t^2/2 & 1 & t & c_{yY} \cdot t^2/2 & 0 & 0 & c_{zY} \cdot t^2/2 \\ 0 & 0 & c_{xY} \cdot t & 0 & 1 & c_{yY} \cdot t & 0 & 0 & c_{zY} \cdot t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & c_{xZ} \cdot t^2/2 & 0 & t & c_{yZ} \cdot t^2/2 & 1 & t & c_{zZ} \cdot t^2/2 \\ 0 & 0 & c_{xZ} \cdot t & 0 & 0 & c_{yZ} \cdot t & 0 & 1 & c_{zZ} \cdot t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (5\text{-}11)$$

The gravitational force is treated as a deterministic input. Since the Z-axis is parallel to the gravity vector, the velocity and the position changes due to the gravity vector should be compensated for in the Z-axis. Thus, the system input matrix of position is

$$\Gamma_{pos} \cdot u_{pos} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & |g_l| \cdot t^2/2 & |g_l| \cdot t & 0 \end{bmatrix}^T. \qquad (5\text{-}12)$$

Since the position and the velocity states are estimated with three accelerations, the process noise is

$$w_{k-1,pos} = q_{accel} \cdot \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}^T, \qquad (5\text{-}13)$$

where q_{accel} is the process noise of the tool acceleration.

The acceleration of each axis is measured by three accelerometers, and the position components are measured by a position sensor. Since both accelerometers and the position sensor are calibrated and initialized, the measurement matrix for the position estimation becomes

$$H_{pos} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (5\text{-}14)$$

The rank of the observability matrix of the position KF is nine. Therefore, all nine states are observable. This position KF utilizes the orientation of the tool which is estimated from the orientation KF. These filters reduce the measurement noise of accelerations and angular velocities. In the next section, the dynamic status of the tool such as stationary state is identified using the filtered measurements.

5.3 Fuzzy Expert System for Tool Tracking System

5.3.1 Tilt Angle Correction

The system in the previous chapter identifies the stationary state by using the IMU measurements. The presented system in this chapter identifies the stationary state of the tool by using three sets of states: (i) angular velocity, (ii) acceleration fluctuation, and (iii) linear velocity. The estimations of accelerations and angular velocities from the KFs are used instead of the direct measurements from the IMU because the estimated values from the KFs have less noise. However, the derivatives of the encoder-based position sensor outputs are chosen for the linear velocity calculation because they are zero when the tool is stationary.

Although the KFs are used to estimate acceleration and angular velocity states, only a portion of the noise is removed, and the linear acceleration and the angular velocity estimations still suffer from nonlinearity and random walk. To identify the stationary state from these measurements, a fuzzy expert system is proposed. The rules are developed based on the properties of the sensors in using linguistic terms. If the fluctuation of the acceleration estimation is high or the angular velocity estimation is high, the tool is not stationary even though the linear velocity from the position sensor is zero. This case occurs when the tool is hung on the wire of the position sensor, and the tool rotates about an axis: i.e., the wire of the position sensor. The fuzzy sets of the expert system inputs are shown in Figure 5-4 a) and b). The magnitude of angular velocity (*Ang_vel*) is calculated with angular velocity estimations after the KF by using (4-15) instead of using angular velocity measurements. Also, the acceleration fluctuation (*Acc_fluc*) is calculated with the acceleration estimations after the KF using (4-16). Figure 5-4 c) exhibits the dynamic states of the tool. Table 5-1 lists the fuzzy rules to identify the stationary state of the tool.

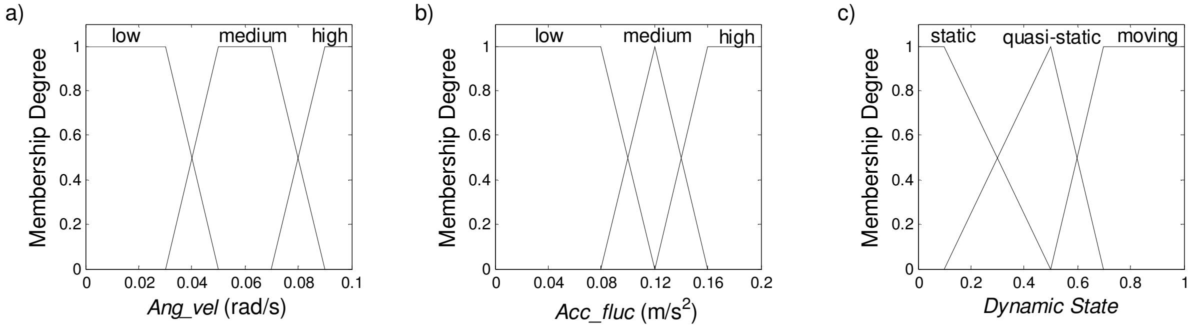


Figure 5-4: Membership functions of the fuzzy expert system for the tilt angle correction algorithm.

Table 5-1: Fuzzy rules to identify the stationary state of the tool.

Fuzzy rules	*Ang_vel*	*Acc_fluc*	Dynamic state
1	Low	Low	Static
2	Low	Med	Quasi-static
3	Med	Low	Quasi-static
4	Others		Moving

When the derivative of the position sensor is zero for 0.1 second and the *dynamic state* is less than 0.5 for 0.1 second, the tool is considered stationary. The 0.1 second period is chosen because it is not realistic for an operator to move a tool so that the *dynamic state* is less than 0.5 and the derivative of the encoder-based position sensor output is zero for 0.1 second unless the tool is stationary. Also, the 0.1 second period is used to calculate the average accelerations in each axis. According to the above assumption, the fuzzy expert rules for the stationary state identification of a tool are shown in Table 5-2 (Rule 5-1 and Rule 5-2). When the system concludes that the tool is stationary, the tilt angles are corrected by using the method in Subsection 4.4.2. To prevent a further orientation drift, the angular velocity is set to zero when the IMU is stationary (Rule 5-3 in Table 5-2).

5.3.2 Fastening Action Detection

To identify the fastening action, the acceleration signature is studied and distinguished from other possible movements that the fastening tool can experience (Figure 5-5). The identification of the fastening action by using the frequency domain is difficult because the frequency contents of the fastening action depend on many variables such as materials and the shape of the workpiece. Instead, the time domain is used for the analysis of the fastening action.

The acceleration signature of the fastening action (Figure 5-5 a)) denotes a very high vibration and high magnitude of acceleration. Figure 5-5 b) shows the base excitation when the tool is free-running in the air without fastening a bolt. The acceleration of the base excitation has a high frequency of vibration, but it does not have a high magnitude of acceleration. Figure 5-5 c) depicts the situation when the tool is manually shaken by a person. It displays high magnitudes of acceleration but does not exhibit a high frequency of vibration. Figure 5-5 d) shows normal movements when the tool is moved around. This movement does not have a high frequency of vibration or high peaks of acceleration. This study demonstrates that fastening action can be identified with acceleration frequency contents and the magnitudes of acceleration. Let the magnitude of acceleration measurement (*Acc_mea*) be

$$Acc\_mea = (A_x^{\ 2} + A_y^{\ 2} + A_z^{\ 2})^{0.5}. \tag{5-15}$$

If at least three peaks satisfy *Acc_mea* >20 m/sec^2 within a 0.1 second interval, the expert system concludes that the tool is fastening a bolt. In order to measure the 0.1 second interval, a time variable, *fastening_period*, is initialized when the accelerometers first detect *Acc_mea* >20 m/sec^2, and increases as the time elapses. If the system does not detect three peaks that satisfy *Acc_mea* >20 m/sec^2 within 0.1 second interval, it is initialized again when the next peak is detected. The rules for the fastening action detection are shown in Table 5-2 (Rule 5-4 to Rule 5-7). The specific values of the expert system conditions such as the peak values of *Acc_mea* may vary depending on the shapes and the materials of the bolt and the workpiece.

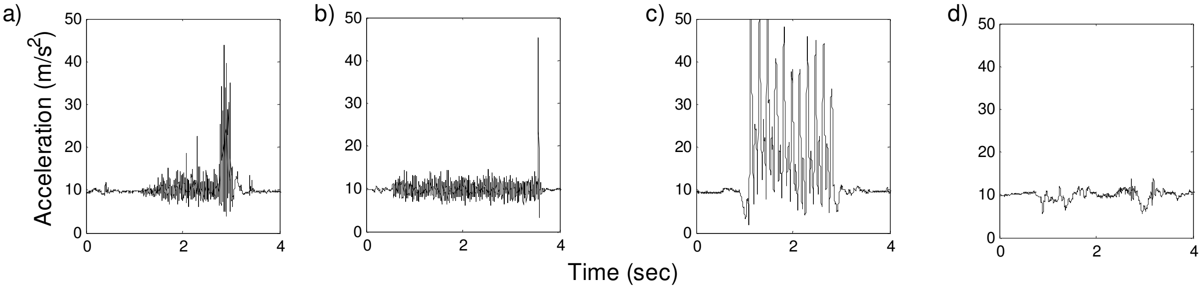


Figure 5-5: Possible acceleration measurement signatures of the fastening tool used in different scenarios: a) fastening action, b) base excitation, c) hand vibration, and d) normal movements.

5.3.3 Fastened Bolt Identification

When the fastening action is detected, the tool tracking system must identify the fastened bolt. The fastened bolt identification process is summarized in Figure 5-6. There are three scenarios for the bolt identification:

1. The position sensor alone identifies the fastened bolt.
2. When Scenario 1 fails, the IMU information is added to verify the fastened bolt.
3. When Scenario 2 fails, the system assumes that the bolt with the highest possibility of being fastened is fastened and waits for the next fastening action. When the system identifies the next fastened bolt, it identifies the previously fastened bolt.

When the system successfully identified the fastened bolt, it corrects the orientation error and outputs the fastened bolt number.

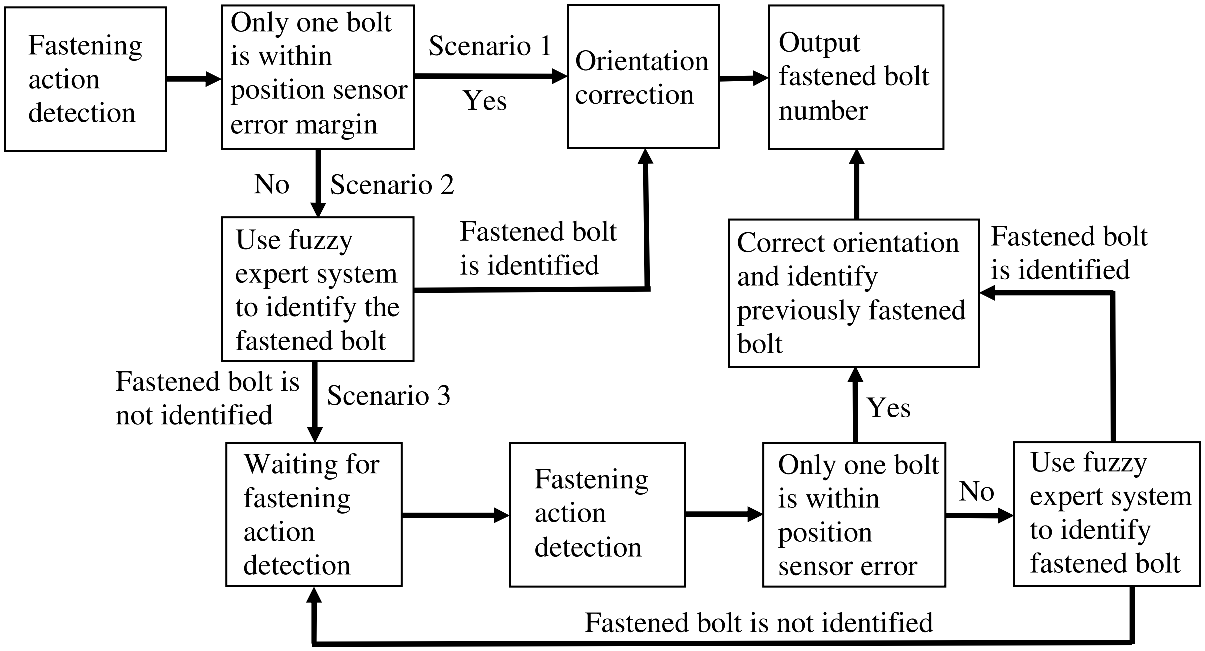

Fastening action detection
Only one bolt is within position sensor error margin
Scenario 1
Yes
Orientation correction
Output fastened bolt number
No
Scenario 2
Use fuzzy expert system to identify the fastened bolt
Fastened bolt is identified
Correct orientation and identify previously fastened bolt
Fastened bolt is identified
Fastened bolt is not identified
Scenario 3
Yes
Waiting for fastening action detection
Fastening action detection
Only one bolt is within position sensor error
No
Use fuzzy expert system to identify fastened bolt
Fastened bolt is not identified

Figure 5-6: Fastened bolt identification process.

When the tool fastens a bolt, the location of the tool tip coincides with one of the bolts (Figure 5-7). Then, the distance between the position sensor and one of the bolt position should be $(L_x{}^2 + L_y{}^2 + L_z{}^2)^{0.5}$. However, the position sensor has error, and the location of the bolt also has error due to manufacturing uncertainties. When the tool fastens a bolt, the position sensor error with respect to the nth bolt is

$$PSE_n = ((n_x - P_x)^2 + (n_y - P_y)^2 + (n_z - P_z)^2)^{0.5} - (L_x{}^2 + L_y{}^2 + L_z{}^2)^{0.5} . \quad (5\text{-}16)$$

PSE_n includes the error of the position sensor as well as the manufacturing error of the workpiece. Thus, when the tool fastens a bolt, the following inequality holds for at least one bolt:

$$PSE_n \le MPSE + MMU . \quad (5\text{-}17)$$

If there is only one bolt that satisfies (5-17), the system concludes that the tool fastens that bolt. In this application, the maximum position sensor error is ±7mm.

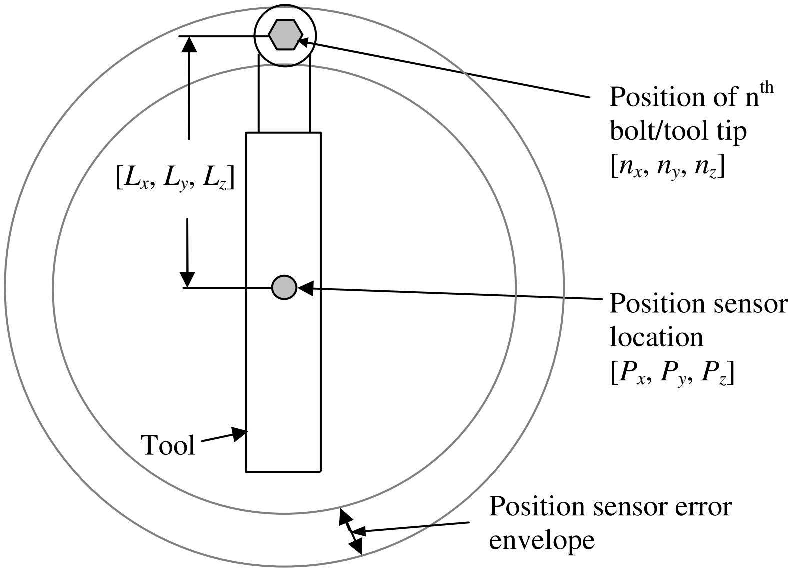

Position of nth bolt/tool tip $[n_x, n_y, n_z]$
$[L_x, L_y, L_z]$
Position sensor location $[P_x, P_y, P_z]$
Tool
Position sensor error envelope

Figure 5-7: Position sensor error envelope while a tool fastens a bolt. The position of the tool tip coincides with one of the bolts.

It is possible that more than one bolt satisfies (5-17). In this case, the tool tip position calculation by using (5-1) is utilized to identify the fastened bolt. However, identifying the fastened bolt does not simply correspond to finding the closest bolt from the calculated tool tip position because the orientation error is unknown. To identify the fastened bolt in the presence of orientation uncertainties, a fuzzy expert system is utilized.

Two inputs are employed to identify the fastened bolt: the calculated tool tip position error and the run-time which is the duration from the previous complete orientation correction to the current time excluding the stationary state period. As the calculated tool tip position error with respect to the bolt decrease, the probability of fastening the bolt gets higher. The calculated tool tip position error is represented as

$$PE_n = ((T_x - n_x)^2 + (T_y - n_y)^2 + (T_z - n_z)^2)^{0.5} . \quad (5\text{-}18)$$

As the run-time increases, the reliability of the calculated tool tip position decreases. Both position error and run-time are modeled, and the membership degrees with respect to the *n*th bolt are shown in Figure 5-8. The two antecedent conditions are aggregated with the min operation [118], [119] to obtain the implication on the consequence membership function. Figure 5-8 c) shows the membership degree output for the *n*th bolt. The bolt with the highest membership degree of Bolt_Output_1 is chosen as the fastened bolt because it has the highest probability of being fastened. When the membership degree of the calculated tool tip position error is lower than that of run-time, the closest bolt from the calculated tool tip position is the fastened bolt as shown in Figure 5-9.

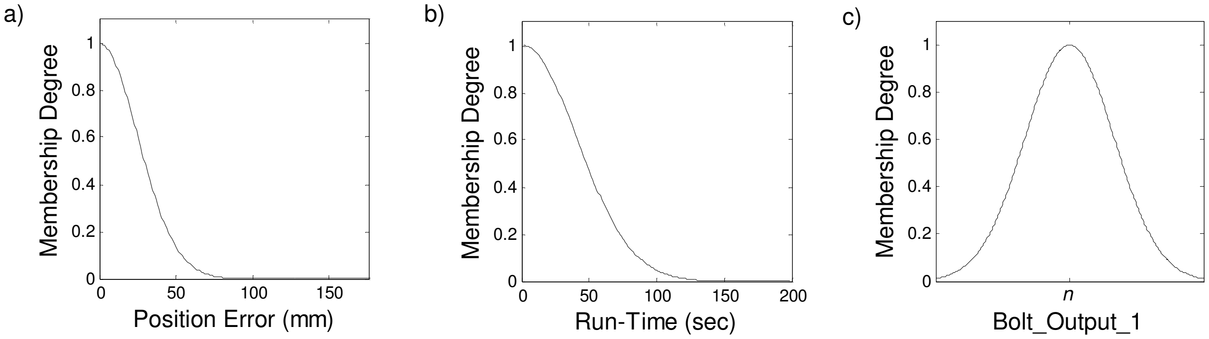


Figure 5-8: Membership degree functions of a) calculated tool tip position error, b) run-time, and c) the output for Bolt *n*.

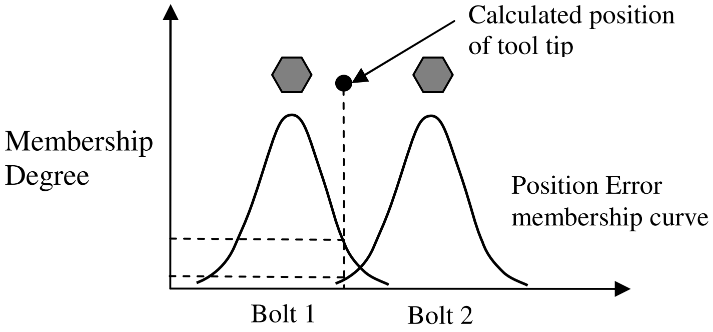

Calculated position of tool tip
Membership Degree
Position Error membership curve
Bolt 1
Bolt 2

Figure 5-9: Fastened bolt identification when position error is the determining factor.

However, if run-time becomes the determining factor, multiple bolts will have the highest membership degree in Bolt_Output_1 fuzzy set in Figure 5-8 (c). If all the bolts are fastened except for one, the system concludes that the unfastened bolt is now being fastened. If more than one bolt is not fastened, the system cannot identify the fastened bolt. Then, the system uses another fuzzy expert system to identify the bolt with the highest probability of being fastened using the position error and position sensor error as shown in Figure 5-10. The two antecedent conditions are multiplied to obtain the membership degree of Bolt_Output_2 for the n^{th} bolt and the bolt with a higher membership degree is assumed fastened. In this scenario, the fastening system also examines the next fastened bolt to ensure that the output of the fuzzy expert system is the correct fastened bolt. When the next fastening action is detected, the system identifies the fastened bolt and finds the path from the fastened bolt to the previous fastened bolt with respect to the previous possible bolt position and orientation. By comparing the distances and the orientations between the possible previously fastened bolts and the currently fastened bolt, the system can identify the previously fastened bolt. The expert rules for the fastened bolt identification phase are shown in Table 5-2 (Rule 5-8 to Rule 5-11).

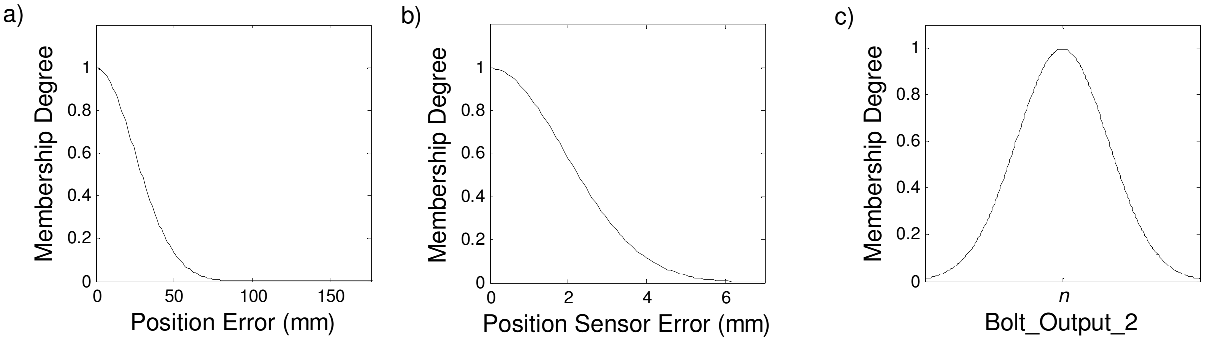


Figure 5-10: Membership degree functions of a) calculated tool tip position error, b) position sensor error, and c) output for Bolt *n*.

5.3.4 Orientation Correction

When a tool fastens a bolt, the orientation of the bolt and the vector that connects the position sensor and the tool tip position, which coincide with the bolt position, are known. Since two vectors are known, the complete orientation of the tool can be determined. The tool frame is chosen so that the z-axis is parallel to the socket, which is almost parallel to the fastened bolt when the tool fastens a bolt. This axis gives the z-axis vector of the direction cosine matrix: c_{zX} , c_{zY} , and c_{zZ} . If the tool frame is selected so that L_y is zero, the location of the tool tip with respect to the centre of mass in the tool frame becomes [L_x 0 L_z]. From (5-1), the x-axis vector of the direction cosine matrix becomes

$$\begin{aligned} c_{xX} &= \frac{n_x - c_{zX}L_z - P_x}{L_x'} \\ c_{xY} &= \frac{n_y - c_{zY}L_z - P_y}{L_x'} \\ c_{xZ} &= \frac{n_z - c_{zZ}L_z - P_z}{L_x'} \end{aligned} , \tag{5-19}$$

where $L_x' = (n_x - c_{zX}L_z - P_x)^2 + (n_y - c_{zY}L_z - P_y)^2 + (n_z - c_{zZ}L_z - P_z)^2$. L_x' is used to normalize the x-axis vector of the direction cosine matrix instead of L_x. This is attributed to the fact that L_x may not match L_x' due to position sensor error (PSE_n). The y-axis vector of the rotation matrix can be calculated from the cross-product of the two vectors of the rotation matrix. When all three vectors of the direction cosine matrix are determined, three tool frame vectors of the rotation matrix must be normalized. Then, four quaternion terms can be determined from the rotation matrix. The four quaternion terms should be normalized to satisfy (4-7).

Table 5-2: Rules for the fastening tool tracking system in linguistic terms.

Stationary State Identification	
Rule 5-1	IF the derivative of the position sensor output is zero for the last 0.1 second AND *dynamic state* < 0.5 for the last 0.1 second, THEN update the average accelerations AND the tool is stationary.
Rule 5-2	IF the tool was stationary in the previous time step AND the derivative of the position sensor output is zero AND *dynamic state* < 0.5, THEN update the average accelerations AND the tool is stationary.
Rule 5-3	IF the tool is stationary, THEN $\omega_x = \omega_y = \omega_z = 0$ AND correct the quaternion terms using (4-23) to (4-26)
Fastening Action Detection	
Rule 5-4	IF *Acc_mea* > 20 m/s^2 AND *fastening_period* > 0.1 second THEN *fastening_period* = 0 AND peak = 0 AND peak_high = 1.
Rule 5-5	IF *Acc_mea* > 20 m/s^2, THEN peak_high = 1

Rule 5-6	IF $Acc\_mea < 20$ m/s^2 AND peak_high = 1 THEN peak = peak +1 AND peak_high = 0
Rule 5-7	IF peak = 3 AND *fastening_period* < 0.1 second, THEN the tool is fastening a bolt.
Fastened Bolt Identification	
Rule 5-8	IF the tool is fastening a bolt AND only one bolt that satisfies $PSE_n \leq MPS + MME$, THEN the tool is fastening the nth bolt.
Rule 5-9	IF the tool is fastening a bolt AND there are more than one bolt that satisfy $PSE_n \leq MPSE + MME$, THEN the bolt with the maximum Bolt_Output_1 membership value is being fastened.
Rule 5-10	IF the tool is fastening a bolt AND there is more than one bolt that has the highest Bolt_Output_1 membership value AND all the bolts were fastened except one bolt, THEN the tool is fastening the unfastened bolt.
Rule 5-11	IF the tool is fastening a bolt AND more than one bolt have the highest Bolt_Output_1 membership value AND more than one bolt is not fastened before, THEN assume that the bolt with the maximum Bolt_Output_2 membership value is being fastened. IF the next fastening action is detected, THEN identify the fastened bolt AND identify the previously fastened bolt.
Rule 5-12	IF the fastened bolt is identified THEN correct the quaternion terms.

5.4 Experiments

The fastening tool tracking system is attached to a right angle tool and examined on the testbed shown in Figure 5-11. Eight bolts are placed on the testbed, and their positions are shown in Table 5-3. When the tool was positioned on Bolt 1, 4, 5, 6, and 7, the bolts were fastened, but when the tool was on Bolt 2, 3, and 8, the tool ran in the air right above the bolts to test if the system can differentiate the two different scenarios. The tool started from the initial position and moved for 204 seconds before it was returned to the initial position. The tool moved from Bolt 1 to Bolt 8 in sequence, and after Bolt 6 was fastened, the tool was left stationary for 80 seconds to test the stationary state identification fuzzy expert system. The test results of tracking the tool tip are illustrated in Figure 5-12.

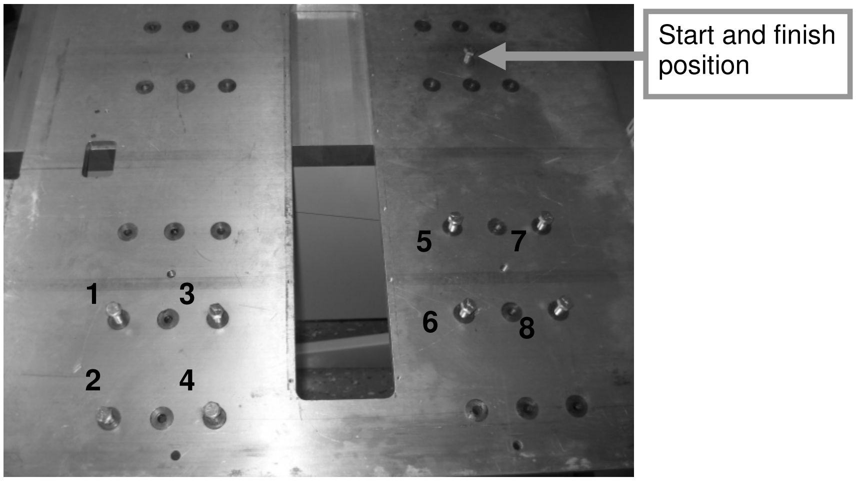


Figure 5-11: Testbed for the lab experiment.

Table 5-3: Bolt positions shown in Figure 5-11.

Bolt number	Bolt position	Bolt number	Bolt position
Bolt 1	(-0.4. -0.34, 0)	Bolt 5	(-0.05, -0.24, 0)
Bolt 2	(-0.4. -0.44, 0)	Bolt 6	(-0.05, -0.34, 0)
Bolt 3	(-0.3. -0.34, 0)	Bolt 7	(0.05, -0.24, 0)
Bolt 4	(-0.3. -0.44, 0)	Bolt 8	(0.05, -0.34, 0)

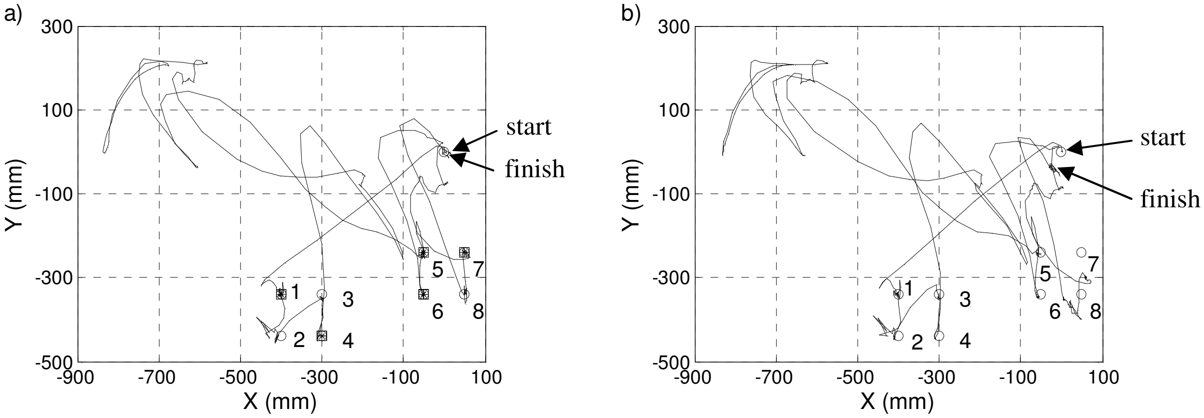


Figure 5-12: Tool tracking results a) with the intelligent system and b) without the intelligent system.

Figure 5-12 a) shows the trajectory of the calculated tool tip position with the intelligent system which includes the KFs and the fuzzy expert system. Figure 5-12 b) depicts the trajectory of the calculated position of the tool tip without the intelligent system. When the system detects the fastening action, a square mark with * symbol is added to the fastened bolt position. Figure 5-12 a) shows that Bolt 1, 4, 5, 6, and 7 are marked; indicating that the system successfully identified the fastened bolt. The start and finish points are almost identical when the intelligent system is used. However, Figure 5-12 b) reveals that the finish point is different from the start point when the tool tip position is calculated without the expert system. The calculated tool tip position errors with and without the intelligent system are summarized in Table 5-4 (with respect to the fastened bolt positions and the finish position). The total position errors of Table 5-4 are calculated by using (5-18). The position error without the intelligent system increases over time because the orientation error drifts over time. However, the proposed intelligent system reduces the position error because the orientation error is corrected when the tool fastens a bolt and when the tool is held stationary.

Table 5-4: Position error comparison between with and without the intelligent system.

Tool position	Time (s)	Position error with the intelligent system (mm)				Position error without the intelligent system (mm)			
		X-axis	Y-axis	Z-axis	Total	X-axis	Y-axis	Z-axis	Total
Bolt 1	19	-2	0	6	6	-2	1	6	6
Bolt 4	43	-12	-6	-6	15	-10	-4	9	14
Bolt 5	85	-2	-8	2	8	-9	-5	37	38
Bolt 6	89	-4	-2	2	5	-8	-7	38	39
Bolt 7	184	3	-6	7	10	8	-58	28	65
Finish	204	-1	-6	0	6	-27	-33	83	93

5.5 Conclusion

This chapter presented an intelligent tool tracking system that utilizes a hybrid sensor configuration consisting of an IMU and a position sensor. KFs were developed to estimate the orientation and the position of the tool more accurately. The outputs of the sensors are related to identify if the tool is stationary or fastening a bolt. When the tool is stationary, the system corrects the tilt angles. When the tool fastens a bolt, the system identifies the fastened bolt and corrects the orientation error. The intelligent system was validated through experiments.

The fastening tool tracking system was tested with a manufacturing assembly example in a laboratory setting. The position error of the tool tip increases as the time of operation elapses when the intelligent system is not used because the orientation error increases over time. However, by utilizing the intelligent system, the fastened bolts are correctly identified, and the position error is reduced.

Chapter 6
Fastening Tool Tracking System Using a Combined Kalman/Particle Filter

This chapter proposes a novel position/orientation estimation technique. The proposed method combines a KF and a PF to hybridize one IMU and one position sensor. The KF is used to estimate the position, and the PF is used to estimate the orientation. Then, this method is revised to apply to the fastening tool tracking system to identify the fastened bolts. The test results of the proposed system in this chapter are compared with those of the KF-based intelligent system that was introduced in Chapter 5.

6.1 Overview of the Position/Orientation Tracking System Combining the KF and the PF

The proposed position/orientation estimation method is presented in Figure 6-1. The proposed method calculates the position and orientation states from IMU measurements and measures the position using a position sensor. An expert system is developed to correct the angular velocities according to the measurements from the IMU and the position sensor. By using the corrected angular velocities (ω_x, ω_y, and ω_z), the rotation matrix from the body frame to the fixed frame is estimated. The rotation matrix of the i^{th} particle from the body frame to the fixed frame at time t_k (${}_b^f C_k^i$) is represented by using quaternions as:

$$
\begin{aligned}
{}_b^f C_k^i &= \begin{bmatrix} c1_k^i & c2_k^i & c3_k^i \\ c4_k^i & c5_k^i & c6_k^i \\ c7_k^i & c8_k^i & c9_k^i \end{bmatrix} \\
&= \begin{bmatrix} {q0_k^i}^2 + {q1_k^i}^2 - {q2_k^i}^2 - {q3_k^i}^2 & 2(q1_k^i q2_k^i - q0_k^i q3_k^i) & 2(q1_k^i q3_k^i + q0_k^i q2_k^i) \\ 2(q1_k^i q2_k^i + q0_k^i q3_k^i) & {q0_k^i}^2 - {q1_k^i}^2 + {q2_k^i}^2 - {q3_k^i}^2 & 2(q2_k^i q3_k^i - q0_k^i q1_k^i) \\ 2(q1_k^i q3_k^i - q0_k^i q2_k^i) & 2(q2_k^i q3_k^i + q0_k^i q1_k^i) & {q0_k^i}^2 - {q1_k^i}^2 - {q2_k^i}^2 + {q3_k^i}^2 \end{bmatrix},
\end{aligned}
\qquad (6\text{-}1)
$$

where $c\#_k^i$ is the component of the direction cosine matrix of the i^{th} particle at time t_k, and $q0_k^i$, $q1_k^i$, $q2_k^i$, and $q3_k^i$ are the four components of a unit quaternion. Since the orientation is calculated using a unit quaternion, each orientation particle consists of four states ($x^i_{PF,k} = [q0^i_k \quad q1^i_k \quad q2^i_k \quad q3^i_k]$) that satisfy

$$q0^{i\,2}_k + q1^{i\,2}_k + q2^{i\,2}_k + q3^{i\,2}_k = 1\,. \tag{6-2}$$

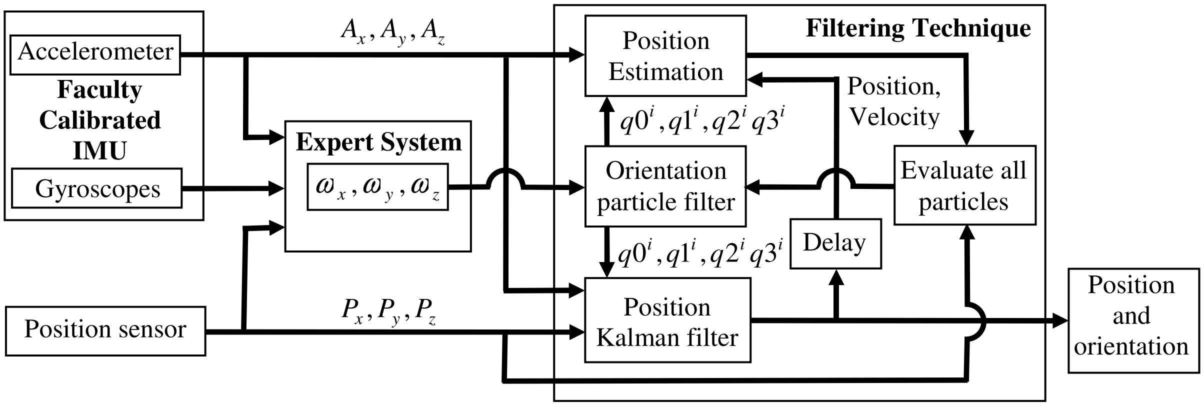


Figure 6-1: Outline of the proposed method.

The quaternion components of each particle can be computed from the angular velocity measurements using:

$$\begin{bmatrix} q0^i_{k+1} \\ q1^i_{k+1} \\ q2^i_{k+1} \\ q3^i_{k+1} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 2 & -\omega_{x,k} \cdot t & -\omega_{y,k} \cdot t & -\omega_{z,k} \cdot t \\ \omega_{x,k} \cdot t & 2 & \omega_{z,k} \cdot t & -\omega_{y,k} \cdot t \\ \omega_{y,k} \cdot t & -\omega_{z,k} \cdot t & 2 & \omega_{x,k} \cdot t \\ \omega_{z,k} \cdot t & \omega_{y,k} \cdot t & -\omega_{x,k} \cdot t & 2 \end{bmatrix} \cdot \begin{bmatrix} q0^i_k \\ q1^i_k \\ q2^i_k \\ q3^i_k \end{bmatrix}. \tag{6-3}$$

The corresponding velocity and position of each orientation particle are estimated using a KF. From (4-14) and (4-17), the motion tracking equations of particle *i* for short distance navigation can be written as:

$${}^f\dot{V}^i = {}^f_bC^i \cdot {}^bA + {}^f g_l\,, \tag{6-4}$$

$${}^f\dot{P}^i = \int {}^fV^i dt\,, \tag{6-5}$$

where ${}^fV^i$ represents the velocity of the i^{th} particle in the fixed frame. Eq. (6-4) reveals that the acceleration measurements in the body frame (${}^bA = [A_x \quad A_y \quad A_z]^T$), which includes the gravity vector, are multiplied by the rotation matrix to calculate the acceleration in the fixed frame. Thus, when the rotation matrix is inaccurate, the misalignment between the true body frame and the calculated body frame leads to an acceleration error in (6-4), which consequently results in velocity and position errors. In other words, a high orientation error can result in a high position error. The proposed filter evaluates orientation particles by comparing the position calculation of each particle with the position estimate of each particle from the KF. The particles with lower position difference between the two are assigned higher weights.

6.2 Position Kalman Filter

A KF is used to estimate the position of each particle in the proposed method because the state space model of an IMU is linear and the noise distribution of many position sensors can be considered zero-mean Gaussian. To utilize the KF, the motion equations in (6-4) and (6-5) must be rewritten in the form of (2-5). The state vector of the KF of the i^{th} particle at time t_k is defined as

$$x^i_{KF,k} = \begin{bmatrix} P^i_{kx,k} & V^i_{kx,k} & A^i_{kx,k} & P^i_{ky,k} & V^i_{ky,k} & A^i_{ky,k} & P^i_{kz,k} & V^i_{kz,k} & A^i_{kz,k} \end{bmatrix}^T, \tag{6-6}$$

where $P^i_{k-axis,k}$, $V^i_{k-axis,k}$, $A^i_{k-axis,k}$ are the position, velocity, and acceleration of the i^{th} particle in each axis using the KF respectively. Then, the system transition matrix of the i^{th} particle, Φ^i_k , is

$$\Phi^i_k = \begin{bmatrix} 1 & t & c1^i_k \cdot t^2/2 & 0 & 0 & c2^i_k \cdot t^2/2 & 0 & 0 & c3^i_k \cdot t^2/2 \\ 0 & 1 & c1^i_k \cdot t & 0 & 0 & c2^i_k \cdot t & 0 & 0 & c3^i_k \cdot t \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c4^i_k \cdot t^2/2 & 1 & t & c5^i_k \cdot t^2/2 & 0 & 0 & c6^i_k \cdot t^2/2 \\ 0 & 0 & c4^i_k \cdot t & 0 & 1 & c5^i_k \cdot t & 0 & 0 & c6^i_k \cdot t \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & c7^i_k \cdot t^2/2 & 0 & t & c8^i_k \cdot t^2/2 & 1 & t & c9^i_k \cdot t^2/2 \\ 0 & 0 & c7^i_k \cdot t & 0 & 0 & c8^i_k \cdot t & 0 & 1 & c9^i_k \cdot t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6-7}$$

Gravity is treated as a deterministic input. When the direction of the Z-axis of the fixed frame is chosen opposite to the local downwards, the velocity and the position changes due to gravity measurements must be compensated for along this direction. Since the system input matrix is not related to the orientation particle, it is written as:

$$\Gamma_k \cdot u_{k-1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & |g_l| \cdot t^2/2 & |g_l| \cdot t & 0 \end{bmatrix}^T. \tag{6-8}$$

Since the magnitude of the local gravity vector can be assumed constant for short distance navigation, the system input matrix is assumed to be constant.

The acceleration and the position components are directly measured using calibrated accelerometers and a position sensor. Then, the measurement matrix is:

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6-9}$$

6.3 Orientation Filtering Technique

The proposed method uses the PF to estimate the orientation of an object. The weight of each particle is determined by using the position calculation of each orientation particle and the position measurements as the most probable value. In such an approach, however, the differences between the position measurements and the position calculation of each orientation state can be a result of not only the orientation state but also sensor errors such as noise. Then, the particles with higher orientation errors can have higher likelihoods of being in the correct orientation. In other words, the position measurements at time t_k are not accurate enough to approximate the true posterior of orientation.

To overcome this problem, the summation of the position differences for a period of time (ΔT_s, where subscript s is the s^{th} orientation iteration, s = 1, 2, …) is selected to determine the weights of the particles instead of the instantaneous position differences at time t_k. In addition, instead of using the direct position measurements, the KF position estimations of each particle, which incorporates the position measurements, are used to reduce the effect of the measurement noise. Then, the likelihood is calculated based on the accumulated position difference between the estimations and the calculated values of the i^{th} particle (APE_s^i) as

$$APE_s^i = \sum_{k=(s-1)\cdot M_S+1}^{M_S \cdot s} \{(P_{px,k}^i - P_{kx,k}^i)^2 + (P_{py,k}^i - P_{ky,k}^i)^2 + (P_{pz,k}^i - P_{kz,k}^i)^2\}, \tag{6-10}$$

where $M_s = \Delta T_s / t$, and $P^i_{p-axis,k}$ are the position states of the i^{th} orientation particle at time t_k. The lower APE^i_s of the given particle signifies the higher likelihood of being the correct orientation. For every ΔT_s period elapses, the weight of each particle is recalculated according to APE^i_s values. Then, the PF must be modified to use APE^i_s instead of the direct position measurements. The posterior approximation up to time t_s is

$$p(x_{0:s} \mid APE^i_{1:s}) \approx \sum_{t=1}^{N} w^i_k \delta(x_{0:s} - x^i_{0:s}) . \tag{6-11}$$

Then, the posterior up to time t_s is expressed as

$$p(x_{0:s} \mid APE^i_{1:s}) \propto p(APE^i_s \mid x_s) \cdot p(x_s \mid x_{s-1}) \cdot p(x_{0:s-1} \mid APE^i_{1:s-1}) , \tag{6-12}$$

and the normalized weight of the i^{th} particle at the s^{th} orientation iteration becomes

$$w^i_s \propto \frac{p(APE^i_s \mid x^i_s) \cdot p(x^i_s \mid x^i_{s-1})}{r(x^i_s \mid x^i_{s-1}, APE^i_s)} \cdot w^i_{s-1} . \tag{6-13}$$

By resampling and choosing the importance density from prior, $p(x_s \mid x_{s-1})$ and the fact that $w^i_{s-1} = 1/N$, the normalized weight has the following relationship:

$$w^i_s \propto p(APE^i_s \mid x^i_{PF,s}) . \tag{6-14}$$

The weight of the orientation particle is calculated based on APE^i_s and the most probable value of APE^i_s. argmin(APE^i_s) is chosen as the most probable value because the orientation with the minimum accumulated position error has the highest likelihood of being the correct orientation. Then, the normalized weight is calculated as:

$$w^i_s \propto \exp\left(\frac{-(APE^i_s - \arg\min(APE^i_s))^2}{2 \times (\sigma(APE^i_s))^2} \right) , \tag{6-15}$$

where $\sigma(APE^i_s)$ is the standard deviation of APE^i_s. Since $3\sigma(APE^i_s)$ contains 99.7% of the values of APE^i_s, it is valid to assume that the mean of APE^i_s less argmin(APE^i_s) is $3\sigma(APE^i_s)$. After resampling, the quaternion terms must be normalized to satisfy (6-2). The summary of the proposed particle filtering technique is described in Table 6-1.

Table 6-1: Summary of the proposed particle filtering technique.

```
If k = 1, Draw $x_0^i$ from $p(x_0)$ and $w_0^i = 1/N$          // Initialization
Calculate $x_{PF,k}^i$                                          // Prediction
Calculate $APE_s^i$ according to (6-10)                         // Likelihood
IF remain $(k/M_s) = 0$                                         // Update
    $x_s^i = x_{PF,k}^i$                                        // Predicted states
    Calculate $w*_s^i = \exp\left(\frac{-(PE_s^i - \arg\min(APE_s^i))^2}{2\times(\sigma(APE_s^i))^2}\right)$   // Weight calculation
    $w_s^i = w*_s^i / \sum_{i=1}^{N} w*_s^i$                    // Normalize weights
    Draw $x_s^i$ based on $w_s^i$                               // Draw the next particles (Resampling)
    $w_s^i = 1/N$                                               // Reset weights
    $x_{PF,k}^i = x_s^i$
End IF
```

Note that ΔT_s should be chosen based on the sensor accuracy. If ΔT_s is too short, the accumulated position difference may not be large enough to help identify the best orientation particle. When ΔT_s is too long, the orientation error may become large prior to the application of the orientation evaluation. This, in turn, impacts the orientation estimation accuracy. In addition, this algorithm requires more particles to represent the PDF to cover a wider range of possible orientation angles leading a higher computational cost.

6.3.1 Angular Velocity Correction Using an Expert System

The expert system is used in two cases: (i) estimating the initial gyro biases and (ii) detecting the stationary state. In Chapter 4, an accelerometer calibration method with a simple procedure was introduced. Although it provides a high accuracy, the procedure takes about thirty seconds and the gyro calibration technique in Appendix B takes about 30 seconds. When the proposed KF-PF combined method in this chapter is used, simpler calibration procedure can be employed.

Both gyros and accelerometers suffer from biases and gain errors primarily caused by temperature drift. Thus, many commercial IMUs are factory calibrated and have a temperature drift compensation feature to reduce the effects of biases and gain errors, as well as misalignment and non-linearity. Also, acceleration sensitivities of the gyros are compensated for. Even when an IMU has a temperature compensation feature, the gains and biases slightly differ whenever the sensor is switched on as discussed in Chapter 2. However, the effects of the accelerometer gain and bias errors due to power-on can be insignificant in this application because the gain and bias differences due to power-on are usually very small and a position sensor is used to correct the velocity and position estimation errors.

The gain difference of a gyro due to power-on is small, and the orientation error caused by the gain error due to power-on is not significantly high. However, the gyro bias error can cause a significant orientation error because gyro bias leads to continuous orientation drift over time and there are no orientation measurements to correct the orientation. As the angular velocity bias increases, the proposed filter will require a higher process noise and more particles to compensate for the orientation uncertainty in order to maintain the orientation accuracy. Since the bias variation due to power-on is also very small, a slight increment in the numbers of particles and process noise can remove the effect of the small bias error. However, this can still result in higher computational complexity. Since factory calibrated IMUs have the acceleration sensitivity compensation for gyros, the gyro biases can be calculated by simply leaving the object stationary. Then, the mean value for the stationary state of each gyro becomes the bias. Since this procedure is very simple and fast, the gyro bias is calibrated in the proposed system. The IMU is left stationary for 1 s, and the mean values are subtracted from gyro measurements to compensate for the biases. The rules for the gyro initial bias compensation are defined as follows:

Rule 6-1: IF time $\leq$ 1 second THEN

$$\omega\_sum_{axis} = \omega\_sum_{axis} + \omega\_meas_{axis} \text{ AND } \omega_{axis} = 0$$

Rule 6-2: IF time = 1 second THEN

$$\omega\_bias_{axis} = \omega\_sum_{axis} \times t$$

Rule 6-3: IF time > 1 second THEN

$$\omega_{axis} = \omega\_meas_{axis} - \omega\_bias_{axis}$$

where ω_sum_{axis} is the angular velocity summation of each axis, ω_bias_{axis} is the calculated angular velocity bias of each axis, and ω_meas_{axis} is the angular velocity measurement of each axis.

It is expected to achieve higher position and orientation accuracy when the accelerometers and gyros are calibrated more accurately. However, when the proposed KF-PF approach is used with a factory calibrated IMU, the effect of calibration is not significant. Although the gyro biases are not significantly high, they are calculated and removed because the procedure is simple and takes only 1 second, and bias removal reduces the orientation error.

Another role of the expert system is to identify the stationary state. Bayesian estimation techniques are suitable for estimating dynamic states, but an alternative approach is required for estimating the states that do not change. Thus, when the object is stationary, the system should output the previous states instead of estimating the current state. When the object is identified as being stationary, the angular velocity is changed to zero. Even when the angular velocity is zero, the yaw angle can still drift during the resampling step. This is due to the fact that the yaw angle drift does not affect the position calculation when the object is stationary. Therefore, while the object is stationary, the resampling step is suspended to ensure the current orientation state is the same as the previous orientation state ($x^i_{PF,k} = x^i_{PF,k-1}$). To suspend the resampling step, ΔT_s is increased by one sample time whenever the object is identified as being stationary ($\Delta T_s = \Delta T_s + t$).

The stationary state is identified by using the similar approach to that in Chapter 4. Since a position sensor is available, the position measurement fluctuation (Pos_fluc_{axis}) of each axis is defined as

$$Pos\_fluc_{axis} = \left| P_{axix} - Avg\_P_{axis} \right|, \tag{6-16}$$

where the average position of each axis (Avg_P_{axis}) are expressed as

$$Avg\_P_{axis} = \sum_{j=1}^{n} P_{axis}(j) / w, . \tag{6-17}$$

When the object is stationary, the magnitude of the angular velocity of each axis should be lower than the maximum angular velocity error of the gyro, the acceleration fluctuation (Acc_fluc_{axis}) is smaller than the maximum acceleration noise from the accelerometer, and the fluctuation of the position measurement is less than the maximum error of the position sensor. Then, the expert rule for the stationary state is

Rule 6-4:

IF $|\omega_{axis}| \le \omega_{axis\_\max\_error}$ for the last $t_{stationary}$ seconds

AND $Acc\_fluc_{axis} < A_{axis\_\max\_noise}$ for the last $t_{stationary}$ seconds AND $Pos\_fluc_{axis} < P_{axis\_\max\_error}$ for the last $t_{stationary}$ seconds,

THEN $\omega_{axis} = 0$, and $\Delta T_s = \Delta T_s + t$

where $P_{axis\_\max\_error}$ is the maximum position error in each axis. The $t_{stationary}$ period should be chosen depending on the application.

6.3.2 Initial Orientation Estimation

In order to use the PF, the initial PDF of the orientation states need to be identified. Two extreme cases can be considered: (i) when the orientation state is almost completely known and (ii) when the orientation of the object is completely unknown. When the initial orientation of the object is known with a high level of certainty, the initial PDF can be represented as a normal distribution with a very small covariance.

However, in many circumstances, the initial orientation is completely unknown. Then, two different approaches are possible. First, quaternion states are drawn from a uniform distribution. In this case, a high initial process noise covariance is selected because the uncertainty of the orientation state is initially high. It is assumed that the orientation converges to the correct value as time elapses. Then, the uncertainty also reduces over time. In order to incorporate this knowledge, the process noise covariance monotonically reduces to the final value where the orientation is known with a high degree of certainty.

The second approach is to utilize the knowledge that the object is stationary for the first 1 second. Then, the average acceleration measurements for the first 1 second can be used to calculate the tilt angles so that the uncertainties in roll and pitch angles (θ_x and θ_y respectively) are reduced. The relationship between roll and pitch angles and the acceleration measurements due to gravity can be expressed as [116]:

$$\theta_x = \arctan(Avg\_A_{y\_i} / Avg\_A_{z\_i}),$$
$$\theta_y = \arcsin\left(\frac{-Avg\_A_{x\_i}}{(Avg\_A_{x\_i}{}^2 + Avg\_A_{y\_i}{}^2 + Avg\_A_{z\_i}{}^2)^{0.5}}\right), \tag{6-18}$$

where $Avg\_A_{axis\_i}$ is the initial average acceleration of each axis for the first one second using (4-17).

Although the roll and pitch angles can be determined from (6-16), the uncertainty in the yaw angle remains an issue. If any information about the yaw angle is available, it can be used to construct the initial PDF. However, if no information is available about the initial state of the yaw angle, a uniform distribution can be used to represent the PDF of the yaw angle. If 20 particles are used, the particles are placed 18° (360°/20 particles) apart in the yaw angle domain. Then, the initial quaternion states by using these angles are defined as follows [116]:

$$\begin{aligned}
q0_0^i &= \cos(\frac{\theta_x}{2})\cdot\cos(\frac{\theta_y}{2})\cdot\cos(\frac{\theta_z^i}{2}) + \sin(\frac{\theta_x}{2})\cdot\sin(\frac{\theta_y}{2})\cdot\sin(\frac{\theta_z^i}{2}) \\
q1_0^i &= \sin(\frac{\theta_x}{2})\cdot\cos(\frac{\theta_y}{2})\cdot\cos(\frac{\theta_z^i}{2}) - \cos(\frac{\theta_x}{2})\cdot\sin(\frac{\theta_y}{2})\cdot\sin(\frac{\theta_z^i}{2}) \\
q2_0^i &= \cos(\frac{\theta_x}{2})\cdot\sin(\frac{\theta_y}{2})\cdot\cos(\frac{\theta_z^i}{2}) + \sin(\frac{\theta_x}{2})\cdot\cos(\frac{\theta_y}{2})\cdot\sin(\frac{\theta_z^i}{2}) \\
q3_0^i &= \cos(\frac{\theta_x}{2})\cdot\cos(\frac{\theta_y}{2})\cdot\sin(\frac{\theta_z^i}{2}) + \sin(\frac{\theta_x}{2})\cdot\sin(\frac{\theta_y}{2})\cdot\cos(\frac{\theta_z^i}{2})
\end{aligned}, \tag{6-19}$$

where θ_z^i is the ith yaw angle at 1 second.

Similar to the first approach, a high initial process noise covariance is selected and monotonically reduced until it reaches the final value. However, since this algorithm already has the roll and pitch angle information, the initial process noise covariance should be selected lower than the level chosen for the first approach of case (ii).

6.4 Experiments

6.4.1 Preliminary Experiment

In order to test the novel PF-KF method described in this chapter, the presented theory is first tested in a lab environment with a factory calibrated IMU, and then applied to the tool tracking experiment.

The proposed method was experimentally tested by using a hybrid system that consists of one factory calibrated IMU (3DM-GX2, Microstrain) and one position sensor (Optotrak, NDI). Optotrak can locate infrared light emitting diode (IRED) position markers with submillimeter error [52], [53]. One IRED position marker was attached to the centre of the IMU. To check the orientation accuracy of the proposed method, the true orientation was calculated from four IRED position markers placed 280 mm apart from each other. The configuration of the sensors is depicted in Figure 6-2. Optotrak requires lines of sight between the IRED position markers and the cameras. Since an IRED position marker has a limited signal emitting angle, three sets of Optotrak cameras were used to track the position of the markers.

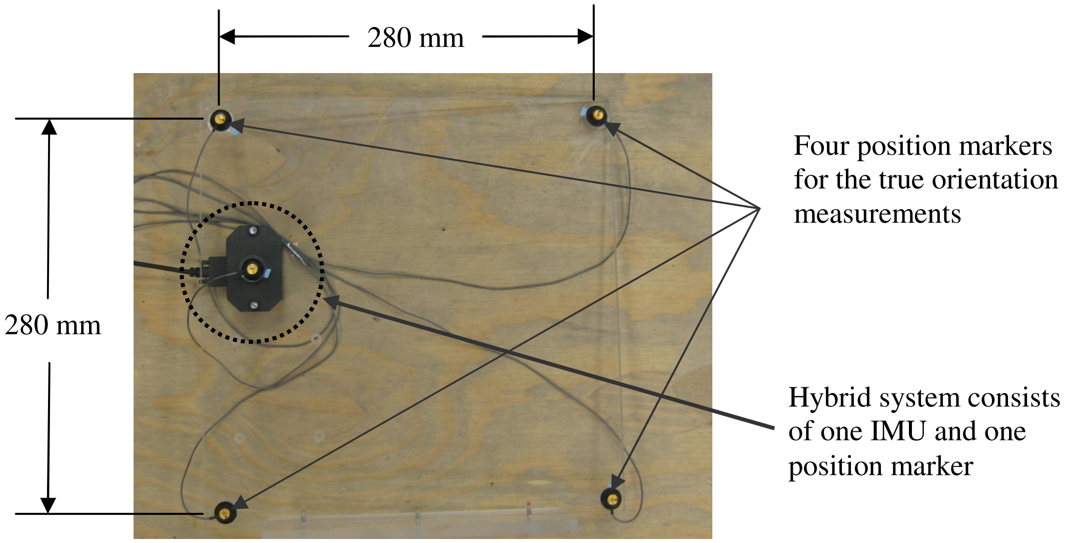

280 mm
280 mm
Four position markers for the true orientation measurements
Hybrid system consists of one IMU and one position marker

Figure 6-2: The proposed hybrid system and the true orientation measurement system.

In this experiment, the sensors shown in Figure 6-2 were moved manually in random 3D motions for 8 minutes and returned to the original position and orientation. The sensor was kept stationary for the first 1 second to remove the gyro bias and for the last 10 second to check if the expert system could detect the stationary state. The true position and orientation measurements using Optotrak are illustrated in Figure 6-3.

Both the position sensor and the IMU were sampled at 100 Hz (t = 0.01 s). Since it is unlikely to move an object and satisfy Rule 6-4 for 1 second when the object is manually moved, $t_{stationary}$ is set to 1 second. The proposed method evaluates the weights of the particles every 1 second interval while the tool is moving ($\Delta T_{1:s}$ = 1 s). When the system detects stationary state, ΔT can be extended as described in Rule 6-4. Four different analyses were conducted for the same test: (i) orientation error comparison between the EKF and the proposed filter, (ii) orientation error comparison of the proposed filters with different numbers of particles, (iii) orientation error comparison of the proposed filters with different numbers of particles when the initial condition is completely unknown, and (iv) position and orientation error comparisons when noise is added to the position measurements.

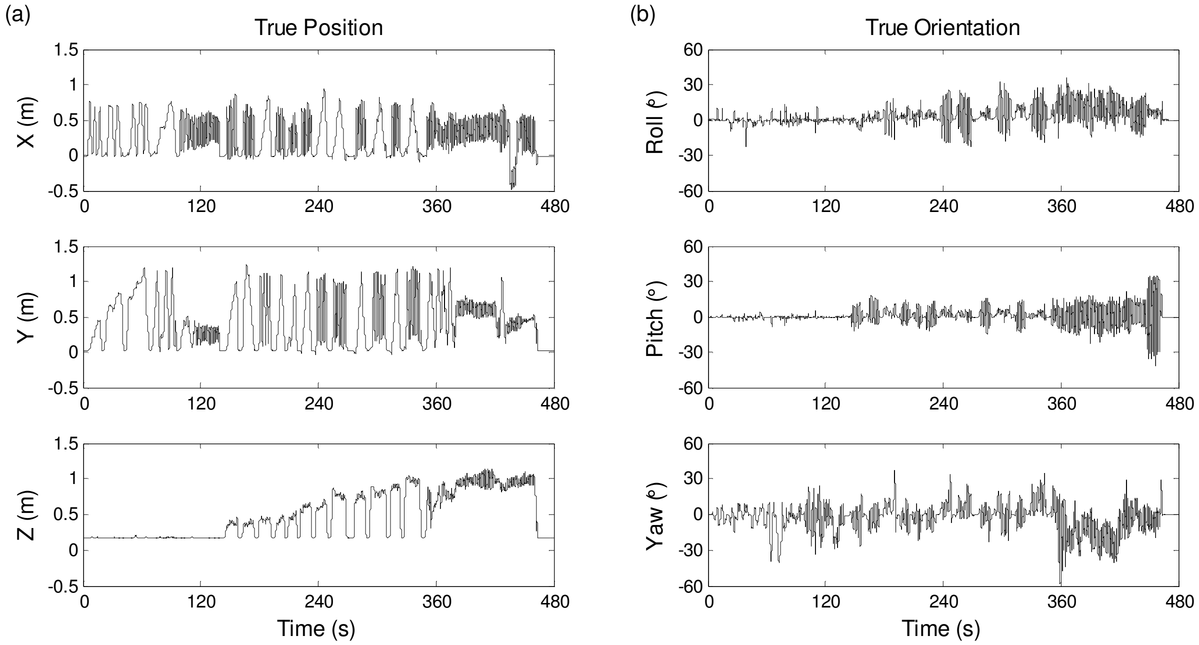


Figure 6-3: True (a) position and (b) orientation measurements using Optotrak.

First, the orientation errors obtained with both an EKF and the proposed filter with 20 particles are compared in Figure 6-4. The expert system described in Section 6.3 was applied to both the EKF and the proposed filter to correct the angular velocity. Figure 6-4 indicates that the orientation errors using the EKF increase over time, but the errors using the proposed filter are significantly reduced and do not grow over time. The roll and pitch angle errors of the proposed method have lower magnitudes than the yaw angle error because the roll and pitch angles are associated with the gravity vector when the acceleration of the object is calculated. As a result, the orientation particles with higher roll and pitch angle errors have lower weights and die out faster. To analyze the orientation estimation accuracy better, the RMS errors of the rotation matrices are compared in Figure 6-5. The results show that the rotation matrix error using an EKF increases over time. When the proposed method is applied, the rotation matrix error is significantly reduced and the error does not increase over time. The errors of the last 10 seconds in Figure 6-4 and Figure 6-5 did not change because the angular velocity was reduced to zero and the resampling was suspended when the object was stationary.

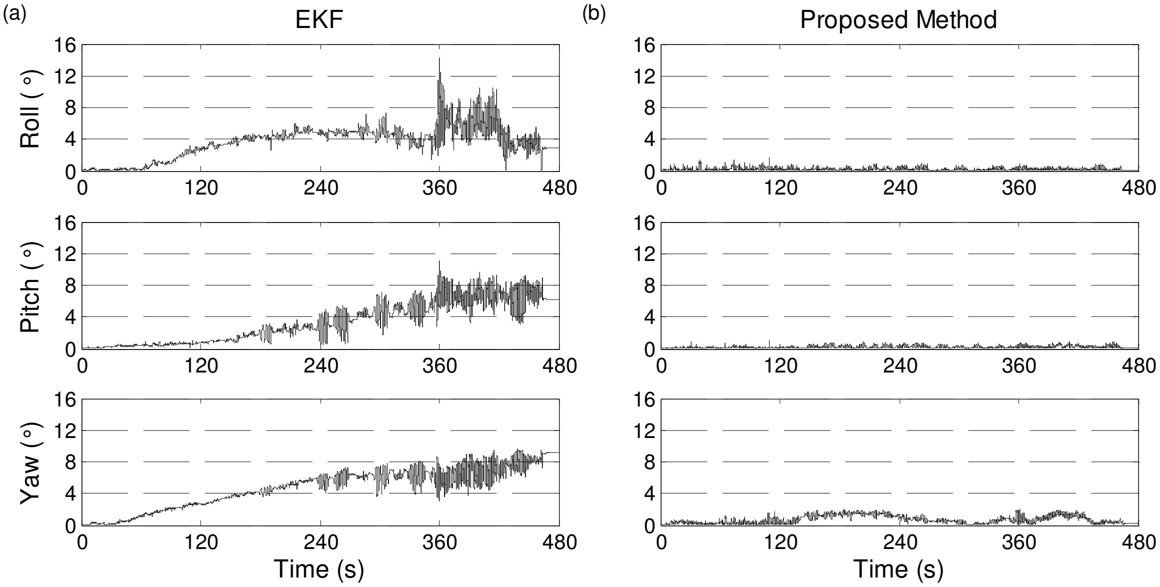


Figure 6-4: Orientation errors using (a) EKF and (b) the proposed method with 20 particles.

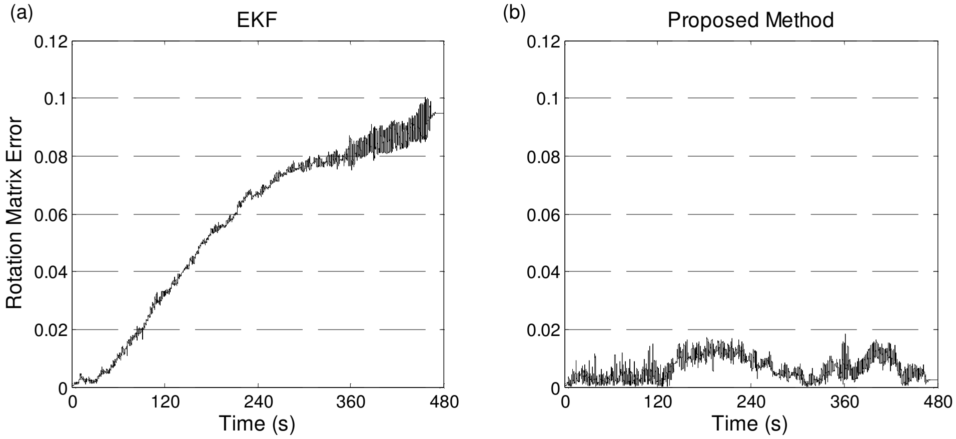


Figure 6-5: RMS rotation matrix error using (a) EKF and (b) the proposed method with 20 particles.

For the second analysis, the orientation errors were calculated using the proposed method with 5, 20, and 80 particles. Figure 6-6 and Figure 6-7 show the corresponding orientation errors in each case. When the number of particles is increased from 5 to 20, the orientation error is significantly reduced. However, when the number of particle is increased from 20 to 80, the graphs show no significant improvement. This experiment shows that increasing the number of particles does not proportionally improve the orientation estimation accuracy.

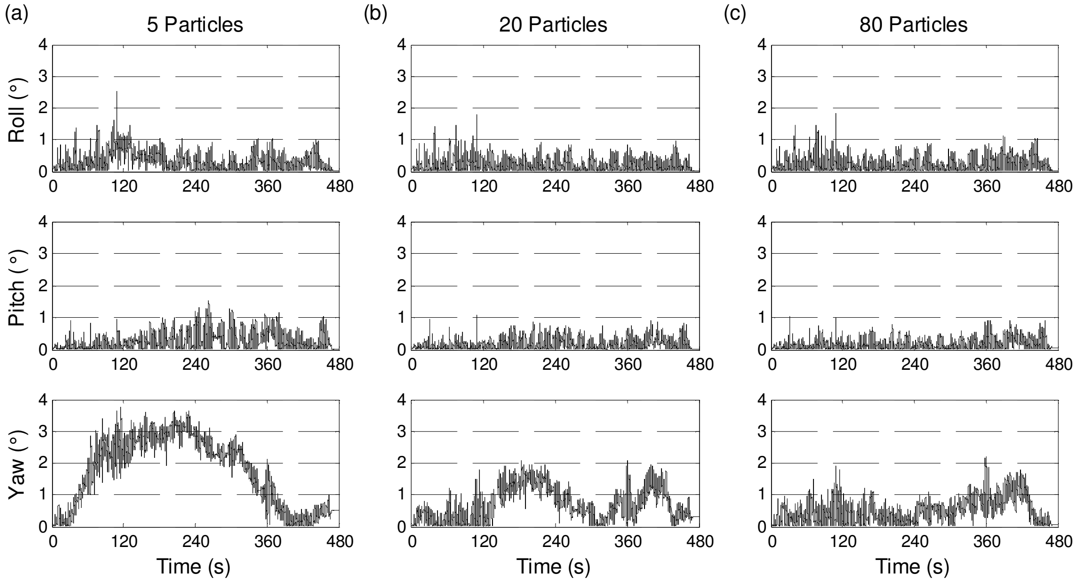


Figure 6-6: Euler angle errors using the proposed method when the initial orientation is known (a) 5 particles, (b) 20 particles, and (c) 80 particles.

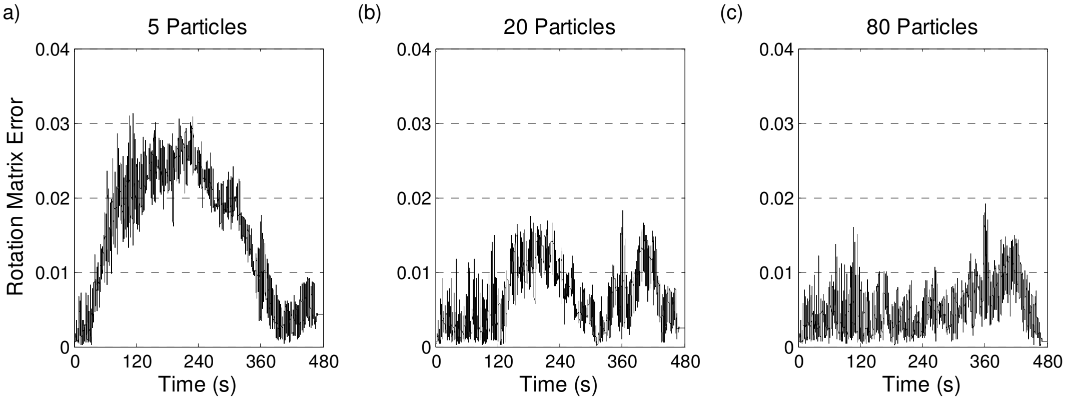


Figure 6-7: RMS rotation matrix errors using the proposed method when the initial orientation is known (a) 5 particles, (b) 20 particles, and (c) 80 particles.

For the third analysis, the initial orientation was assumed completely unknown, and the orientation of the object was estimated using the proposed filters with 5, 20, and 80 particles. The initial orientation is distributed using (6-19) at 1 second. In addition, ΔT_1 is chosen 3 seconds instead of 1 second so that the majority of particles with small weights die out in the first orientation iteration. When the initial orientation is unknown, high initial process noise is selected and then decreased gradually over time to its final value used in the previous two analyses.

The initial process noise and the settling time are reduced as the number of particles increase. When 5 particles are used, the initial process noise is 40 times higher than the final value. Then, the process noise converges monotonically to the final value in 390 second. When 20 particles are used, the initial process noise is 16 times higher than its final value and converges to this value in 150 second. When 80 particles are used, the initial process noise is four times higher than its final value. This process noise converges to the final value in 30 second.

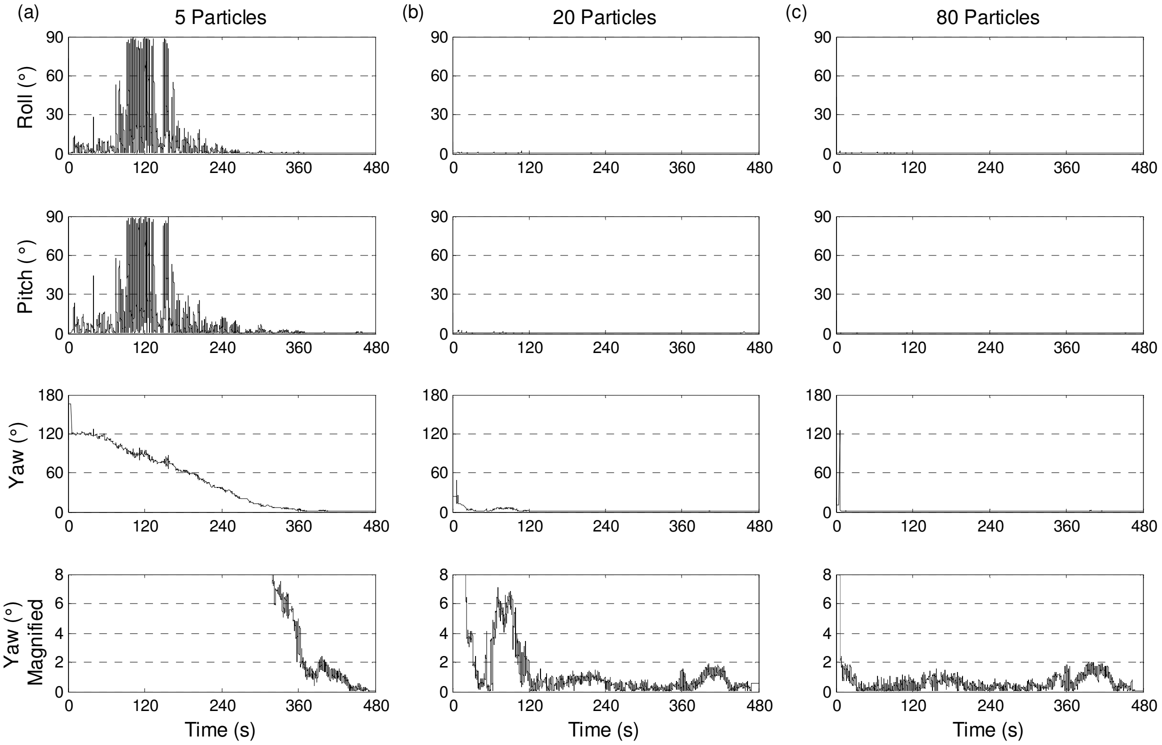


Figure 6-8: Euler angle errors when the initial orientation is unknown: (a) 5 particles, (b) 20 particles, and (c) 80 particles.

Figure 6-8 shows the absolute values of Euler angle errors (the actual roll and pitch angle errors vary from -90° to 90°, and the yaw angle error varies from -180° to 180°) for the previous three scenarios when the initial orientation is completely unknown. Figure 6-8 (a) depicts the orientation error using the proposed filter with 5 particles. Since only 5 particles are used to span the entire yaw angle range, this approximation cannot represent the true PDF very well. As a result, the particle with a high yaw angle error (120°) has the highest weight at the beginning and slowly converges to the correct orientation. Figure 6-8 (a) shows that the orientation converges to the correct values as time elapses even though the initial values were far from the correct orientation. When the yaw angle is around 90°, the roll and pitch angle errors become unstable in Figure 6-8 (a), which is a well-known Euler angle problem. The last row of Figure 6-8 shows the magnified yaw angle errors. The plots in the last rows of Figure 6-8 depict that as time elapses, the yaw angle errors almost match their counterparts in Figure 6-6. The last row of Figure 6-8 (b), the proposed method with 20 particles, illustrates that the correct orientation was found at 70 second. However, the yaw angle error started increasing significantly because the number of particles is small and the process noise is high. Since the process noise is designed to reduce monotonically to the final value in 150 s, the yaw angle error does not significantly increase after 120 second. Figure 6-8 demonstrates that the proposed filter finds the correct orientation faster as the number of particles increase.

For a close up view of Figure 6-8 (c), the first 15 seconds of the proposed method with 80 particles are shown in Figure 6-9. Initially, the orientation particles were randomly drawn from a uniform distribution. At 1 second, the roll and pitch angels were estimated by using (6-18). At this instant, the roll and pitch angle errors are close to zero, and the yaw angle is represented by 80 particles that are uniformly distributed using (6-19). Around 6 seconds, after 3 seconds of movement period excluding the stationary state, the orientation particle with about 130° difference from the correct yaw angle is chosen as the output because it has the highest weight. At 7 seconds, however, the yaw angle error is considerably reduced indicating that the filter contains a set of particles with low orientation errors. In other words, since 80 particles are used, many particles with low yaw angle error survived during the resampling at 6 seconds and were chosen as output at 7 seconds.

The same experiment was repeated with 20 particles without using (6-18) to estimate the initial roll and pitch angles. The initial process noise is 26 times higher than the final value and is allowed to converge to the final value in 250 seconds. Figure 6-10 displays the experimental results. As expected, this method requires a longer time to converge to the correct orientation due to the uncertainties in the roll and pitch angles.

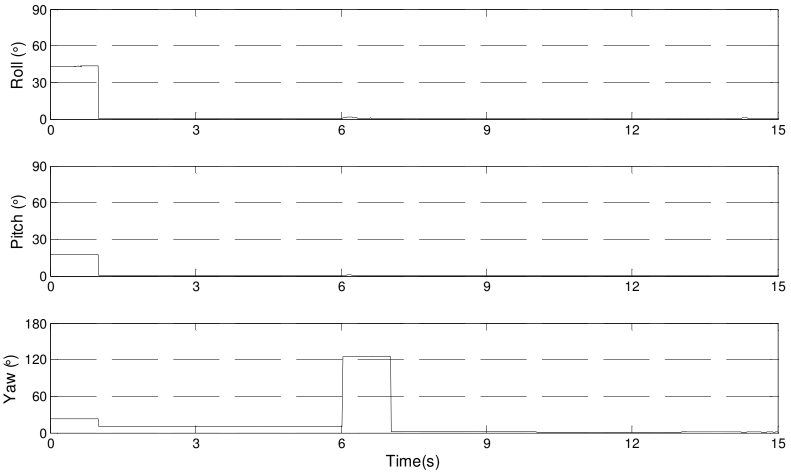


Figure 6-9: First 15 seconds of Figure 6-8 (c), the orientation error with 80 particles when the initial orientation is unknown.

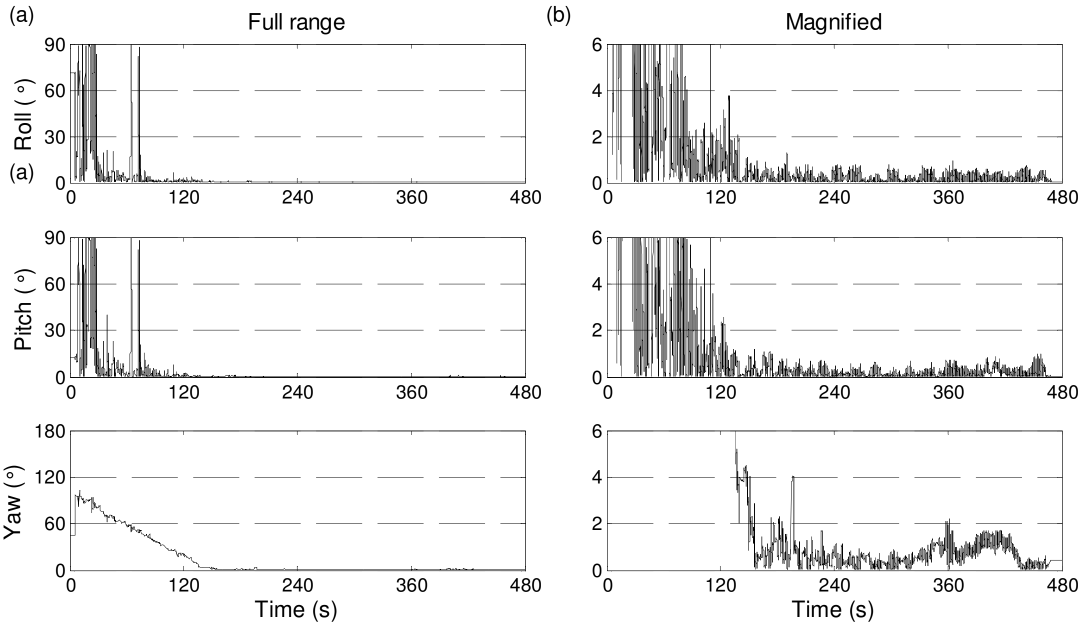


Figure 6-10: Euler angle errors using the proposed method with 20 particles when the initial orientation is unknown and accelerometer measurements are not used to estimate the roll and pitch angles, (a) full range and (b) range from 0° to 6°.

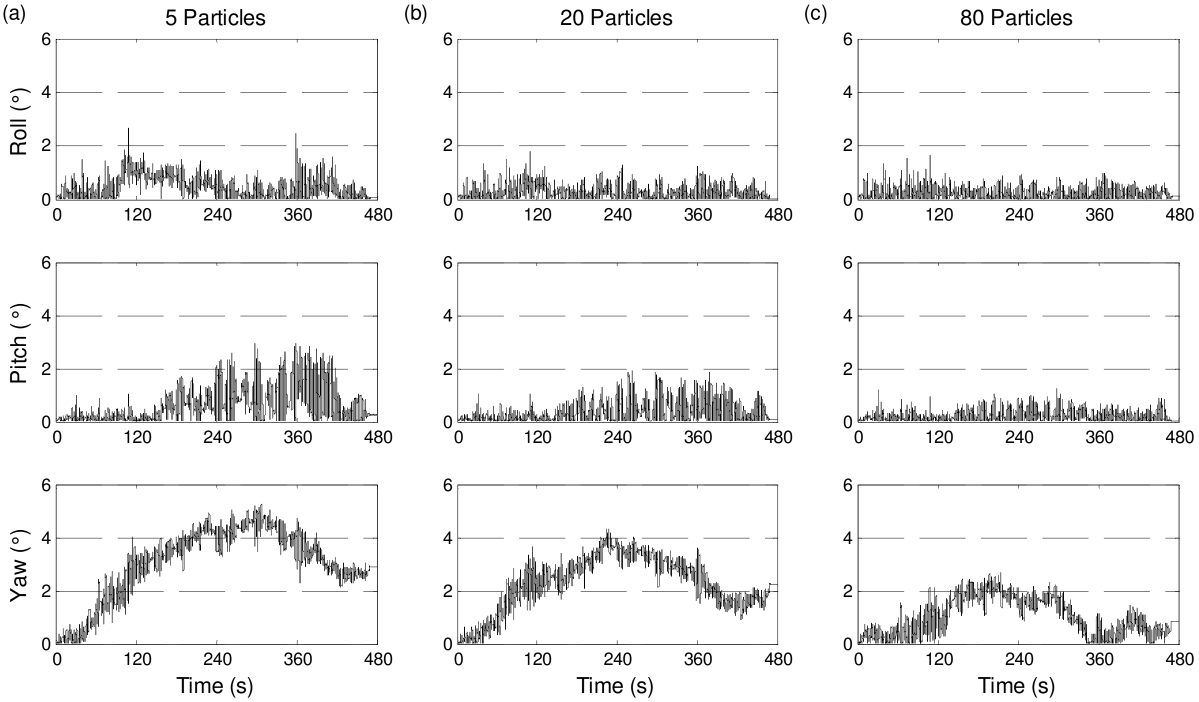


Figure 6-11: Euler angle errors of the proposed method with (a) 5 particles, (b) 20 particles, and (c) 80 particles when Gaussian noise is added to the position measurements.

Lastly, the effect of the position measurement noise on the orientation error is studied. To investigate the effects of the position measurement noise on the orientation and position accuracies, the same experiment was repeated with Gaussian noise added to the Optotrak position measurements. In this analysis, the position measurements are treated as the true position because Optotrak has only submillimeter error. Figure 6-11 shows the orientation error using the proposed method with additional position noise, and Fig 6-12 shows the corresponding position error. The plots in Figure 6-11 display lower orientation errors than those obtained using the EKF in Figure 6-4 (a). This is particularly evident for the roll and pitch angles. This result reveals that the proposed method has a good performance even in the presence of additional Gaussian position noise. As expected, Figure 6-6 and Figure 6-11 illustrate that the position measurement noise tends to reduce the orientation accuracy. Figure 6-11 also shows that the orientation accuracy increases as the number of particle increases. However, increasing the number of particles increases the computational cost as shown in Table 6-2. The proposed method was calculated by using Intel Core™2 Duo Processor E8400 with 3 Gb memory. Matlab was chosen to analyze the data off-line. Based on the results in Table 6-2, the proposed method with 80 particles is not feasible for on-line application with the current system because the processing time exceeds the experiment duration, 480 seconds. However, the processing time can be reduced by using faster processor or more dedicated solvers such as those that utilize C or Java development platform. It can also be concluded that the number of particles should be chosen to achieve a tradeoff between computational power and estimation accuracy.

Table 6-2: Number of particles and their processing time.

Number of particles	5 particles	20 particles	80 particles
Processing time	43 s	142 s	521 s

Figure 6-12 shows the position RMS error using three different approaches for position estimation: (i) no filter (added position noise), (ii) an EKF, and (iii) the proposed methods with three different particle sizes. The position estimations of both the EKF and the proposed methods show improved results over the case when no filter was used. The proposed methods in all three cases yield lower position errors than the EKF estimation method. Figure 6-12 also suggests that the proposed method provides higher position accuracy when the number of particles is increased.

Figure 6-11 relates that the position measurement noise affects the orientation accuracy, and the orientation error decreases as the number of particles increases. In the next experiment, a random position noise is added to the position measurements instead of a zero-mean Gaussian noise. The orientation errors of this test with different numbers of particles were compared in Figure 6-13, and the position error is compared in Figure 6-14.

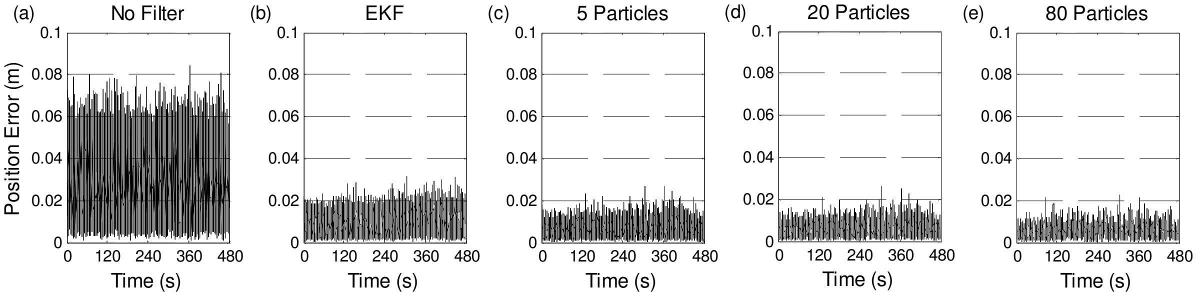


Figure 6-12: RMS position error when Gaussian noise is added to the position measurements: (a) added Gaussian noise using no filter (b) using an EKF (c) using the proposed filter with 5 particles, (d) using the proposed filter with 20 particles, and (e) using the proposed filter with 80 particles.

The position error in Figure 6-14 is a little bit worse than the position error in Figure 6-12. This is most likely because the position measurement error is not Gaussian. The orientation error comparison between Figure 6-13 and Figure 6-11 also shows that the orientation error is slightly less accurate when the position measurement has a uniform noise distribution. Since the proposed method utilizes the KF to estimate position, the accuracy can decrease when the position measurement noise is non-Gaussian. However, both the position and orientation estimations are improved when the proposed PF-KF combination is used.

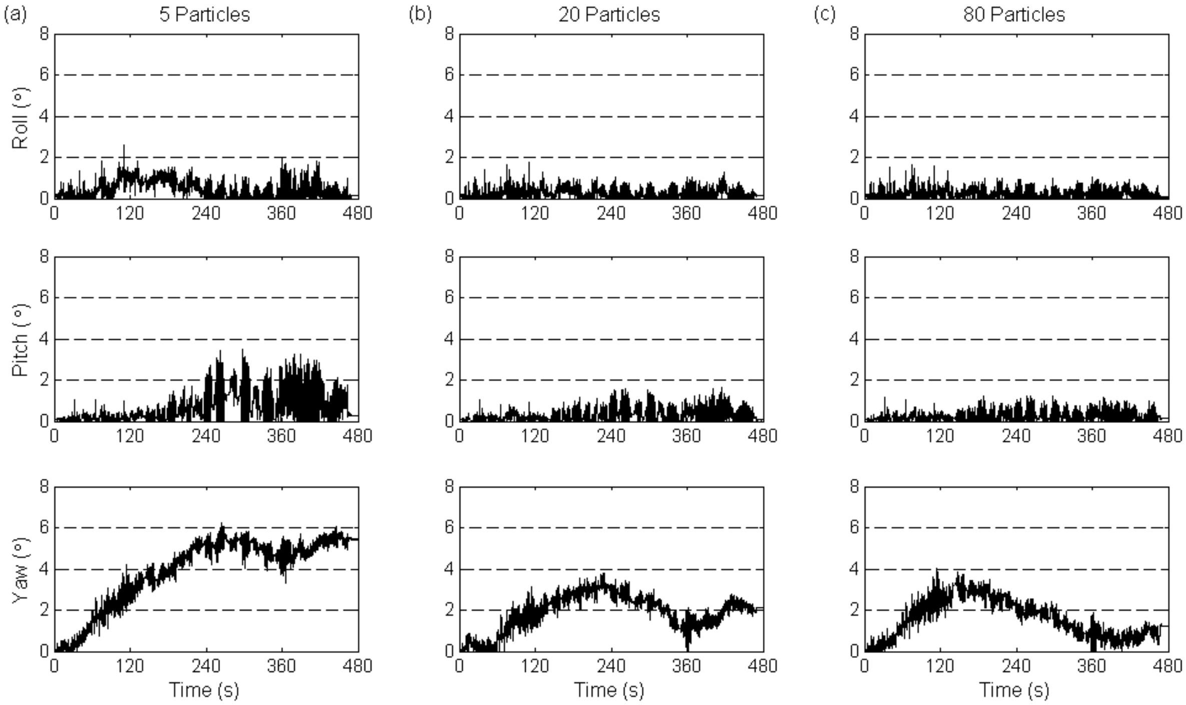


Figure 6-13: Euler angle error of the proposed method with (a) 5 particles, (b) 20 particles, and (c) 80 particles when random noise is added to position measurements.

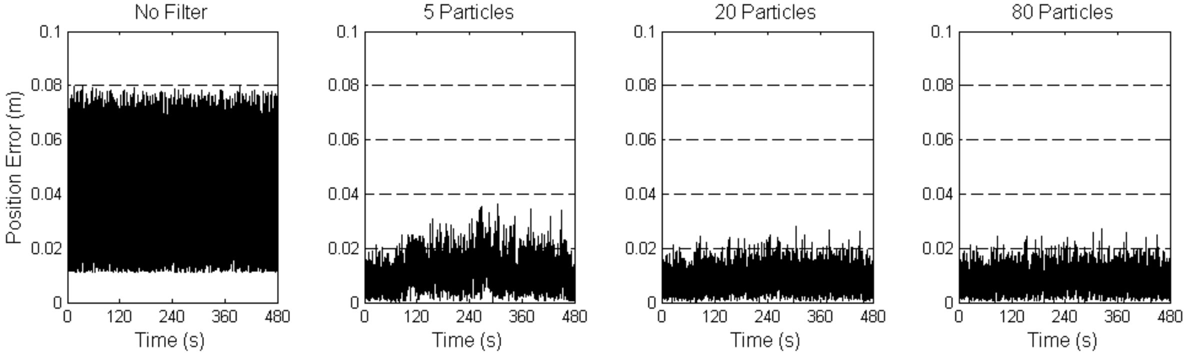


Figure 6-14: RMS position error when random noise is added to the position measurements: (a) added random noise using no filter (b) using the proposed filter with 5 particles, (c) using the proposed filter with 20 particles, and (d) using the proposed filter with 80 particles.

In the next experiment, the position sensor mimics the encoder-based position sensor that was used for the fastened bolt tracking system. The encoder-based position sensor does not have noise and have maximum 0.007 m of error. Also, the position error of the encoder-based position sensor gradually increases or decreases. This error characteristic of the encoder-based position sensor is represented with added sinusoidal position signal with the attitude of 0.007 m. The sinusoidal position error of each position axis has different frequencies (0.1, 0.15, and 0.3 rad/sec). Although the sinusoidal position error is a source of error, it is not measurement noise. Thus, the measurement noise covariance is the same as the noise covariance of Optotrak. Figure 6-15 reflects the orientation error comparison with different numbers of particles. The graphs show that they are almost the same as Figure 6-6, where the additional measurement error was not added. The orientation error is improved when the particle size is increased from 5 to 20, but there is no significant difference between the orientation errors with 20 particles and with 80 particles.

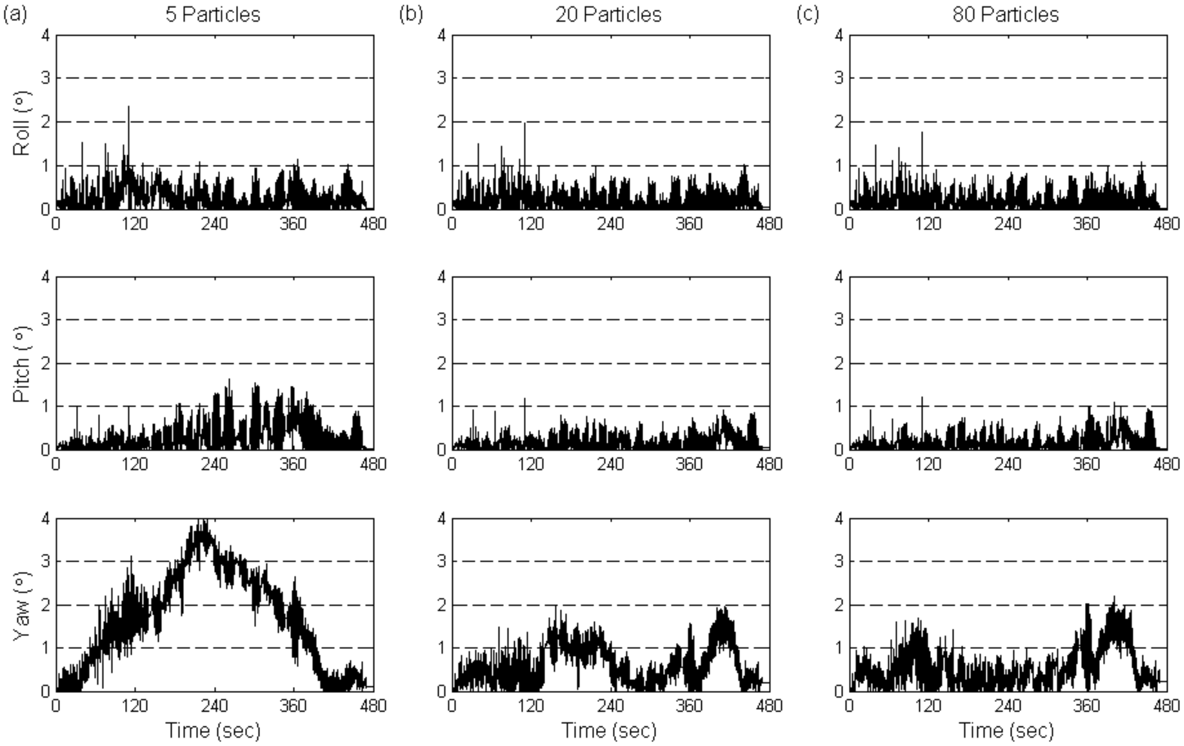


Figure 6-15: Euler angle error with the proposed method with (a) 5 particles, (b) 20 particles, and (c) 80 particles when sinusoidal position errors with an attitude of 0.007 m is added to the position measurements.

6.5 PF-KF-Based Tool Tracking System

6.5.1 Application to Tool Tracking System - Theory

The experiment used the same sensors and trajectory that were used in Section 5.4, and the position errors of the tool tip were compared when the position of the tool tip is known (e.g., when the tool fastens a bolt). Although the IMU was not factory calibrated, it was calibrated before use. The last experiment, Figure 6-15, mimics the encoder-based position sensor used in Section 5.4. Based on this experiment, the variables for the fastened bolt tracking system experiment are chosen. The filter with 20 particles is used for this application because Figure 6-15 shows no significant difference in the orientation error between the filter with 20 particles and the filter with 80 particles.

In order to apply the PF-KF tracking system to the fastening tool tracking application, the system should be able to identify the fastened bolt. First, the fastening action was identified by using Rule 5-

4 to Rule 5-7 in Table 5-2. When the fastening action was detected, the tool tracking system should identify the fastened bolt. To narrow down the possible fastened bolt, the system identifies which bolts are within the position sensor error range (PSE_n) by using (5-17). If only one bolt satisfies (5-17), the system concludes that the tool fastens that bolt. When more than one bolt satisfies (5-17), the orientation and position of each particle and the possible bolt positions are used to identify the fastened bolt. Let the calculated tool tip position components of particle *i* at time t_k be Tx_k^i, Ty_k^i, and Tz_k^i. Then, the position error of particle *i* with respect to Bolt *n* at time t_k is

$$PEP_n^i = ((Tx_k^i - n_x)^2 + (Ty_k^i - n_y)^2 + (Tz_k^i - n_z)^2)^{0.5} . \tag{6-18}$$

The bolt with the minimum PEP_n^i value is chosen as the fastened bolt for particle *i*. When all the particles indicate that Bolt *n* is being fastened, the system concludes that the tool is fastening Bolt *n*. Then, each particle is corrected by using the orientation correction method in subsection 5.3.4, which uses the position measurements from the position sensor and the position and orientation of the fastened bolt. When the orientation is corrected, all particles have the same orientation because the position measurements and the position and orientation of the fastened bolt are the same for all particles. In order to keep the variety of orientation particles, the particles are resampled immediately after the orientation correction with 1/10th of the original process noise value.

The advantage of the PF is that the initial states are not required to be known. When the initial orientation is unknown, the possible initial orientations can be calculated using the method in 6.3.2, which utilizes accelerometer measurements to estimate the tilt angles. In order to find the correct position of the tool tip, the tool tracking system waits until the first bolt is fastened if the initial orientation is unknown. When the first fastening action is detected and only one bolt satisfies (5-17), the system concludes that the tool fastens that bolt and all the particles converge to one orientation. However, when there are multiple possibilities for the fastened bolt, each possibility has the same number of particles and each particle tracks the position of the tool tip. Each possibility of the fastened bolt sequence resamples within its own sampling pool to avoid a possibility of the fastened bolt sequence disappear due to low weight. In order to evaluate the fastened bolt sequence, total accumulated position error ($TAPE_k^i$) is used, which is defined as

$$TAPE_{k+1}^i = TAPE_k^i + \{(P_{px,k}^i - P_{kx,k}^i)^2 + (P_{py,k}^i - P_{ky,k}^i)^2 + (P_{pz,k}^i - P_{kz,k}^i)^2\}^{0.5} . \tag{6-19}$$

$TAPE_k^i$ is zeroed when there are multiple possible fastened bolts for the first time or when there is one possibility of previous fastened bolt and multiple possible fastened bolt exist currently. When the next fastening action is detected, each particle identifies the fastened bolt by identifying the bolt with the minimum PEP_n^i. Since PEP_n^i is calculated with respect to the almost absolutely correct value (bolt position), it was weighted twenty times more than the accumulated position error. When the fastening action is detected, $TAPE_k^i$ is updated as follows:

$$TAPE_{k+1}^i = TAPE_k^i + 20 \times PEP_n^i \,. \tag{6-20}$$

When all the particles indicate that they are fastening the same bolt, the system chooses the particle with the minimum $TAPE_k^i$ value as the most probable possibility. Then, the system outputs the bolt fastened sequence of the chosen particle, and all $TAPE_k^i$ values are set to zero. However, if there are still multiple possibilities, $TAPE_k^i$ keeps accumulating until there is only one possible fastened bolt.

When the calculated position error of a particle is higher than the position error limitation of the PF-KF method, the particle should be removed because this particle has an incorrect path. Therefore, the maximum position error of the tool tip should be identified. From Figure 6-15, the maximum error of the PK-PF system with 20 particles is 2°. However, since this 2° error is calculated from the particle with the highest weight, an extra 2° of orientation error margin is added to ensure that all the possible particles that fasten the bolt are included. Thus, 4° is chosen as the maximum orientation error of the PK-PF system with 20 particles. Since the maximum position measurement error is 0.007m, the maximum position error of the tool tip position is

$$MPE = 165mm \times \sin(2 + 2°) + 7mm = 18.5mm = 0.0185m \,. \tag{6-21}$$

From (6-21), it is concluded that all the bolts should be at least 0.037 (0.0185 × 2) m apart for the system with 20 particles to identify the fastened bolt correctly. In case there are four possible fastened bolt positions, each possibility has five particles. In this case, the maximum orientation error is 4° as shown in Figure 6-15. Then, the *MPE* with a 2° error margin can be written as

$$MPE = 165mm \times \sin(4 + 2°) + 7mm = 24.2mm = 0.0242m \,. \tag{6-22}$$

Therefore, when *MPE* is greater than 0.025 m, the i^{th} particle is most likely not fastening Bolt *n*. However, to keep more variety, 0.035 m is used as the position error limitation. Therefore, if the distance between Bolt *n* and the particle *i* is greater than 0.035 m, the system removes this particle and replaces it with a particle whose PEP_n^i value is less than 0.035 m. To reduce this position error limitation, more particles can be used. For example, if 80 particles are used, *MPE* becomes 0.0185 m instead of 0.0242 m when there are four possibilities; thus, the position error limitation is lowered and better accuracy can be achieved. Therefore, it is beneficial to use more particles when the bolts are closely placed.

6.5.2 Experiment Results

Four different scenarios with two different tool trajectories are examined for the tool tracking experiments. In the first scenario, the results using the KF-based system described in Section 5.4 and the results using the PF-KF-based system with 20 particles are compared. For the second scenario, the tool was moved beyond the angular velocity measurement limitation and the experimental results of the two methods were compared. The third experiment shows how the PF-KF-based system finds the correct orientation when the initial orientation is completely unknown. The last experiment shows the scenario when the initial orientation is unknown and there are multiple possible fastened bolts.

For the first scenario, the PF-KF-based system with 20 particles was used in the tool tracking application, and the results are compared with those using the KF-based system in Chapter 5. Table 6-3 summaries the position error using the KF-based system and using the PF-KF-based system with 20 particles. The results show that the PF-KF-based system has generally a higher accuracy than the KF-based intelligent system.

In order to see how accurately the PF-KF-based system can estimate the orientation when the angular velocity is slightly unreliable, the tool was rotated beyond the angular velocity measurement range of the gyros. Figure 6-16 shows the angular velocity of each axis for the second experiment. Figure 6-16 (a) depicts the entire time range of the experiment and Figure 6-16 (b) displays the duration that the angular velocity of the tool is beyond the linear range of the gyros. The dashed lines in the figures represent the measurement limit of gyros, which is ±75°/sec. When the angular velocity is beyond this rate, the nonlinearity can be higher than the specification. The figures indicate that the angular velocities in the x-axis and the y-axis are far beyond the linear range. For the y-axis, the angular velocity measurements between 67.7 second to 69 second are cut off because the angular velocity output is beyond the measurement limit. The comparison results of the second test are summarized in Table 6-4.

Table 6-3: Position error comparison between the KF-based system and the PF-KF-based system.

Tool position	Time (s)	Position error with the KF-based system (mm)				Position error with the PF-KF-based system with 20 particles (mm)			
		X-axis	Y-axis	Z-axis	Total	X-axis	Y-axis	Z-axis	Total
Bolt 1	19	-2	0	6	6	-3	6	-2	7
Bolt 4	43	-12	-6	-6	15	-6	4	-2	7
Bolt 5	85	-2	-8	2	8	-5	5	4	8
Bolt 6	89	-4	-2	2	5	-4	2	5	7
Bolt 7	184	3	-6	7	10	1	1	6	6
Finish	204	-1	-6	0	6	-1	-2	-5	5

In Table 6-4, both systems correctly identified the fastened bolt. Most of the bolts exhibit a low position error, but the error of Bolt 5 is much higher than that of the others due to the angular velocity measurement error between 65 second and 69 second. This error resulted in 34 mm when the KF-based system is used. However, when the PF-KF-based system was used, the error is 25 mm. This is due to the fact that the KF is suitable for zero-mean Gaussian noise while PF can adapt to any noise distribution.

The orientation error due to angular velocity measurement error can be considered as an uncertainty. Thus, the error can be reduced when a higher process noise is chosen and more particles are placed to compensate for the high uncertainty. Table 6-5 shows the position error of the PF-KF-based system with different numbers of particles and covariance. In general, there is no significant difference among the nine scenarios that are presented in Table 6-5 except for Bolt 5. When 20 particles are used, or the covariance is 0.006, there is no significant difference. When the number of particle and covariance are increased, the position of Bolt 5 is estimated more accurately. However, when the number of particles is changed from 40 to 60, there is no significant difference in the position accuracy. This experiment shows that when the uncertainty increases, both the measurement covariance and the number of particles should be increased to achieve a better result.

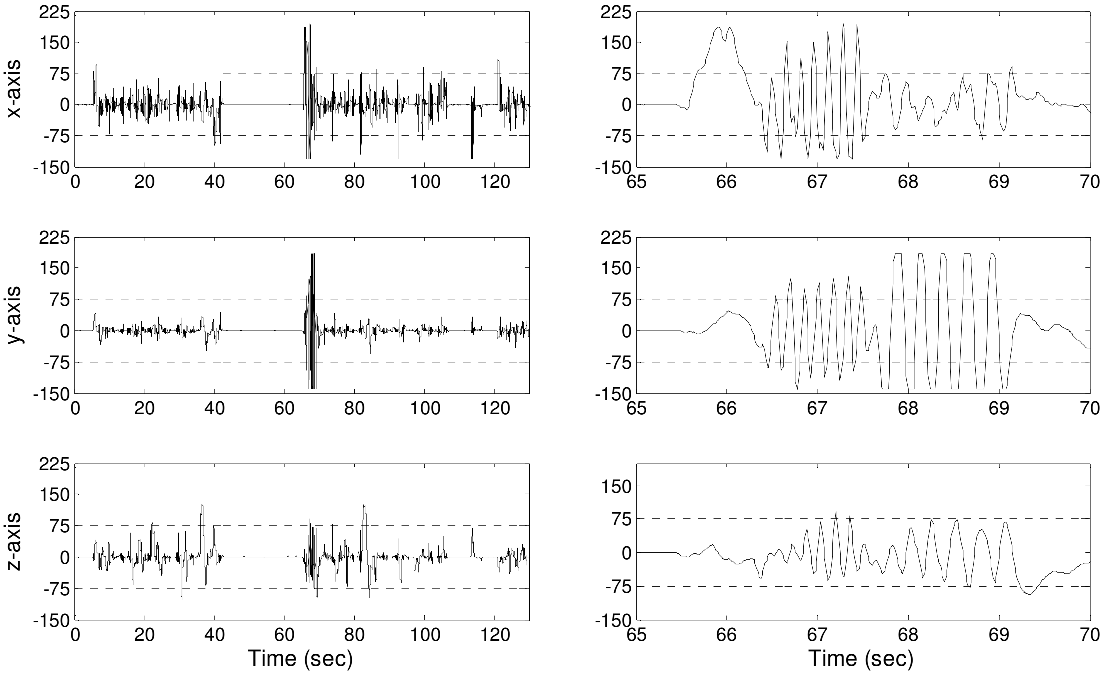


Figure 6-16: The angular velocity (°/sec) of each axis: (a) the entire time span and (b) between 65 second and 70 second.

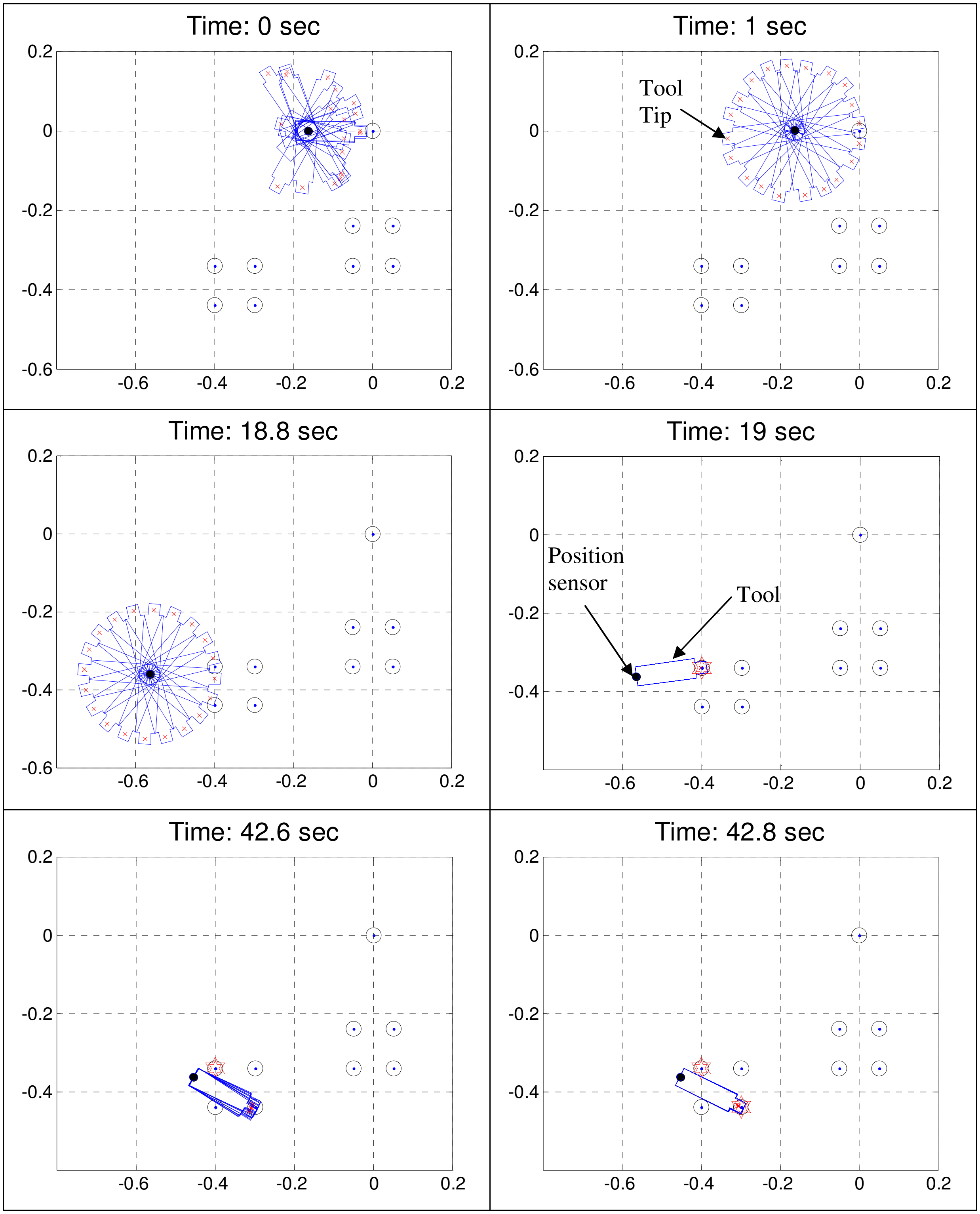
Table 6-4: Position error comparison between the KF-based system and the PF-KF-based system when the angular velocity components have high errors.

Tool position	Time (sec)	Position error with the KF-based system (mm)				Position error with the PF-KF-based system with 20 particles (mm)			
		X-axis	Y-axis	Z-axis	Total	X-axis	Y-axis	Z-axis	Total
Bolt 1	11	-6	2	4	7	-6	2	-3	7
Bolt 2	21	-5	0	-2	5	-4	0	0	4
Bolt 3	27	-2	6	0	6	0	1	3	3
Bolt 4	34	-5	3	5	8	-5	3	2	6
Bolt 5	73	-7	-10	-32	34	-6	-6	-23	25
Bolt 6	79	-5	3	-8	10	-5	3	-2	6
Bolt 7	89	-1	0	10	10	-1	0	-1	1
Bolt 8	102	0	12	1	12	-3	4	4	6
Finish	135	-2	4	-2	5	2	1	4	4

Table 6-5: Total position error comparison among the PF-KF-based system with different numbers of particles and different covariance when the angular velocity components have high errors.

	20 particles			40 particles			60 particles		
Covariance	0.006	0.009	0.012	0.006	0.009	0.012	0.006	0.009	0.012
Bolt 1	7	7	7	7	7	7	7	7	7
Bolt 2	4	5	6	4	4	5	4	4	4
Bolt 3	3	1	3	1	5	2	1	1	0
Bolt 4	6	6	6	6	6	7	6	6	6
Bolt 5	25	29	22	24	19	8	23	19	9
Bolt 6	6	6	8	8	7	6	6	6	6
Bolt 7	1	1	1	1	2	1	0	2	1
Bolt 8	6	6	7	5	8	8	5	5	5
Finish	4	10	5	5	6	7	5	8	2

The third experiment shows the case where the trajectory is the same as the first experiment but the initial orientation is unknown. The experimental results are shown in Figure 6-17. The two rectangles symbolize the tool (clearly shown at 19 second) and the small black dot signifies the position sensor measurement. The "x" marks represent the calculated positions of the tool tip. When the tool identifies the fastened bolt, two hexagrams are marked on the bolt position. First, the orientation of the tool was randomly selected. At 1 second, the orientation of the tool was estimated using (6-19), utilizing the average accelerometer measurement of each axis. Since the yaw angle is unknown, the particles were equally spread. Even if the tilt angle is not corrected this way, the full orientation can be identified when the first fastened bolt is identified. At 18.8 second, there are still 20 different possibilities of orientations, but when the system identified the fastening bolt at 19 second, the orientation converges to one solution because only one bolt satisfy (5-17). The remaining figures in Figure 6-17 indicates that the particles spread over time, but the particles are very closely placed immediately after the fastened bolt is identified.


Time: 0 sec
Time: 1 sec
Tool
Tip
Time: 18.8 sec
Time: 19 sec
Position
sensor
Tool
Time: 42.6 sec
Time: 42.8 sec
0.2
0
-0.2
-0.4
-0.6
-0.6
-0.4
-0.2
0
0.2

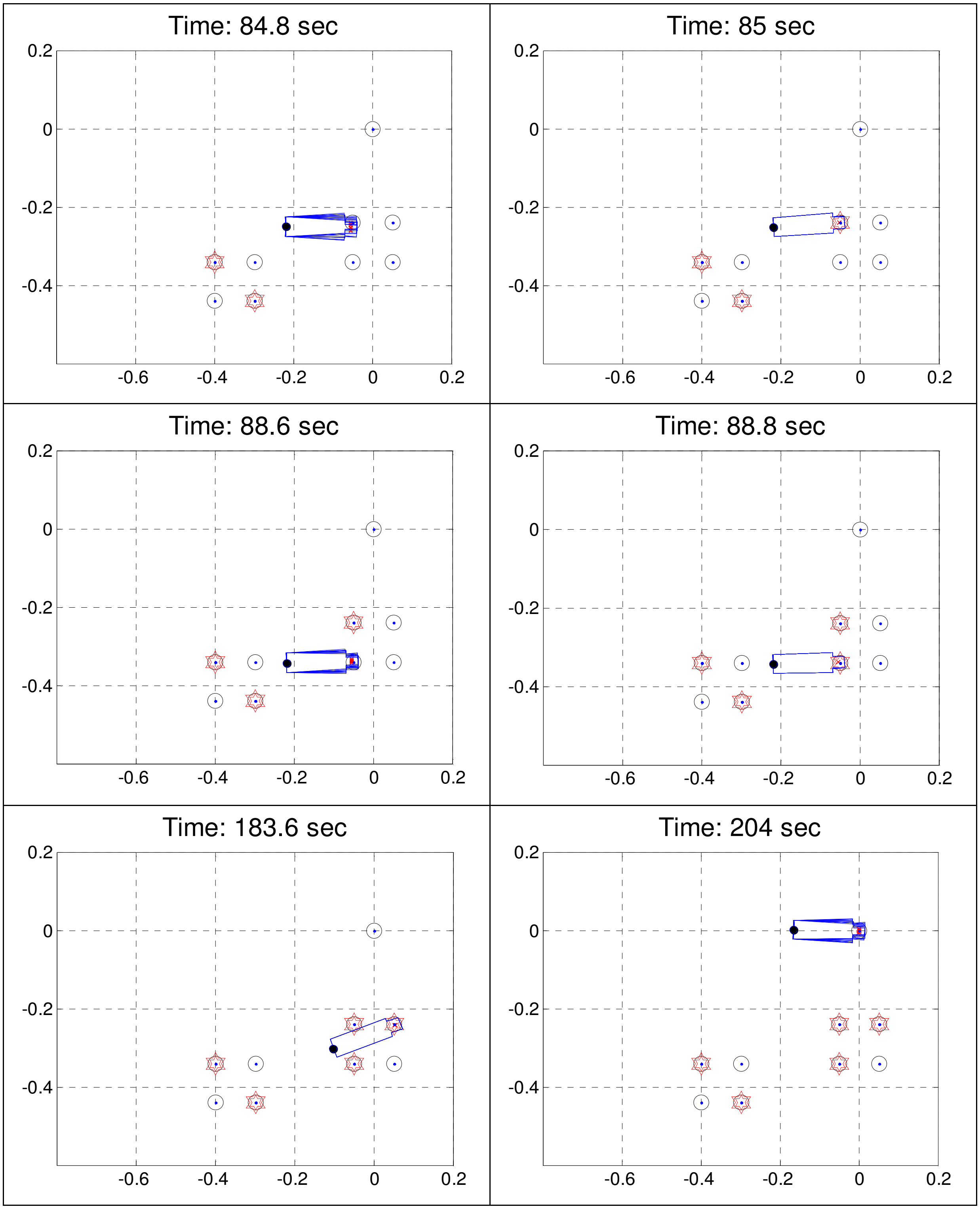


Figure 6-17: Tracking the fastened bolts when the initial orientation is unknown.

For the fourth experiment, four additional bolt positions (Bolt 9 to Bolt 12) are added so that multiple possibilities for the fastened bolt can be presented. All bolt positions are follows.

Table 6-6: Bolt locations.

Bolt number	Bolt position	Bolt number	Bolt position
Bolt 1	(-0.4. -0.34, 0)	Bolt 7	(0.05, -0.24, 0)
Bolt 2	(-0.4. -0.44, 0)	Bolt 8	(0.05, -0.34, 0)
Bolt 3	(-0.3. -0.34, 0)	Bolt 9 (added)	(-0.46. -0.24, 0)
Bolt 4	(-0.3. -0.44, 0)	Bolt 10 (added)	(-0.41. -0.3, 0)
Bolt 5	(-0.05, -0.24, 0)	Bolt 11 (added)	(-0.4. -0.39, 0)
Bolt 6	(-0.05, -0.34, 0)	Bolt 12 (added)	(-0.3 -0.4, 0)

Three bolt positions are added so that four possible fastened bolt positions exist when the first fastening action is detected, and one bolt position is added so that two possible fastened bolt positions exist when the second fastening action is detected. Figure 6-18 shows the experimental results. At 0 second, the orientation is unknown, so randomly chosen orientations were selected as the initial orientation. At 19 second, the first fastening action was detected, and there are four possible fastened bolt positions (Bolt 1, Bolt 9, Bolt 10, and Bolt 11) that satisfy (5-17). For each possible fastened bolt position, a square mark is placed. Since there are four possible fastened bolts, five particles are allocated to each possibility.

At 42.6 second, there are four sets of possible positions of the tool tip, and it appears three possible fastened bolts exist (Bolt 3, Bolt 4, and Bolt 12, whose previous fastened bolts are Bolt 9, Bolt 1, and Bolt 10 respectively). However, only two bolts (Bolt 4 and Bolt 12) satisfy (5-17). The particles whose previous fastened bolts are Bolt 10 and Bolt 1 are fastening Bolt 12 and Bolt 4 respectively (the particles that fastened Bolt 10 and then fastened Bolt 12 in sequence will be denoted as Bolt 10-12 hereafter). Since Bolt 3 did not satisfy (5-17), the particles that previously fastened Bolt 9 were either resampled or assumed fastening Bolt 12 based on the distance between the particle and the position of Bolt 12. If the particle is placed more than 0.035 m from Bolt 12, this particle has likely chosen incorrect previous fastened bolt. Therefore, the particle is removed and replaced with a random particle that has less than a 0.035 m error from the possible fastened bolt. When the particle is less than 0.035 m from Bolt 12, it was assumed that the particle is fastening Bolt 12 (Bolt 9-12).

The particles previously fastened Bolt 11 were also either resampled or assumed fastening Bolt 4 (Bolt 11-4).

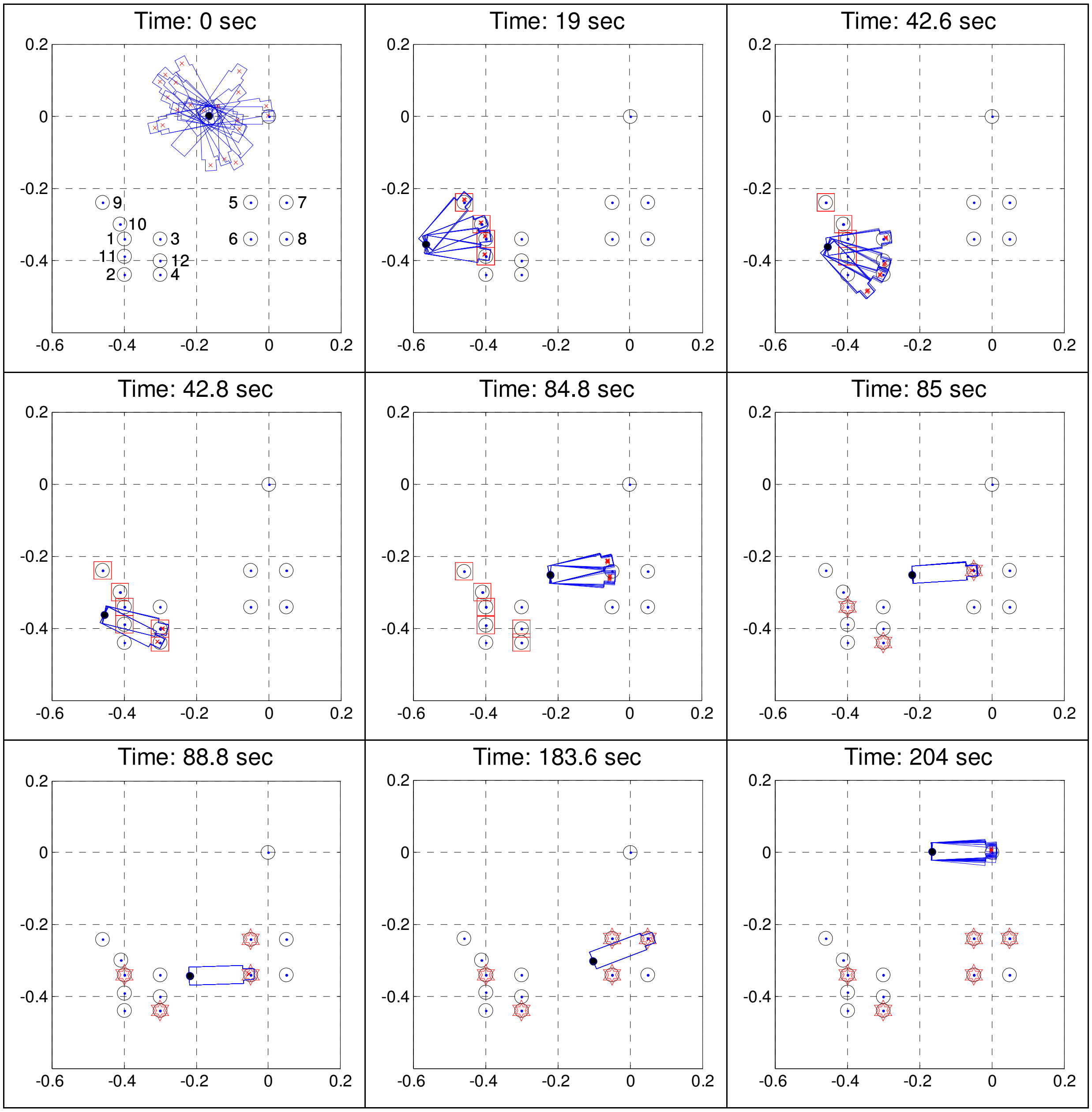


Figure 6-18: Fastened bolt detecting sequence. Square blocks indicate possible fastened bolts and two hexagrams on the bolt signify the fastened bolts.

At 84.8 second, there is only one possible fastened bolt, and the two possibilities are very close to Bolt 5. When the fastening action was detected at 85 second, the total accumulated position errors ($TAPE_k^i$) are compared to identify the fastened bolt sequence. At this point, the possible fastened bolt sequences could be a) Bolt 9-12-5, b) Bolt 10-12-5, c) Bolt 1-4-5, d) Bolt 11-4-5, e) Bolt 1-12-5, f) Bolt 10-4-5. The system concludes that the particle with the Bolt 1-4-5 sequence is the correct fastened bolt sequence because this particle has the minimum total accumulated position error. Thus, Bolt 1, Bolt 4, and Bolt 5 are marked with two hexagrams on the bolt position at 85 second, and the other possibilities (square marks) are removed. This is the correct sequence as shown in Table 6-3. After identifying the true fastened bolt sequence, the system can identify the rest of the fastened bolts without any multiple possibilities. This experiment was repeated twenty times, and the system identified the correct fastened bolt sequence in each trial.

The previous experiment was repeated with the trajectory illustrated in Table 6-5. At 10.8 second, four possible fastened bolt positions exist (Bolt 1, Bolt 9, Bolt 10, and Bolt 11). When the next fastening action was detected at 20.8 second, five bolts satisfied (5-17) (Bolt 1, Bolt 2, Bolt 9, Bolt 10, and Bolt 11), but the possible fastened bolts are reduced to three positions based on the particle positions (Bolt 1, Bolt 11, and Bolt 2). Then, the possible bolt sequence was reduced to three (Bolt 9-1, Bolt 10-11, and Bolt 1-2) because the sequence possibility of Bolt 11-2 is eliminated due to a high position error (more than 0.035 m). When the next fastening action was detected, the system concludes that the tool is fastening Bolt 3, and the system identifies the previous fastened bolts according to the total accumulated position error. Since the sequence with Bolt 1-2-3 has the minimum total accumulated position error, the system concludes that the fastening sequence is Bolt 1-2-3. This is the correct fastened sequence as shown in Table 6-5. After the correct path is found, the system also estimate the rest of the fastened bolts in the correct sequence.

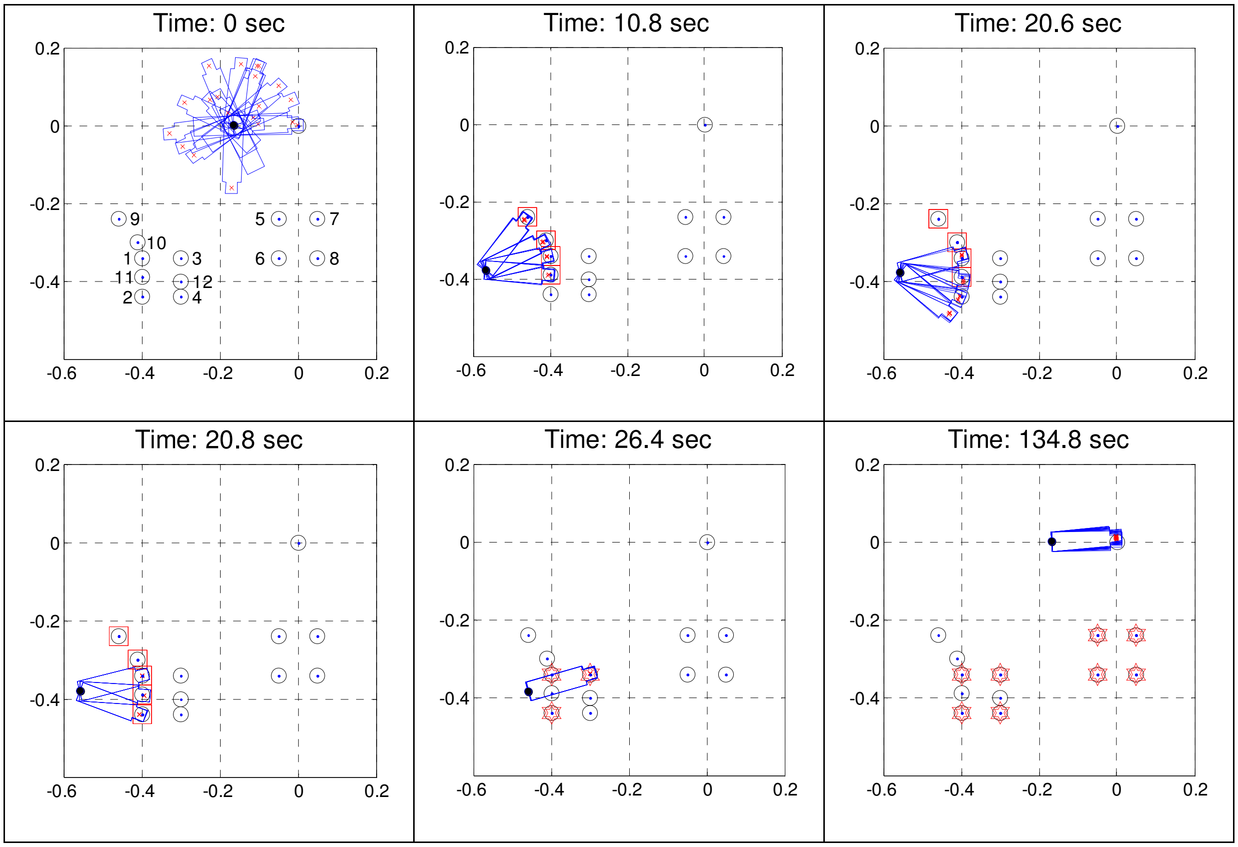


Figure 6-19: Fastened bolt detecting sequence. Square blocks indicate possible fastened bolts and two hexagrams on the bolt signify the fastened bolts.

6.6 Conclusion

This chapter presented a position/orientation estimation method using a KF-PF combination and how this method was applied to the fastening tool tracking system. The proposed KF-PF combination method uses the PF to estimate the object orientation while the KF is used to estimate the position. In addition, an expert system was developed to correct the angular velocity bias as well as to identify the stationary state of the object.

The experimental results indicate that the orientation errors using the proposed method are significantly reduced compared to the errors using the EKF estimation method. Two factors that affect the orientation error were studied: (i) the number of particles and (ii) position sensor noise. As the number of particles increases, the orientation error by using the proposed method tends to decrease. However, increasing the number of particles requires higher computational costs.

Therefore, the number of particles should be chosen to achieve a tradeoff between accuracy and computational efficiency. The experimental results demonstrate that the proposed method can estimate orientation even in the presence of Gaussian or random position sensor noise. However, the experimental results also reveal that the position noise decreases the orientation estimation accuracy. The experiments demonstrated that the proposed filter can find the correct orientation of the object even when the initial orientation is completely unknown.

The KF-PF-based system yields a higher accuracy, and the orientation does not increase over time. The test results of KF-PF-based system indicate that the proposed system can estimate the fastened bolt even when the bolts are closely placed (0.04 m apart). When the bolts are more than 0.04 m apart and the initial orientation is known, the fastened bolt can be correctly identified. For a workpiece that requires a specific fastening order such as an engine mount, the KF-PF-based system can be used as a fastening tool control system. When the operator tries to fasten a bolt out of sequence, the tool control system can be programmed so that the power of the tool is off. In addition, the tool control system can have multiple torque settings based on the bolt position. For example, when two bolts have the same bolt size but each requires different torque, one tool can be used to fasten both bolts by using the control system that controls torque based on the tool tip location.

The KF-PF-based system can yield accurate orientation estimation even when the gyros have a small error and when the initial orientation is unknown. When high computational power is available, the KF-PF combination system is the better choice to estimate the position and orientation of the tool than the KF system described in Chapter 5.

Chapter 7
Thesis Contributions and Future Work

7.1 Thesis Contributions

In this thesis, four contributions related to fastening tool tracking systems are proposed. These objectives are summarized as follows.

- An accurate triaxial calibration method that has a simple procedure.
- A tool tracking system that does not require a line of sight by using an IMU and a triaxial magnetometer.
- A tool tracking system by using KFs and a fuzzy expert system with an IMU and an encoder position sensor.
- A tool tracking system by using the KF-PF combination with an IMU and an encoder position sensor.

An accurate calibration method for an IMU is important especially when the system does not have a position sensor to correct the position estimation. The newly developed triaxial calibration method only requires placing the sensor in six different tilt angles for calibration and offers a high accuracy.

Three different fastening tool tracking systems using an IMU and one additional sensor are developed. The first system relies on an IMU and a triaxial magnetometer. In order to reduce the position error, an expert system is used to correct the velocity, position, and orientation error. The advantage of this low-cost system is that it does not require lines of sight and has a low computational cost. However, even with the position error correction algorithm, the position error grows rapidly over time. Therefore, this tracking system can only be used for a workpiece that has different bolt orientations or the distances between the bolts are far from each other.

To overcome the disadvantages of such a system, an encoder-based position sensor is employed instead of a triaxial magnetometer. The KF is developed to hybridize the IMU and the position sensor. A fuzzy expert system is utilized to identify the fastening action and correct the position and orientation. Although this system is computationally inexpensive, the orientation error grows over time, which affects the position estimation of the tool tip. This system is useful for applications where the bolts are not too closely apart or the fastening sequence is not important.

The third system uses the KF-PF combination to hybridize a position sensor and an IMU. The experimental results demonstrate that the maximum orientation error of the presented system with 20 particles is about 2º, and the orientation error of the system does not grow over time. Therefore, the KF-PF combination system can be applied to build a tool control system so that the tool is powered on only when the tool is in designated positions. Then, an operator cannot fasten bolts out of fastening sequence. The KF-PF-based system can find the fastened bolt accurately even when the initial orientation is unknown. In addition, the proposed method can accurately estimate orientation by utilizing enough number of particles and high process noise even when the gyros have a small error. The advantages and disadvantages of each system are summarized in Table 7-1.

Table 7-1: Advantages and disadvantages of the system presented in this thesis.

System	Advantages	Disadvantages
Option 1: IMU and magnetometer using an expert system	1. No line of sight requirement. 2. Low computational cost. 3. Low cost.	1. Low accuracy. 2. The bolts must be a long distance apart or the orientations of the bolts should be different.
Option 2: IMU and encoder position sensor using KF and fuzzy expert system	1. Computationally inexpensive. 2. The fastened bolt can be accurately found when the bolts are some distance apart.	1. A line of sight is required. 2. The position and orientation errors of the tool tip slowly grow over time.
Option 3: IMU and encoder position sensor using KF-PF and expert system	1. The tool tip position can be accurately estimated. 2. The tool tip position can be estimated even when the initial orientation is unknown. 3. A fastening tool control system can be built.	1. A line of sight is required. 2. High computational cost.

7.2 Future Work

This thesis describes the developments of tool tracking systems that utilize one IMU and one additional sensor. The most attractive option for a fastening tool tracking system is to use a high accuracy position sensor without a line of sight requirement. Since an IMU must be attached to a fastening tool, the sensor has to be small and lightweight. This limitation led to the use of a MEMS IMU whose accuracy is limited. However, since the MEMS IMU technology is advancing rapidly, more accurate MEM IMUs can be used to measure the acceleration and angular velocity. This, in turn, will lead to a more accurate position and orientation estimations especially when Option 1 in Table 7-1 is selected.

In this thesis, an encoder-based position sensor was chosen as the hybridization for Option 2 and Option 3. In the future, other position sensor options can be studied. Although it is challenging to use a magnetic position sensor in the automotive industry, this sensor is a good option. This is due to the fact that a magnetic position sensor has a high accuracy and does not require a line of sight. Also, an UWB position sensor can be a good option when lower position accuracy is required. When an UWB position sensor is hybridized with an IMU, the position accuracy will be greatly improved.

Option 3 in Table 7-1 offers a possible tool torque control system that can be used to control the power of the tool based on the position of the tool tip. Therefore, this work can be extended to the development of a tool torque control system which outputs the torque based on the position and orientation information obtained from the system presented in Option 3.

In this thesis, fastening tool tracking systems are investigated as an application of a remote sensing. However, the application of remote sensing is not limited to tool tracking systems. Therefore, a general hybrid method of the KF that combines an IMU and a position sensor is investigated in Chapter 5, and a general hybrid method of a KF-PF combination was developed to estimate the position and orientation in Chapter 6.

In the future, the remote sensing system using the KF-PF combination can be applied to estimating the position and orientation of a rehabilitation patient's body part by using an ultrasonic position sensor, and electromagnets, and an IMU. In such an application, a MEMS IMU is attached to the position of interest and an ultrasonic emitter is attached to the IMU. When the lines of sight are available, the position and orientation can be estimated with the position sensor and the IMU using the KF-PF combination method. Meanwhile, a neural network system is used to create a magnetic field map related to the position. Since the magnetic field can be changed when an electrical device is switched on or off, the map might be updated if necessary. When occlusion occurs, the position information can be retrieved from the magnetic field map, and the KF-PF combination system can be used to estimate the position and orientation of the body part. By using the position and orientation estimations, the rehabilitation system will direct the movement of the patient on a monitor.

Appendix A: Quaternions

The quaternion was developed by William Rowan Hamilton to extend 3D vector algebra. He introduced a concept of a hyper-complex number of rank 4: one real number and three imaginary numbers (*i*, *j*, and *k*). A quaternion *q* can be written as:

$$q = q_0 + q_1 i + q_2 j + q_3 k \ , \tag{A-1}$$

or it can also be written as:

$$q = q_0 + q_v \, , \tag{A-2}$$

where q_0 is the scalar part of the quaternion, and q_v is the vector part (imaginary part) of the quaternion. When q_0 is zero, the quaternion is entirely imaginary and is called a pure quaternion. The imaginary numbers follow the right hand rule such that

$$ij = k \, , \quad ji = -k \, , \quad jk = i \, , \quad kj = -i \, , \quad ki = j \, , \quad ik = -j \, .$$

Also, they follow imaginary number rules such that

$$i^2 = -1 \, , \quad j^2 = -1 \, , \quad k^2 = -1 \, .$$

In addition, quaternions follow the basic rules of vector addition and multiplication. Let two quaternions *p* and *q* be

$$p = p_0 + p_1 i + p_2 j + p_3 k \, , \tag{A-3}$$

$$q = q_0 + q_1 i + q_2 j + q_3 k \ . \tag{A-4}$$

Then, the sum of the two quaternions becomes

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k \, , \tag{A-5}$$

and the multiplication becomes

$$\begin{aligned} pq &= (p_0 + p_1 i + p_2 j + p_3 k)(q_0 + q_1 i + q_2 j + q_3 k) \\ &= (p_0 q_0 + p_1 q_0 i + p_2 q_0 j + p_3 q_0 k) + (p_0 q_1 i + p_1 q_1 i^2 + p_2 q_1 ij + p_3 q_1 ik) \\ &\quad + (p_0 q_2 j + p_1 q_2 ji + p_2 q_2 j^2 + p_3 q_2 jk) + (p_0 q_3 k + p_1 q_3 ki + p_2 q_3 kj + p_3 q_3 k^2) \, . \\ &= (p_0 q_0 - p_1 q_1 - p_2 q_2 - p_3 q_3) + (p_0 q_1 + p_1 q_0 + p_2 q_3 - p_3 q_2)i \\ &\quad + (p_0 q_2 + p_2 q_0 + p_3 q_1 - p_1 q_3)j + (p_0 q_3 + p_3 q_0 + p_3 q_3 - p_3 q_3)k \end{aligned} \tag{A-6}$$

As it can be seen by inspection, commutivity is preserved for addition but is not preserved for multiplication. The conjugate of quaternions *q*, denoted as *q**, is given by

$$q^* = q_0 - q_v = q_0 - q_1 i - q_2 j - q_3 k \ . \qquad \text{(A-7)}$$

In order to use quaternions to determine the orientation in 3D vector space, the real part of the quaternion must be zero (i.e., $v = 0 + v_1 i + v_2 j + v_3 k$). This also means that the 3D vector *v* is a pure quaternion once rotated by quaternion *q*. Let a quaternion *q* represent a four dimensional rotation vector, and a vector *v* represent a vector in a three dimensional space. Then, the quaternion after rotation becomes

$$\begin{aligned} q \cdot v &= (q_0 + q_v)(0 + v_v) \\ &= q_v \cdot v_v + q_0 \cdot v_v + q_v \times v_v \, . \end{aligned} \qquad \text{(A-8)}$$

Eq. (A-8) cannot be a pure quaternion unless $q_v \cdot v_v = 0$. Hence a single multiplication of a quaternion does not yield a three dimensional vector. If a 3D vector *v* is multiplied by two quaternions, *p* and *q*, the vector *v* after multiplication is written as

$$q \cdot v \cdot p = (q_0 + q_v)(0 + v_v)(p_0 + p_v) \ . \qquad \text{(A-9)}$$

The real part of (A-9) has to be zero in order to make the multiplication a pure quaternion. The real part calculation is expressed as

$$- p_0(q_v \cdot v_v) - q_0(v_v \cdot p_v) + (q_v \times p_v) \cdot v_v = 0 \, . \qquad \text{(A-10)}$$

If $p_0 = q_0$, (A-10) can be simplified as

$$- p_0(q_v + p_v) \cdot v_v + (q_v \times p_v) \cdot v_v = 0 \, . \qquad \text{(A-11)}$$

Eq. (A-11) is always true if $p_v = -q_v$. Therefore, the quaternion *p* becomes

$$p = p_0 + p_v = q_0 - q_v = q^* . \qquad \text{(A-12)}$$

Then, the vector *v* after rotation becomes

$$\begin{aligned} &(q_0 + q_1 i + q_2 j + q_3 k)(0 + xi + yj + zk)(q_0 - q_1 i - q_2 j - q_3 k) \\ &\qquad = 0 \\ &\qquad + \{(q_0{}^2 + q_1{}^2 + q_2{}^2 + q_3{}^2)x + 2(q_1 q_2 - q_0 q_3)y + 2(q_1 q_3 + q_0 q_2)z\}i \\ &\qquad + \{2(q_1 q_2 + q_0 q_3)x + (q_0{}^2 - q_1{}^2 + q_2{}^2 - q_3{}^2)y + 2(q_2 q_3 - q_0 q_1)z\}j \ . \\ &\qquad + \{2(q_1 q_3 - q_0 q_2)x + 2(q_2 q_3 + q_0 q_1)y + (q_0{}^2 - q_1{}^2 - q_2{}^2 + q_3{}^2)z\}k \end{aligned} \qquad \text{(A-13)}$$

This can be written in the matrix form as follows:

$$qvq^* = Qv\,, \tag{A-14}$$

where $Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & q_0{}^2 + q_1{}^2 - q_2{}^2 - q_3{}^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 0 & 2(q_1q_2 + q_0q_3) & q_0{}^2 - q_1{}^2 + q_2{}^2 - q_3{}^2 & 2(q_2q_3 - q_0q_1) \\ 0 & 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0{}^2 - q_1{}^2 - q_2{}^2 + q_3{}^2 \end{bmatrix}$ and $v = \begin{bmatrix} 0 \\ x \\ y \\ z \end{bmatrix}$.

When a quaternion q is a unity vector, it satisfies

$$q_0{}^2 + |q_v|^2 = 1\,. \tag{A-15}$$

Then, (A-15) can be rewritten as

$$\cos^2\theta + \sin^2\theta = 1\,, \tag{A-16}$$

where $\cos^2\theta = q_0{}^2$ and $\sin^2\theta = |q_v|^2$

A vector u is defined as a unit vector parallel to the vector part of quaternion q as

$$u = \frac{q_v}{|q_v|} = \frac{q_v}{\sin\theta}\,. \tag{A-17}$$

Then, the unit quaternion q is

$$q = q_0 + q_v = \cos\theta + u\sin\theta\,. \tag{A-18}$$

Then, qvq^* can be written as

$$\begin{aligned} qvq^* &= (q_0 + q_v)(0 + v_v)(q_0 - q_v) \\ &= (q_0{}^2 - |q_v|^2)v_v + 2(q_v \cdot v_v)q_v + 2q_0(q_v \times v_v)\,. \end{aligned} \tag{A-19}$$

Expressing v in terms of v_p (parallel to q_v) and v_n (normal to q_v) yields the following:

$$\begin{aligned} q\cdot v\cdot q^* &= (q_0{}^2 - |q_v|^2)(v_p + v_n) + 2(q_v \cdot (v_p + v_n))q_v + 2q_0(q_v \times (v_p + v_n)) \\ &= (q_0{}^2 - |q_v|^2)v_p + 2(q_v \cdot v_p)q_v + 2q_0(q_v \times v_p) + (q_0{}^2 - |q_v|^2)v_n + 2(q_v \cdot v_n)q_v + 2q_0(q_v \times v_n) \\ &= (q_0{}^2 - |q_v|^2 + 2|q_v|^2)v_p + (q_0{}^2 - |q_v|^2)v_n + 2q_0|q_v|(u \times v_n) \qquad . \\ &= v_p + (\sin^2\theta - \cos^2\theta)v_n + 2\cos\theta\sin\theta(u \times v_n) \\ &= v_p + \cos(2\theta)v_n + \sin(2\theta)(u \times v_n) \end{aligned}$$

$$\tag{A-20}$$

Therefore, vector *v* after rotation is calculated by

$$qvq^* = v_p + \cos(2\theta)v_n + \sin(2\theta)(u \times v_n), \tag{A-21}$$

where *u* is a unit vector of q_v. Consequently, $u \times v_n$ is perpendicular to both q_v and v_n.

The relationship between arbitrary vector *v* and *qvq** is illustrated by the geometry in Figure A-1 and Figure A-2. Eq. (A-21) illustrates that vector *v* is rotated by angle 2θ because the 3D vector *v* is rotated by θ once about *q* and once about *q**. In order to rotate *v* by θ, the rotation of quaternion *q* needs to be halved. In other words,

$$q = q_0 + q_v = \cos\frac{\theta}{2} + u\sin\frac{\theta}{2} = \cos\frac{\theta}{2} + (u_x i + u_y j + u_z k)\sin\frac{\theta}{2} \tag{A-22}$$

Therefore, each component of the quaternion *q* should be written as follows:

$$\begin{aligned} q_0 &= \cos\frac{\theta}{2} \\ q_1 &= \frac{u_x}{|u|}\sin\frac{\theta}{2} \\ q_2 &= \frac{u_y}{|u|}\sin\frac{\theta}{2} \\ q_3 &= \frac{u_z}{|u|}\sin\frac{\theta}{2} \end{aligned} \tag{A-23}$$

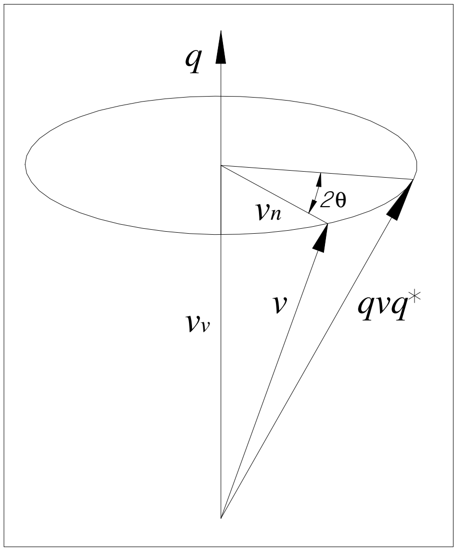


Figure A-1: Rotation of vector *v* by the angle of 2θ about vector *q*.

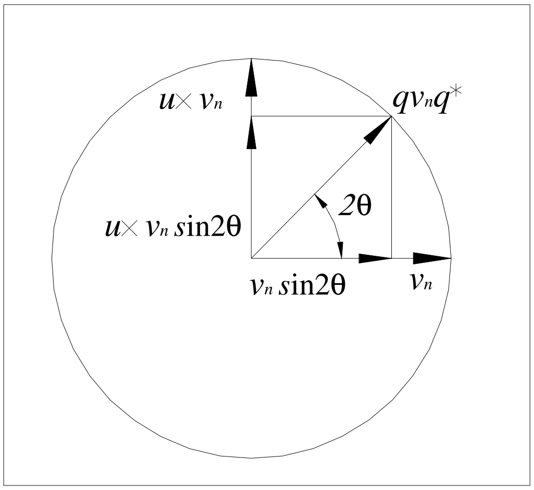


Figure A-2: Rotated vector component v_n before and after rotation.

Appendix B: Gyro Calibration

For gyro calibration, the method presented in [16] is adapted because this method does not require any additional sensor and the procedure is simple enough to be used in automotive industries. In this method, a gyro is modeled as follows:

$$\omega\_meas_{axis} = GG_{axis} \cdot \omega_{axis} + GB_{axis} + GAS_{axis} \cdot A_{axis}\,, \tag{B-1}$$

where ω_meas_{axis} is the gyro measurement of each axis, GG_{axis} is the gyro gain of each axis, ω_{axis} is the true angular velocity, and GB_{axis} is the gyro bias of each axis, and GAS_{axis} is the linear acceleration sensitivity of each axis.

The linear acceleration sensitivity can be modeled as linear. When the sensor is stationary, the angular velocity is zero, and the gyro output is equal to the gyro biases plus acceleration sensitivity multiplied by the acceleration. When the accelerometer is calibrated, the IMU is placed stationary in six tilt angles for at least two seconds. After accelerometer calibration, the six tilt angles can be calculated. Then, the average acceleration that the gyro axis is subjected to during the stationary state can be calculated. By using the two average accelerations from the six tilt angles, the linear acceleration sensitivity is calculated as

$$GAS_{axis} = \frac{\omega\_A1_{axis} - \omega\_A2_{axis}}{Avg\_A1 - Avg\_A2}\,, \tag{B-2}$$

where Avg_A1 is the first average acceleration, ω_A1_{axis} is the corresponding gyro output of the first acceleration, and Avg_A2 is the minimum calculated acceleration and $\omega\_A\min_{axis}$ is the corresponding gyro output. When the acceleration sensitivity is calculated, the gyro biases can be calculated by substituting the gyro acceleration sensitivity and the subjected acceleration in (3-19).

In order to calculate the gyro gain, the angular velocity should be measured. In this method, instead of utilizing an additional sensor to obtain the angular velocity, the integration of the angular velocity is used. By integrating both sides of (B-1),

$$\int (\omega\_meas_{axis} - GB_{axis} - GAS_{axis} \cdot A_{axis}) = \int GG_{axis} \cdot \omega_{axis} = GG_{axis} \cdot \int \omega_{axis}\,. \tag{B-3}$$

Since the gyro measurement, gyro bias, acceleration and gyro acceleration sensitivity are known, the left side of (B-3) is known. When the final orientation is Θ, (B-3) is written as

$$\int(\omega\_meas_{axis} - GB_{axis} - GAS_{axis} \cdot A_{axis}) = GG_{axis} \cdot \int \omega_{axis} = GG_{axis} \cdot \Theta \,. \tag{B-4}$$

Then, the gain of the gyro can be calculated as

$$GG_{axis} = \frac{\int(\omega\_meas_{axis} - GB_{axis} - GAS_{axis} \cdot A_{axis})}{\Theta} \,. \tag{B-5}$$

Equation (3-4) is true only when the sensor is rotated in one axis. Therefore, when a gyro is calibrated, the sensor should be placed on a smooth surface, and one side of the sensor should be placed on a reference block as shown in Figure B-1. Then, the sensor is rotated 360º so that the same sensor side faces the reference block after its rotation. Then, the final orientation (Θ) becomes 360º. Detailed theory and test results of gyro calibration are reported in [16].

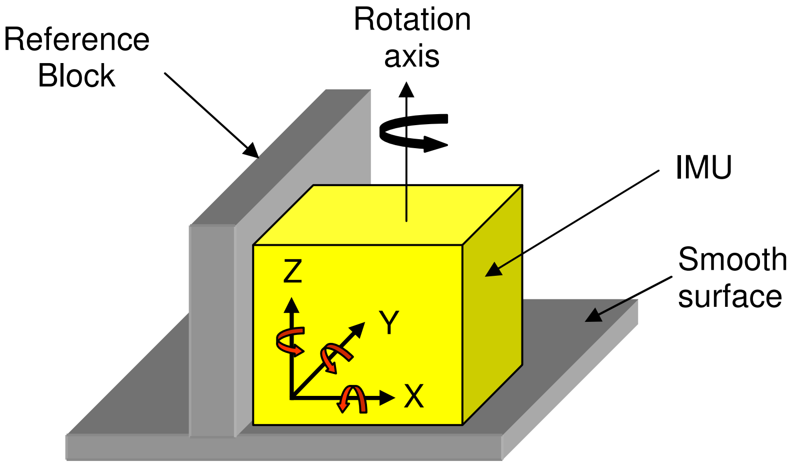


Figure B-3: The calibration of z-axis gyro using a reference block.

Bibliography

[1] X. Yun , E. R. Bachmann , R. B. McGhee , R. H. Whalen , R. L. Roberts , R. H. Knapp , A. J. Healey, and M. J. Zyda “Testing and Evaluation of an Integrated GPS/INS System for Small AUV Navigation,” IEEE Journal of Oceanic. Engineering, Vol. 24, Jul. 1999, pp. 396-404.

[2] K. Kobayashi, K. C. Cheok, K. Watanabe, and F. Munekata, “Accurate Differential Global Positioning System via Fuzzy Logic Kalman Filter Sensor Fusion Technique”, IEEE Transactions on Industrial Electronics, Vol. 45 No. 3, June, 1998, pp.510-518.

[3] G. Hyslop, D. Gerth and J. Kraemer, “GPS/INS Integration on the Standoff Land Attack Missile (SLAM)”, IEEE AES Magazine, Jul. 1990, pp. 29–34.

[4] N. Parnian and M. F. Golnaraghi, “Integration of Vision and Inertial Sensors for Industrial Tools Tracking”, Sensor Review, Vol. 27, No. 2, 2007, pp.132-141.

[5] R. Oktem, E. Aydin, and N. Cagiltay, “An Indoor Navigation Aid Designed for Visually Impaired People”, 34th Annual Conference of the IEEE Industrial Electronics Society, IECON 2008, 10-13 November 2008, Orlando, Florida, USA. pp. 2982 – 2987.

[6] Jing Yang, Eun-Seok Choi, Wook Chang, Won-Chul Bang, Sung-Jung Cho, Jong-Koo Oh, Joon-Kee Cho, and Dong-Yoon Kim “A Novel Hand Gesture Input Device Based on Inertial Sensing Technique”, IECON 2004, 2-6 November 2004, Busan, Korea. pp. 2786 – 2791.

[7] Y. Tao, H. Hu, and H. Zhou, “Integration of Vision and Inertial Sensors for 3D ARM Motion Tracking in Home-based Rehabilitation,” International Journal of Robotics Research, Vol. 26, No. 6, 2007, pp. 607-624.

[8] “Company website - PINpoint Information Systems”, [Online]. Available: http://www. pinpointinfo.com/ [Accessed Feb. 17, 2010].

[9] “Company website - peeperl+Fuchs”, [Online]. Available: www.am.pepperl-fuchs.com /products/productfamily.jsp?productfamily_id=1730 [Accessed Feb. 17, 2010].

[10] Georg Ogris, Thomas Stiefmeier, Holger Junker, Paul Lukowicz, and Gerhard Tröster, "Using Ultrasonic Hand Tracking to Augment Motion Analysis Based Recognition of Manipulative Gestures", in Proceedings of the 9th IEEE International Symposium on Wearable Computers, Osaka, Japan, October 18-21, 2005 .

[11] E. Foxlin, “Extended Draft Version of Chapter 8 in Handbook of Virtual Environment Technology”, Kay Stanney, Editor, Lawrence Erlbaum Associates, 2002.

[12] Walid Abdel-Hamid, “Accuracy Enhancement of Integrated MEMS-IMU/GPS Systems for Land Vehicular Navigation Applications”, PhD Thesis, University of Calgary, Calgary, AB, Canada, January 2005.

[13] P. H. LaFond, “Modeling for Error Reduction Accelerometers”, in Proceedings of IEEE PLANS 1992. Mar. 23-27, 1992, pp. 126-132.

[14] H. J. Luinge and P. H. Veltink, "Inclination Measurement of Human movement using a 3-D Accelerometer with Autocalibration", IEEE Transactions on Neural Systems and Rehabilitation Engineering 12, 2004, pp. 112-121.

[15] "3DM-GX1 Sensor Performance Improvement with Temperature Compensation", Microstrain, [Online]. Available: http://www.microstrain.com/pdf/Comparision%20With%20and%20Without%20Temp%20Comp.pdf [Accessed Feb.17, 2010].

[16] F. Ferraris, U. Grimaldi, and M. Parvis, "Procedure for Effortless In-field Calibration of Three-axis Rate Gyros and Accelerometers", Sensors and Materials, 1995, Vol. 7, No. 5, pp. 311-330.

[17] D. Giansanti, G. Maccioni, V. Macellari, "Guidlines for Calibration and Drift Compensation of a Wearable Device with Rate-Gyroscopes and Accelerometers", in Proceedings of the 29th Annual International Conference of the IEEE EMBS, Lyon, France, Aug. 22-26, 2007, pp. 2342-2345.

[18] P. Lang and A. Pinz, "Calibration of hybrid vision / inertial tracking systems"' in Proceedings of 2nd InerVis Workshop on Integration of Vision and inertial Sensors, Apr. 18, 2005.

[19] Akira Umeda, Mike Onoe, Kohji Sakata, Takehiro Fukushia, Kouichi Kanari, Hiroshi Iioka, and Toshiyuki Kobayashi, "Calibration of Three-axis Accelerometers using a Three-dimensional Vibration Generator and Three Laser Interferometers", Sensors and Actuators A, 114, 2004, pp. 93–101.

[20] E. L. Renk, M. Rizzo, W. Collins, F. Lee, and D. S. Bernstein, "Calibrating a Triaxial Accelerometer-magnetometer - Using Robotic Actuation for Sensor Reorientation During Data Collection", IEEE Control Systems Magazine, Vol. 25, No. 6, 2005, pp. 86–95.

[21] A. Kim and M. F. Golnaraghi, "Initial Calibration of an Inertial Measurement Unit Using an Optical Position Tracking System", Proceedings of the PLANS 2004, IEEE Position Location and Navigation Symposium, April 26–29, 2004, pp. 96-101.

[22] Zhuxin Dong, Guanglie Zhang, Yilun Luo, Chi Chiu Tsang, Guangyi Shi, Sze Yin Kwok, Wen J. Li, Philip H. W. Leong, and Ming Yiu Wong, "A Calibration Method for MEMS Inertial Sensors Based on Optical Tracking", in Proceedings of 2007 IEEE International Conference on Nano/Micro Engineered and Molecular Systems (IEEE-NEMS2007), Bangkok, Thailand, Jan. 16-19, 2007, pp. 542-547.

[23] H. J. Luinge and P. H. Veltink, "Inclination Measurement of Human Movement Using a 3-D Accelerometer with Autocalibration", IEEE Transactions on Neural Systems and Rehabilitation Engineering 12, 2004, pp. 112-121.

[24] E-H. Shin and N. El-Sheimy, "A New Calibration Method for Strapdown Inertial Navigation Systems", Zeitschrift fűr Vermessungswesen, Vol. 127, No. 1, Jan. 2002, pp. 41–50.

[25] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy, "A New Multi-position Calibration Method for MEMS Inertial Navigation Systems", Meas. Sci. Technol. Vol. 18, No. 7, July 2007, pp. 1897-1907.

[26] A. Lai, D. A. James, P. Hayes, and E. C. Harvey, "Semi-automatic Calibration Technique using Six Inertial Frames of Reference", in Proceedings of SPIE 5274, Perth, WA, Australia, Dec. 10-12, 2003, pp. 531-542.

[27] J. C. Lötters, J. Schipper , P. H. Veltink , W. Olthius, and P. Bergveld, “Procedure for In-use Calibration of Triaxial Accelerometers in Medical Applications”, Sensors Actuators A 68, 1998, pp. 221-228.

[28] Z. C. Wu, Z. F. Wang, and Y. Ge, “Gravity Based Online Calibration for Monolithic Triaxial Accelerometers' Gain and Offset Drift”, Proceedings of the 4th World Congress on Intelligent Control and Automation 3, Shanghai, China, Jun. 10-14, 2002, pp. 2171-2175.

[29] L. Ojeda, H. Chung, and J. Borenstein, “Precision-calibration of Fiber-optics Gyroscopes for Mobile Robot Navigation,” in Proceedings of 2000 IEEE Int. Conf. Robotics and Automation San Francisco, CA, Apr. 24–28, 2000, pp. 2064-2069.

[30] Kim, and M. F. Golnaraghi, “A Quaternion-Based Orientation Estimation Algorithm Using an Inertial Measurement Unit”, IEEE Position Location and Navigation Symposium 2004, Monterey, CA, April 26 - 29, 2004, pp. 268-272.

[31] H. Zhou, H. Hu, and Y. Tao, “Inertial Measurements of Upper Limb Motion,” Journal of Medicine, Biological Engineering and Computing, 2006, 44, pp. 479-487.

[32] H. J. Luinge P. H. Veltink, “Measuring Orientation of Human Body Segments Using Miniature Gyroscopes and Accelerometers”, Medical & Biological Engineering & Computing 2005, Vol. 43, pp. 273-282.

[33] M. Koifman and I. Y. Bar-Itzhack, “Inertial Navigation System Aided by Aircraft Dynamics”, IEEE Transactions on Control Systems Technology, Vol. 7, No. 4, July 1999, pp. 487 – 493.

[34] W. Abdel-Hamid, A. Noureldin, and N. El-Sheimy, "Adaptive Fuzzy Prediction of Low-Cost Inertial-Based Positioning Errors", IEEE Transactions on Fuzzy Systems, Vol. 15, No. 3, Jun. 2007, pp. 519-529.

[35] R. Rahbari, B. W. Leach, J. Dillon, and C. W. de Silva, “Expert System for an INS/DGPS Integrated Navigator Installed in a Bell 206 Helicopter”, IEEE Aerospace Conference, Big Sky, MT, March 6–13, 2004 pp. 1569-1579.

[36] Lovren, N.; Pieper, J.K. “Error Analysis of Direction Cosines and Quaternion Parameters Techniques for Aircraft Attitude Determination”, IEEE Transaction on Aerospace And Electronic Systems, Vol. 34, No. 3, July 1998, pp. 983 – 989.

[37] Lekskulchai Pongsak, Masafumi Okada, Tetsuya Sinohara, and Yoshihiko Nakamura, “Attitude Estimation by Compensating Gravity Direction”, In Proceedings of the 21th Annual Conference of the Robotics Society in Japan, Sep. 20-23, 2003, 2A23.

[38] H. J. Luinge and P. H. Veltink, “Measuring Orientation of Human Body Segments Using Miniature Gyroscopes and Accelerometers”, Medical & Biological Engineering & Computing 2005, Vol. 43, pp. 273-282.

[39] H. J. Luinge, P.H. Veltink, C.T.M. Baten, “Ambulatory Measurement of Arm Orientation” Journal of Biomechanics, Vol. 40, No. 1, 2007, pp. 78-85.

[40] H. Rehbinder and X. Hu, “Drift-free Attitude Estimation for Accelerated Rigid Bodies”, Automatica 40, 2004, pp. 653-659.

[41] Ashutosh Saxena, Gaurav Gupta, Vadim Gerasimov, and Sébastien Ourselin, “In Use Parameter Estimation of Inertial Sensors by Detecting Multilevel Quasi-static States”, In KES2005, Melbourne, Australia, September 2005, pp. 595-601.

[42] J-H Wang, and Y. Gao, “Multi-sensor Data Fusion for Land Vehicle Attitude Estimation using Fuzzy Expert System”, Data Science Journal. Vol. 4, 2005, pp.127-139.

[43] L. Ojeda and J. Borenstein, “FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain.” In Proceedings of the 2002 IEEE Int. Conference on Robotics and Automation, May 11-15, 2002, Washington DC, USA, pp. 317-322.

[44] D. Roetenberg, P. J. Slycke, and P. H. Veltink, “Ambulatory Position and Orientation Tracking Fusing Magnetic and Inertial Sensing”, Biomedical engineering, Vol. 54 No. 5, 2007, pp. 883-890.

[45] M. Sabatini, “Quaternion-based Extended KF for Determining Orientation by Inertial and Magnetic Sensing,” Biomedical Engineering, Vol. 53, No. 7, 2006, pp. 1346–1356.

[46] J. Hummel, M. Figl, C. Kollmann, H. Bergmann, and W. Birkfellner “Evaluation of a Miniature Electromagnetic, Position Tracker” Medical Physics Vol. 29, No. 10, 2002, pp. 2205–2212.

[47] E. R. Bachmann, X. Yun, C. W. Peterson, “An Investigation of the Effects of Magnetic Variations on Inertial/Magnetic Orientation Sensors”, in Proceedings of ICRA 2004, Apr. 26-May 1, 2004, New Orleans, LA, pp. 1115-1122.

[48] Daniel Roetenberg, Henk J. Luinge, Chris T. M. Baten, and Peter H. Veltink, “Compensation of Magnetic Disturbances Improves Inertial and Magnetic Sensing of Human Body Segment Orientation”, IEEE Trans on Neural Systems and rehabilitation engineering, Vol. 13 No. 3, 2005, pp. 395- 405.

[49] Daniel Roetenberg, Henk Luinge, and Peter Veltink, “Inertial and Magnetic Sensing of Human Movement Near Ferromagnetic Materials” in Proceedings of 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR ’03, Tokyo, Japan, Oct. 7-10, 2003, pp. 268-269.

[50] C. T. M. Baten, H. J. Luinge, and H. V. Moerkerk**,** “Estimating Body Segment Orientation Applying Inertial Sensing”, Neural Systems and Rehabilitation Engineering, Vol. 15, No. 3, pp. 469-471.

[51] “Firefly Prototype Specifications”, Firefly Motion Capture System, [Online]. Available: http://www.cybernet.com/interactive [Accessed: Feb. 17, 2010]

[52] R. A. States and E. Pappas, “Precision and Repeatability of the Optotrak 3020 Motion Measurement System”, Journal of Medical Engineering & Technology, Vol. 30, No. 1, Jan. 2006, pp.11 – 16.

[53] Lorin P. Maletsky, Junyi Sun, and Nicholas A. Morton, “Accuracy of an Optical Active-marker System to Track the Relative Motion of Rigid Bodies”, Journal of Biomechanics, Vol. 40, No. 3, 2007. pp. 682-685.

[54] A. D. Wiles, D.G. Thompson, and D.D Frantz, “Accuracy Assessment and Interpretation for Optical Tracking Systems”, In Proceedings of SPIE 5367, Medical Imaging, San Diego, CA, United States, 2004, pp. 421-432

[55] N. Parnian, “Integration of Local Positioning System & Strapdown Inertial Navigation System for Hand-Held Tool Tracking”, Ph.D. Thesis, University of Waterloo, 2008

[56] M. Ribo, M. Brandner, and A. Pinz. “A Flexible Software Architecture for Hybrid Tracking”, Journal of Robotics Systems”, Vol. 21, No. 2, Feb. 2004, pp.53-62.

[57] Eric Foxlin and Leonid Naimark, “Miniaturization, Calibration & Accuracy Evaluation of a Hybrid Self-Tracker”, Proc. of 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR ’03, Tokyo, Japan, Oct. 7-10, 2003, pp. 151-160.

[58] D. Roetenberg, P. Slycke, A. Ventevogel, and P.H. Veltink, “A Portable Magnetic Position and Orientation tracker”, Sensors and Actuators A, Vol. 135, No. 2, Apr. 2007, pp. 426–432.

[59] Mason, A. Shaw, A. I. Al-Shamma'a, “Intelligent Radio Frequency Identification Positioning using Wireless Sensor Networks”, Antennas and Propagation Conference, April 2-3, 2007, Loughborough, UK. pp. 145-148.

[60] Andreas Stelzer, Alexander Fischer, Franz Weinberger, and Martin Vossiek, “RF-Sensor for a Local Position Measurement System”, in Proceedings of SPIE, Vol. 5048, 2003, pp. 136-144.

[61] Paramvir Bahl and Venkata N. Padmanabhan, “RADAR: An In-Building RF-based User Location and Tracking System” in Proceedings of IEEE Infocom 2000, Tel-Aviv, Israel, March 2000, pp 775-784.

[62] K. Yu and I. Oppermann, “Performance of UWB Position Estimation Based on TOA Measurements”, in Proceedings of the Joint UWBST and IWUWBS, Kyoto, Japan, 2004, pp. 400-404.

[63] James F. O'Brien, Robert E. Bodenheimer, Gabriel J. Brostow, and Jessica K. Hodgins, “Automatic Joint Parameter Estimation from Magnetic Motion Capture Data”, in Proceedings of Graphics Interface, Montreal, Quebec, Canada, May 15-17, 2000, pp. 53-60.

[64] J. Hummel, M. Figl, C. Kollmann, and H. Bergmann, “Evaluation of a Miniature Electromagnetic Position Tracker”, Medical Physics, Vol. 29, No. 10, October 2002, pp. 2205–2212.

[65] Suzanne LaScalza, Jane Arico, and Richard Hughes, “Effect of Metal and Sampling Rate on Accuracy of Flock of Birds Electromagnetic Tracking System”, Journal of Biomechanics, Vol. 36, No. 1, 2003, pp. 141–144.

[66] Nirupama Bulusu, John Heidemann, and Deborah Estrin. “GPS-less Low Cost Outdoor Localization for Very Small Devices”, IEEE Personal Communications Magazine, Vol. 7, No. 5, Oct. 2000, pp. 28-34.

[67] A. Mason, A. Shaw, and A. I. Al-Shamma’a, “Intelligent Radio Frequency Identification Positioning Using Wireless Sensor Networks”, IEEE Antenna and Propagation Conference, Loughborough, UK, 2-3 April, 2007 pp. 145-148.

[68] Robert J. Fontana, “Recent Applications of Ultra Wideband Radar and Communications Systems” in Ultra-Wideband, Short-Pulse Electromagnetics 5, Kluwer Academic/Plenum Publishers, New York, 2002. pp. 225-234.

[69] Volker Schwarz, Alex Huber, and Michael Tüchler, “Accuracy of a Commercial UWB 3D Location/Tracking System and its Impact on LT Application Scenarios”, IEEE international Conference on Ultra-wideband, Sept 5-8, 2005, pp 599 – 603.

[70] P. Arcara and C. Melchiorri, “3D Position Measurement Based on Force Sensors for a One-Wire Haptic Interface”, 16th IEEE Instrumentation and Measurement Technology Conference, Venice, Italy, May 24-26, 1999, Vol. 2, pp. 1272-1277.

[71] “Space Sensor: 3D Sensing with Ease and Affordability”, Space Age Control Inc. [Online]. Available: http://www.spaceagecontrol.com/Main/3DDisplacementTransducer [Accessed Jan. 28, 2010].

[72] D. Lin, L. Keck Voon, H. Guo Rong, and N. Nagarajan, “GPS-based Attitude Determination for Microsatellite Using Three-Antenna Technology”, in Proceedings of Aerospace Conference, March 6-13, 2004, Vol. 2, pp. 1024-1029.

[73] Lu, G., M. E. Cannon and G. Lachapelle, “Attitude Determination in a Survey Launch Using Multi-antenna GPS Technologies”, in Proceedings of the ION 1993 National Technical Meeting, San Francisco, Jan. 20-22, 1993, pp. 251-260.

[74] E. Cattrysse, S. Provyn, P. Kool, O. Gagey, J. Clarys, P. and Van Roy, “Reproducibility of Kinematic Motion Coupling Parameters During Manual Upper Cervical Axial Rotation Mobilization: A 3-dimensional in Vitro Study of the Atlanto-axial Joint”, Journal of Electromyography and Kinesiology, Vol. 19, No. 1, 2009, pp. 93-104.

[75] Nikolaos Strimpakos, Vasiliki Sakellari, Georgios Gioftsos, Eleni Kapreli, and Jacqueline Oldham, “Cervical Joint Position Sense: an Intra- and Inter-examiner Reliability Study,” Gait & Posture, Vol. 23, 2006, pp. 22-31.

[76] Wolf, R., G. W. Hein, B. Eissfeller, and Loehnert, “An Integrated Low Cost GPS/INS Attitude Determination and Position Location System,” in Proceedings of ION GPS-96 Meeting, Kansas City, Missouri, 1996, pp. 975–981.

[77] Eric Foxlin, Michael Harrington, and Yury Altshuler, “Miniature 6-DOF Inertial System for Tracking HMDs”, in Proceedings of SPIE Vol. 3362, the AeroSense 1998 Conference on Helmet- and Head-Mounted Displays III, Orlando, FL, 1998, pp. 214-228.

[78] Y. C. Ho and R. C. K. Lee, “A Bayesian Approach to Problems in Stochastic Estimation and Control,” IEEE Trans. Automatic Control, Vol. 9, No. 4, Oct. 1964, pp. 333-339.

[79] D. Fox, J. Hightower, L. Liao, D. Schulz, G. Borriello, "Bayesian Filters for Location Estimation", IEEE Pervasive Computing Magazine, 2003, p. 24-33.

[80] B. M. Scherzinger, J. J. Hutton, and J. C. McMillan, "Low Cost Inertial/GPS Integrated Position and Orientation System for Marine Applications," IEEE Aerospace and Electronic Systems Magazine, Vol. 22, No. 12, May 1997, pp. 15-19.

[81] A. Noureldin, R. Sharaf, A.H. Osman, and N. El-Sheimy, “*INS/GPS Data Fusion Technique Utilizing Radial Basis Functions Neural Networks*” in Proceedings of the IEEE PLANS, Monterey, CA, April 2004, pp. 280-284.

[82] K. Touil, M. Zribi, J. B. Choquel, and M. Benjelloun, “Bayesian Bootstrap Filter for integrated GPS and Dead Reckoning Positioning”, IEEE, International Symposium on Industrial Electronics, ISIE07, Spain, 2007, pp. 1520-1524.

[83] Ali Asadian, Behzad Moshiri, and Ali Khaki Sedigh, "A Novel Data Fusion Approach in an Integrated GPS/INS System Using Adaptive Fuzzy Particle Filter", 5th International Conference on Technology and Automation, Oct. 15-16, 2005, Thessaloniki, Greece. pp. 125-130.

[84] Nguyen Ho Quoc Phuong, Hee-Jun Kang, Young-Soo Suh, Young-Shick Ro, and Kyu-Chan Lee, “A GPS/INS Integration System with New Orientation Measurement Scheme for Land Vehicle Application”, in Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics, Nov. 6-10, 2006, Paris, France, pp. 3099-3104.

[85] Pifu Zhang, Jason Gu, Evangelos E. Milios, and Peter Huynh, ``Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter", in Proceedings of IEEE International Conference on Mechatronics and Automation, Jul. 29 – Aug. 1, 2005, Niagara Falls, Ontario , Canada, pp. 1497 – 1502.

[86] E. Foxlin and L. Naimark “VIS-Tracker: A Wearable Vision-Inertial Self-Tracker” Proceedings of IEEE Virtual Reality, Los Angeles, USA, March 22-26, 2003, pp. 199-206.

[87] L. Naimark and E. Foxlin, “Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-inertial self-tracker”, in Proceedings of IEEE International Symposium on Mixed and Augmented Reality, Sep. 30 – Oct. 1, Darmstadt, Germany, 2002, pp. 27–36.

[88] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, P. Nordlund, "Particle Filters for Positioning, Navigation, and Tracking," IEEE Transactions on Signal Processing, Vol. 50, No. 2, 2002, pp. 425-437.

[89] N. Yang, W. F. Tian, Z. H. Jin, and C. B. Zhang, “Particle Filter for Sensor Fusion in a Land Vehicle Navigation System,” Measurement science and technology, Vol. 16, 2004, pp. 677-681.

[90] P-J. Nordlund and F. Gustafsson. “Sequential Monte Carlo Filtering Techniques Applied to Integrated Navigation Systems”, in Proceedings of the 2001American Control Conference, Vol. 6, 2001, pp. 4375-4380.

[91] R. E. Kalman, “A New Approach to Linear Filtering and Prediction Problems”, Journal of Basic Engineering, Vol. 82 D, 1960, pp. 35–45.

[92] S. Thrun, W, Burgard, and D. Fox, *Probabilistic Robotics*, MIT Press, Cambridge, MA, 2005.

[93] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking”, IEEE Trans. on Signal Processing, Vol. 50, No. 2, February 2002, pp. 174-188.

[94] Changhe Yuan and Marek J. Druzdzel, "Theoretical Analysis and Practical Insights on Importance Sampling in Bayesian Networks", International Journal of Approx. Reasoning 46, 2007, pp. 320-333.

[95] Byoung-Tak Zhang, "A Bayesian Framework for Evolutionary Computation," Proc. of the 1999 Congress on Evolutionary Computation, Washington DC. USA, Jul. 6 -9, 1999, pp. 722-728.

[96] S. Park, J. Hwang, K. Rou, and E. Kim, "A New Particle Filter Inspired by Biological Evolution: Genetic Filter", in Proceedings of World Academy of Science, Engineering, and Technology, Vol. 21, Jan., 2007, pp. 459-463.

[97] R. Fung and K. C. Chang, "Weighting and Integrating Evidence for Stochastic Simulation in Bayesian Networks", in Proc. of the Fifth Conference on Uncertainty in Artificial Intelligence, Windsor, Ontario, Aug. 18-20, 1989, pp. 209-219.

[98] K. Kanazawa, D. Koller, and S. J. Russell, "Stochastic Simulation Algorithms for Dynamic Probabilistic Networks," in Proceedings of the Eleventh Annual Conference Uncertainty AI, 1995, pp. 346–351.

[99] Nelson F. Ford and Jarick Rager. "Expert System Support in the Textile Industry: End Product Production Planning Decisions", Expert Systems with Applications. Vol. 9, No. 2 1995, pp. 237-246.

[100] K. D. Schnelle and R. S. H. Mah, "A Real-time Expert System for Quality Control", IEEE Expert, October 1992, pp. 36-42.

[101] Alexander Prusak and Axel R. Hidde, "The Use of Knowledge-based Systems for PCB Testing and Repair", Expert Systems with Applications, Vol. 5, No. 1-2, 1992, pp. 141-151.

[102] Spengler, M. Stanton, and M. Rowlands, "Expert Systems and Quality Tools for Quality Improvement", in Proceedings of Seventh IEEE International Conference on Emerging Technologies and Automation, Vol. 2, October 19-21, 1999, Barcelona, Spain, pp. 955-962.

[103] L. A. Zadeh, "Fuzzy Sets", Information and Control Vol. 8, 1965, pp. 338-353.

[104] L. A. Zadeh, "Fuzzy Algorithms", Information and Control, Vol. 12, 1968, pp. 94-102.

[105] L. A. Zadeh, "Outline of a New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-3, No.1, 1973, pp. 28-44.

[106] Steven R. Swanson, "A fuzzy Navigation State Estimator for GPS/INS Integration", Plans 1998, Apr. 20-23, 1998, Houston, TX, USA, pp. 541-548.

[107] Walid Abdel-Hamid, Aboelmagd Noureldin, and Naser El-Sheimy "Adaptive Fuzzy Modeling of Low Cost Inertial Based Positioning Errors", Fuzzy Systems, Vol. 15, No. 3, 2007, pp. 519-529.

[108] Hiliuta, R. Jr. Landry and F. Gagnon, "Fuzzy Correction in a GPS/INS Hybrid Navigation System", IEEE Trans. Aerospace and Electronic Systems, Vol. 40, No. 2, 2004, pp. 591-600.

[109] Immanuel Ashokaraj, Antonios Tsourdos, Peter Silson, Brian White, John Economou, “Feature Based Robot Navigation: Using Fuzzy Logic and Interval Analysis”, FUZZ-IEEE July, 25-29, 2004 Budapest, Hungary pp. 1461-1466.

[110] “Canadian Gravity Standardization Network”, Natural Resources of Canada, [online]. Available: http://csrsjava.geod.nrcan.gc.ca/csrsjcpe/GSDreportsEN?user_name=CGIS2658&querytype=CGSN&key=93231965 [Accessed: Oct. 2, 2008].

[111] A. G. Ledroz, E. Pecht, D. Cramer, and M. P. Mintchev, “FOG-Based Navigation in Downhole Environment During Horizontal Drilling Utilizing a Complete Inertial Measurement Unit”, IEEE Trans Instrumentation and Measurement, Vol. 54 No.5, Oct. 2005, pp. 1997-2006.

[112] Y. Suh, “Development of an INS Integrated Autonomous Positioning System for Assisting Effective Fire-fighting Activity”, KSCE Journal of Civil Engineering, Vol. 8, No. 5, Sept. 2004, pp. 567-574

[113] L. Ojeda and J. Borenstein, "Personal Dead-reckoning System for GPS-denied Environments", I*EEE* International Workshop on Safety, Security, and Rescue Robotics (SSRR2007) in Rome, Italy, September 27-29, 2007, pp. 1-6.

[114] Y. Suh, “Development of an INS Integrated Autonomous Positioning System for Assisting Effective Fire-fighting Activity”, KSCE Journal of Civil Engineering, Vol. 8, No. 5, Sept. 2004, pp. 567-574.

[115] J. C. K. Chou, “Quaternion Kinematic and Dynamic Differential Equations”, IEEE Trans. Robot. Autom., Vol. 8, No. 1, Feb. 1992, pp. 53–64.

[116] D. H. Titterton and J. L. Weston, 1997, *Strapdown Inertial Navigation Technology*, Peter Peregrinus Ltd. United Kingdom.

[117] J. Huddle, “Trends in Inertial Systems Technology for High Accuracy AUV Navigation”, In Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles MA, USA, Aug. 20-21 1998, pp. 63-73.

[118] L.A. Zadeh, “Outline of a New Approach to the Analysis of Complex Systems and Decision Processes”, IEEE Transaction on Systems, Man, and Cybernetics, Vol. SMC-3, No.1, 1973, pp. 28-44

[119] L. A. Zadeh, “Similarity Relations and Fuzzy Orderings”, Inform. Sci., Vol. 3, 1971, pp. 177-200.