FINAL REPORT

Handheld UXO Sensor Improvements to Facilitate UXO/Clutter Discrimination Volume 2

SERDP Project MM-1381

NOVEMBER 2007

Dr. Thomas Bell
SAIC

Dr. Leslie Collins
Duke University

DoD EPA DOE SERDP

Strategic Environmental Research and Development Program

Signal Processing for Handheld UXO Sensor Improvements

Final Report
May 2007

Prepared by
Stacy L. Tantum and Leslie M. Collins
Duke University, Durham, NC

for
Nagi Khadr
SAIC – Advanced Sensors and Analysis Division, Arlington, VA

May 22, 2007

Contents

List of Figures **vii**

1 Introduction **1**

2 Iterative Integration Algorithm for Velocity Stabilization **2**
2.1 Angle (Attitude) Stabilization . . . 2
2.2 Velocity Stabilization . . . 3
2.3 Gain Calculation . . . 5
2.4 Experimental Results . . . 9
2.4.1 Sweeping Data . . . 9
2.4.2 Linear Data . . . 9
2.5 Summary . . . 16

3 Iterative Integration for Position Calculation **17**
3.1 Sensor Measurement Error Models . . . 17
3.2 Stabilized Angle Calculation . . . 18
3.3 Velocity Stabilization and Position Calculation . . . 18
3.4 Gain Calculation . . . 22
3.4.1 Fixed Gain Calculation . . . 22
3.4.2 Adaptive Gain Calculation . . . 22
3.5 Experimental Results . . . 23
3.5.1 Interpolation of Ground Truth Data . . . 23
3.5.2 Position Calculation . . . 24
3.6 Summary . . . 28

4 Positioning of Handheld UXO Sensor via ZUPT Algorithm **38**
4.1 Velocity calculation . . . 38
4.2 Position calculation . . . 40
4.3 Experimental Results . . . 41
4.3.1 Results of sweeping motion . . . 42
4.3.2 Results of linear motion . . . 47
4.4 Summary . . . 47

5 System Modeling for Improved Acceleration Estimation **51**
5.1 Parametric methods for system identification . . . 51
5.2 Model parameter estimation and its applications to acceleration enhancement . . . 52
5.3 Summary . . . 54

6 Board Data Processing (July 2006) **57**
6.1 Adaptive Error Mitigation Algorithms . . . 57
6.2 System Identification Algorithms . . . 73
6.3 Summary . . . 73

7 APG SAINT Data Processing (September 2006) 78
7.1 Adaptive Error Mitigation Algorithms 78
7.2 System Identification Algorithms 115
7.3 Summary 115

List of Figures

1 Body frame and earth frame geometry. (X,Y) represent the earth frame coordinates and (x',y') represent the body frame coordinates. . . . 2
2 Iterative integration correction algorithm for pitch stabilization. . . . 3
3 Iterative integration correction algorithm for velocity stabilization along the y' axis. . . . 4
4 Iterative integration correction algorithm for velocity stabilization along the x' axis. . . . 5
5 Example of the partition of the path along the x orientation. . . . 6
6 Example of the partition of the path along the y orientation. . . . 6
7 Example illustrating choosing K_x. . . . 7
8 Example illustrating choosing K_y. . . . 8
9 Stabilized velocities along the x axis with adaptive K and fixed K algorithms. . . . 10
10 Stabilized velocities along the y axis with adaptive K and fixed K algorithms. . . . 10
11 Stabilized positions along the x axis with adaptive K and fixed K algorithms. . . . 11
12 Stabilized positions along the y axis with adaptive K and fixed K algorithms. . . . 11
13 Errors in stabilized positions along the x axis with adaptive K and fixed K algorithms. . . . 12
14 Errors in stabilized positions along the y axis with adaptive K and fixed K algorithms. . . . 12
15 Calculated velocity along the x axis based on the linear data. . . . 13
16 Calculated velocity along the y axis based on the linear data. . . . 13
17 Calculated position along the x axis based on the linear data. . . . 14
18 Calculated position along the y axis based on the linear data. . . . 14
19 Estimation error of the position along the x axis based on the linear data. . . . 15
20 Estimation error of the position along the y axis based on the linear data. . . . 15
21 Iterative integration correction algorithm for pitch angle calculation. . . . 18
22 Geometry of earth frame and body frame. xoy is the coordinate system of the body frame and XOY is the coordinate system of the earth frame. . . . 19
23 Iterative integration correction algorithm for velocity stabilization along y axes. . . . 20
24 Iterative integration correction algorithm for velocity stabilization along x axes. . . . 21
25 Root of mean square error versus gains K_x and K_y for calculating position of hand-held UXO sensor system along X orientation. . . . 24
26 Top panel: Root of mean square error versus gain K_x for calculating position of hand-held UXO sensor system along X orientation. Bottom panel: Root of mean square error versus gain K_y for calculating position of hand-held UXO sensor system along X orientation. . . . 25
27 Root of mean square error versus gains K_x and K_y for calculating position of hand-held UXO sensor system along Y orientation. . . . 26
28 Top panel: Root of mean square error versus gain K_x for calculating position of hand-held UXO sensor system along Y orientation. Bottom panel: Root of mean square error versus gain K_y for calculating position of hand-held UXO sensor system along Y orientation. . . . 27
29 Calculated velocity and position along X orientation directly using the conventional integration method. . . . 28
30 Calculated velocity and position along Y orientation directly using the conventional integration method. . . . 29
31 Stabilized velocity along X orientation with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$. . . . 30

32 Calculated error of stabilized velocity along X orientation with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$. 30
33 Stabilized position along X axes with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$. 31
34 Calculated error of stabilized position along X axes with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$. 31
35 Stabilized velocity along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$. 32
36 Calculated error of stabilized velocity along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$. 32
37 Stabilized position along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$. 33
38 Calculated error of stabilized position along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$. 33
39 Stabilized velocity along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$. 34
40 Calculated error of stabilized velocity along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$. 34
41 Stabilized position along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$. 35
42 Calculated error of stabilized position along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$. 35
43 Stabilized velocity along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$. 36
44 Calculated error of stabilized velocity along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$. 36
45 Stabilized position along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$. 37
46 Calculated error of stabilized position along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$. 37
47 Comparison of zero velocity points in X and Y directions. Sweeping motion. 38
48 Yaw angular rate of linear motion of handheld UXO sensor. 38
49 Yaw angular rate of nonlinear motion of handheld UXO sensor. 39
50 Yaw angle calculation before de-trended and after de-trended. 39
51 Geometry of earth frame and body frame. *xoy* is the coordinate system of the body frame and XOY is the coordinate system of the earth frame. 40
52 Example of zero-position-update points. The solid lines are the actual track of the sensor. The dash lines are the coordinate axes of the earth frame. 40
53 Scheme for positioning the handheld UXO sensor. ZVP denotes zero-velocity points. ZPP is zero-position points. ZPPT represents the zero-position-update algorithm. . . . 41
54 Velocity calculated from raw acceleration by the conventional numerical integration. Sweeping motion. 42
55 Position calculated from raw acceleration by the conventional numerical integration. Sweeping motion. 42
56 Calculated velocity along X orientation based on estimated acceleration. Sweeping motion. 43
57 Calculated velocity along Y orientation based on estimated acceleration. Sweeping motion. 43

58 Calculated velocity along X orientation based on raw acceleration. Zero velocity points are accurately determined. Sweeping motion. . . . 43
59 Calculated velocity along Y orientation based on raw acceleration. Zero velocity points are accurately determined. Sweeping motion. . . . 43
60 Position along X orientation calculated from the estimated velocity by direct integration method. Sweeping motion. . . . 44
61 Position error along X orientation calculated from the estimated velocity by direct integration method. Sweeping motion. . . . 44
62 Position along X orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion. . . . 45
63 Position error along X orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion. . . . 45
64 Position along Y orientation calculated from the estimated velocity by direct integration method. Sweeping motion. . . . 45
65 Position error along Y orientation calculated from the estimated velocity by direct integration method. Sweeping motion. . . . 45
66 Position along Y orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion. . . . 46
67 Position error along Y orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion. . . . 46
68 Searching for zero velocity points by relying on acceleration in linear motion. The zero velocity points are the green circle points that are between the red star points. . . . 48
69 Example of zero-position updates for linear motion. The solid lines are the actual track of the sensor. The dash lines are the coordinate axes of the earth frame. . . . 48
70 Velocity calculated from raw acceleration by the conventional numerical integration. Linear motion. . . . 48
71 Position calculated from raw acceleration by the conventional numerical integration. Linear motion. . . . 48
72 Calculated velocity along X orientation based on estimated acceleration. Linear motion. 49
73 Calculated velocity along Y orientation based on estimated acceleration. Linear motion. 49
74 Calculated position along X orientation based on estimated acceleration. Linear motion. 49
75 Calculated position along Y orientation based on estimated acceleration. Linear motion. 49
76 Calculated position error along X orientation based on estimated acceleration. Linear motion. . . . 50
77 Calculated position error along Y orientation based on estimated acceleration. Linear motion. . . . 50
78 Black box structure. $u(t)$ is the input, $y(t)$ denotes the output and $e(t)$ represents the model error. . . . 51
79 Estimated acceleration along x orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 8$, $nb = 7$ and $nk = 2$. . . . 53
80 Estimated acceleration along y orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 7$, $nb = 6$ and $nk = 5$. . . . 53
81 Estimated acceleration along x orientation. a_{meas} is input and a_{true} is output. $na = 6$, $nb = 1$ and $nk = 1$. . . . 53
82 Estimated acceleration along y orientation. a_{meas} is input and a_{true} is output. $na = 4$, $nb = 6$ and $nk = 2$. . . . 53

83 Estimated acceleration along x orientation. a_{meas} is input and v_{true} is output. $na = 4$, $nb = 6$ and $nk = 2$. 54
84 Estimated acceleration along y orientation. a_{meas} is input and v_{true} is output. $na = 4$, $nb = 3$ and $nk = 1$. 54
85 Correlation analysis and spectral analysis for ARMAX model. 55
86 Estimated acceleration along x orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 5$, $nb = 8$, $nc = 5$ and $nk = 1$. 56
87 Estimated acceleration along y orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 11$, $nb = 11$, $nc = 11$, and $nk = 3$. 56
88 ArcSecond ground truth for data set 'AA'. 58
89 ArcSecond ground truth for data set 'BB'. 59
90 ArcSecond ground truth for data set 'CC'. 60
91 ArcSecond ground truth for data set 'DD'. 61
92 Measured accelerations after pre-processing. 62
93 Stabilized velocities after bias and trend removal. 63
94 Stabilized attitude angles (roll, pitch, yaw) after bias and trend removal. 64
95 Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'AA'. 65
96 Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'BB'. 66
97 Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'CC'. 67
98 Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'DD'. 68
99 Estimated positions (in Earth frame) and comparison to ground truth for data set 'AA'. 69
100 Estimated positions (in Earth frame) and comparison to ground truth for data set 'BB'. 70
101 Estimated positions (in Earth frame) and comparison to ground truth for data set 'CC'. 71
102 Estimated positions (in Earth frame) and comparison to ground truth for data set 'DD'. 72
103 Comparisons of true and measured accelerations (in IMU body frame) for data set 'AA'. 74
104 Comparisons of true and measured accelerations (in IMU body frame) for data set 'BB'. 75
105 Comparisons of true and measured accelerations (in IMU body frame) for data set 'CC'. 76
106 Comparisons of true and measured accelerations (in IMU body frame) for data set 'DD'. 77
107 ArcSecond ground truth for data set 1 from the morning collection. 79
108 ArcSecond ground truth for data set 2 from the morning collection. 80
109 ArcSecond ground truth for data set 3 from the morning collection. 81
110 ArcSecond ground truth for data set 4 from the morning collection. 82
111 ArcSecond ground truth for data set 5 from the morning collection. 83
112 ArcSecond ground truth for data set 1 from the afternoon collection. 84
113 ArcSecond ground truth for data set 2 from the afternoon collection. 85
114 ArcSecond ground truth for data set 3 from the afternoon collection. 86
115 ArcSecond ground truth for data set 4 from the afternoon collection. 87
116 ArcSecond ground truth for data set 5 from the afternoon collection. 88
117 Measured accelerations after pre-processing. 89
118 Measured accelerations after pre-processing. 90
119 Stabilized velocities after bias and trend removal. 91
120 Stabilized velocities after bias and trend removal. 92
121 Stabilized attitude angles (roll, pitch, yaw) after bias and trend removal. 93
122 Stabilized attitude angles (roll, pitch, yaw) after bias and trend removal. 94
123 Stabilized velocities (in Earth frame) for data set 1, morning collection. 95
124 Stabilized velocities (in Earth frame) for data set 2, morning collection. 96
125 Stabilized velocities (in Earth frame) for data set 3, morning collection. 97

126 Stabilized velocities (in Earth frame) for data set 4, morning collection. 98
127 Stabilized velocities (in Earth frame) for data set 5, morning collection. 99
128 Stabilized velocities (in Earth frame) for data set 1, afternoon collection. 100
129 Stabilized velocities (in Earth frame) for data set 2, afternoon collection. 101
130 Stabilized velocities (in Earth frame) for data set 3, afternoon collection. 102
131 Stabilized velocities (in Earth frame) for data set 4, afternoon collection. 103
132 Stabilized velocities (in Earth frame) for data set 5, afternoon collection. 104
133 Estimated positions (in Earth frame) and comparison to ground truth for data set 1, morning collection. 105
134 Estimated positions (in Earth frame) and comparison to ground truth for data set 2, morning collection. 106
135 Estimated positions (in Earth frame) and comparison to ground truth for data set 3, morning collection. 107
136 Estimated positions (in Earth frame) and comparison to ground truth for data set 4, morning collection. 108
137 Estimated positions (in Earth frame) and comparison to ground truth for data set 5, morning collection. 109
138 Estimated positions (in Earth frame) and comparison to ground truth for data set 1, afternoon collection. 110
139 Estimated positions (in Earth frame) and comparison to ground truth for data set 2, afternoon collection. 111
140 Estimated positions (in Earth frame) and comparison to ground truth for data set 3, afternoon collection. 112
141 Estimated positions (in Earth frame) and comparison to ground truth for data set 4, afternoon collection. 113
142 Estimated positions (in Earth frame) and comparison to ground truth for data set 5, afternoon collection. 114

1 Introduction

The goal of this work is to develop robust algorithms to precisely determine the position of a handheld UXO sensor as it is swept above a suspected UXO object. It is critical that the sensor position be known precisely because research results have established that one of the most important factors in using UXO sensor data to characterize buried targets is precise knowledge of the sensor location and attitude while the data are being collected.

To assist in tracking the sensor position, an inertial measurement unit (IMU) is integrated with the sensor to record the sensor motion. Specifically, the sensor accelerations in the x-, y-, and z-coordinate directions are recorded, as well as the rotation rates about those axes. The algorithms under development are intended to determine the position and attitude of the handheld UXO sensor using the information recorded by the IMU.

The IMU accelerometers and angular rate sensor are most accurate during different types of motion and have different limitations. The accelerometer measurements are quite stable over long time periods, but are prone to errors due to saturation in response to very quick motions. The angular rate sensors, on the other hand, are quite accurate over short time periods, including during quick motions, but tend to suffer from drift over long time periods. Thus, the angular rate sensor can be used to compensate for the accelerometer saturation during quick motions and the accelerometer can be used to compensate for the angular rate sensor drift over long time periods. Adaptive error mitigation algorithms, which exploit these differences in the accelerometers and angular rate sensors are developed. These algorithms employ a feedback loop to use the more accurate sensor (accelerometer or angular rate) to mitigate errors in the less accurate sensor (accelerometer or angular rate).

System identification has been explored as a means to directly model the errors in the measured quantities (acceleration and angular rate). In this approach, the IMU is viewed as a black box with some unknown transfer function that relates the true quantity (acceleration or angular rate) to the quantity reported by the IMU. In other words, in this approach the IMU is viewed as a filter that alters the true accelerations and angular rates. System identification attempts to determine the transfer function of the unknown filter from known input and output data, and thus model the relationship between the observed data (output) and the ground truth (input(.

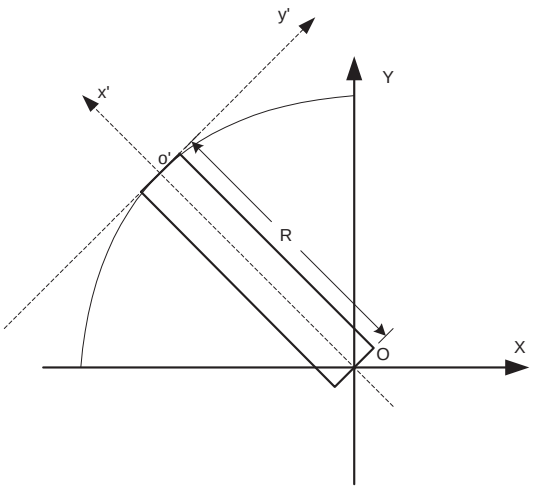


Figure 1: Body frame and earth frame geometry. (X,Y) represent the earth frame coordinates and (x',y') represent the body frame coordinates.

2 Iterative Integration Algorithm for Velocity Stabilization

The purpose of the velocity stabilization algorithm is to use the accelerometer and angular rate measurements together to estimate the sensor's velocity. Each of the sensors has its limitations. The accelerometer measurements are quite stable over long time periods, but are prone to errors due to saturation in response to very quick motions. The angular rate sensors, on the other hand, are quite accurate over short time periods, including during quick motions, but tend to suffer from drift over long time periods. The velocity stabilization algorithm uses both the accelerometer and the angular rate measurements to produce a stabilized velocity. The stabilization algorithms presented here were inspired by application notes obtained from XBow, the IMU manufacturer.

The velocity stabilization algorithm operates in the body frame, meaning the measured quantities and the stabilized velocities are measured relative to the orientation of the IMU. After determining the stabilized velocities in the body frame, they are transformed to the earth frame at which point they can be compared to the ground truth. The relationship between the body frame and the earth frame is illustrated in Fig. 1.

2.1 Angle (Attitude) Stabilization

An accurate estimate of the sensor's attitude is important for determining the stabilized velocity because the attitude of the sensor determines what part of the measured acceleration, if any, is due to gravity. Here, the approach is described for estimating the pitch angle of the sensor. Estimation of the sensor roll angle is achieved in a similar manner.

A block diagram illustrating the procedure for stabilizing the pitch angle is shown in Fig.2. In this depiction, Δt_k denotes the sample separation, $\theta_{e,pitch}$ is the error signal used to correct the raw angle $\hat{\theta}_{pitch}$, and K_θ is an adjustable gain parameter. Selection of the gain parameter is described in the Sec. 2.3

After an estimate of the IMU attitude has been found, it is used to subtract the gravity component from the measured accelerations. The corrected accelerations, with the estimate of the gravity compo-

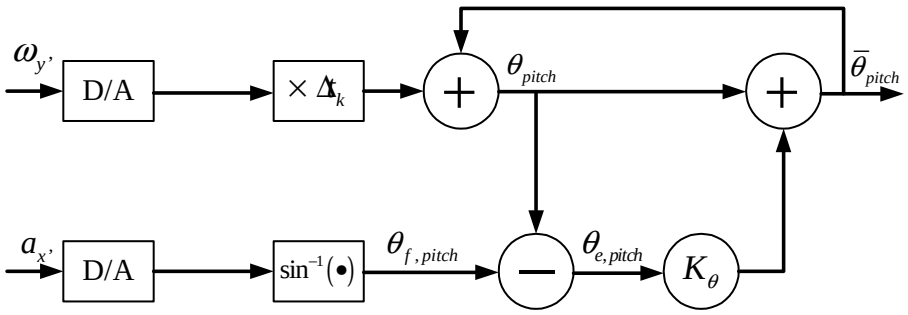


Figure 2: Iterative integration correction algorithm for pitch stabilization.

nent removed, can be expressed as

$$\hat{a}_{x'} = a_{x'} cos(\bar{\theta}_{pitch}) \tag{1}$$

$$\hat{a}_{y'} = a_{y'} cos(\bar{\theta}_{roll}) \tag{2}$$

where $\bar{\theta}_{pitch}$ and $\bar{\theta}_{roll}$ are the stabilized pitch angle and the stabilized roll angle.

2.2 Velocity Stabilization

The velocity stabilization algorithm operates in the body frame. After determining the stabilized velocity in the body frame the attitude of the sensor is utilized to determine its position in the earth frame.

In the absence of IMU measurement noise, the true velocity of the EMI sensor along the y' axis can be determined from either the angular rate sensor data or the accelerometer data. The estimated velocity inferred from the angular rates measurements is obtained by transforming the angular velocity to a tangential velocity. In actuality, the IMU measurements contain noise. The resulting relationships between the estimated velocities and the true velocity can be expressed as

$$\hat{V}_t = V_{y',0} + n_g \tag{3}$$

and

$$\hat{V}_{y'} = V_{y',0} + n_a, \tag{4}$$

where $V_{y',0}$ denotes the true velocity along y' direction in the body frame, $\hat{V}_t$ represents the velocity estimate from the angular rate sensor, $\hat{V}_{y'}$ is the velocity estimate from the accelerometer, n_g represents the noise of angular rate sensor, and n_a is the noise of accelerometer. Comparing these two quantities yields an error signal $V_{y',e}$ that is used to stabilize the raw velocity $\hat{V}_{y'}$. This is process is illustrated in the block diagram shown in Fig. 3. In this schematic, Δt_k denotes the sample separation and $V_{y',s}$ is the corrected velocity along y' direction. It should be noted that this approach uses an iterative integration method to calculate the velocity.

The velocity stabilization algorithm for the x' direction, shown in Fig. 4, is similar. In order to stabilize the velocity, two quantities are required, one derived from the accelerometer and one derived from the angular rate sensor.

Recall that the gravity angles are computed from the accelerations by

$$\theta_{y'} = sin^{-1}(a_{x'}). \tag{5}$$

Therefore, after stabilization, the angle can be used to determine the acceleration by

$$a_\theta = sin(\bar{\theta}_{pitch}), \tag{6}$$

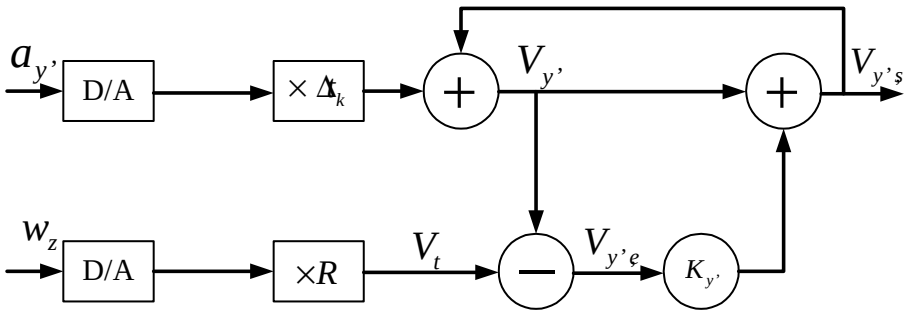


Figure 3: Iterative integration correction algorithm for velocity stabilization along the y' axis.

where $\bar{\theta}_{pitch}$ denotes the stabilized pitch angle which can be expressed as a function of both $w_{y'}$ and $a_{x'}$ in discrete form, namely

$$\bar{\theta}_{pitch,i+1} = (1-K_\theta)\left(\bar{\theta}_{pitch,i} + w_{y',i}\Delta t_i\right) + K_\theta sin^{-1}(a_{x'}), \tag{7}$$

where K_θ is the adjustable gain, $i = 1,2,\cdots,N$ and N represents the number of snapshots. This relationship describes the algorithm depicted in the block diagram shown in Fig. 2.

Equations (5) and (6) imply that the acceleration a_θ is more accurate than $\hat{a}_{x'}$ within short time periods since the angular rate sensor can provide a more accurate measurement than the accelerometer on a short time scale. Therefore, a_θ, which relies primarily on the angular rate sensor, is used to calculate the velocity on short time scales and $\hat{a}_{x'}$, which is stable over long time periods, is used to compute the velocity on long time scales. The iterative integration correction algorithm utilized both of these quantities to produce the stabilized velocity $V_{x',s}$ which is stable over both short and long time periods.

This relations used to stabilize the velocity along the x' orientation can be expressed as

$$\hat{V}_{x'} = \int \hat{a}_{x'} dt = V_{x',0} + n_a \tag{8}$$

$$\hat{V}_\theta = \int \hat{a}_\theta dt = V_{x',0} + n_g. \tag{9}$$

The iterative integration correction algorithm for velocity along x' orientation is shown in Fig. 4. Here, $\hat{V}_{x'}$ is the integration of the measured accelerations along the x' direction, $\hat{V}_\theta$ denotes the reference velocity derived from the stabilized angle $\bar{\theta}_{pitch}$, $V_{x',e}$ is the error signal, and $V_{x',s}$ is the stabilized velocity in the x' orientation.

The stabilized velocities are determined in the body frame (x',y') and must be utilized in conjunction with the estimated sensor attitude to transform them to the earth frame (x,y). The velocities in the earth frame can be written as

$$\hat{V}_x = \hat{V}_{y'} cos(\theta_{yaw}) - \hat{V}_{x'} sin(\theta_{yaw}) \tag{10}$$

and

$$\hat{V}_y = \hat{V}_{y'} sin(\theta_{yaw}) + \hat{V}_{x'} cos(\theta_{yaw}). \tag{11}$$

The stabilized positions in the earth frame are obtained by integrating of the stabilized velocities in the earth frame:

$$\hat{P}_x = \int \hat{V}_x dt \tag{12}$$

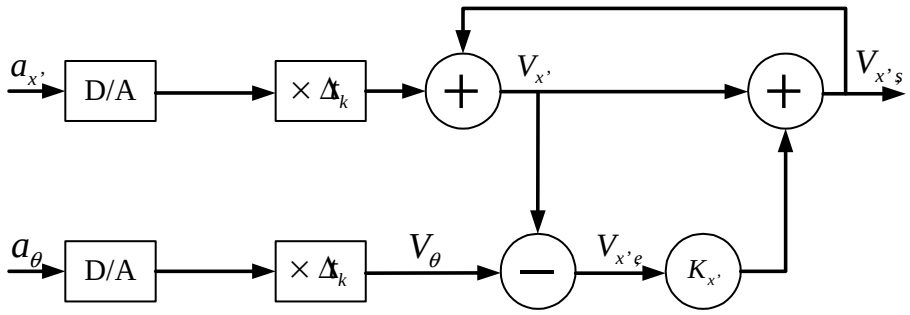


Figure 4: Iterative integration correction algorithm for velocity stabilization along the x' axis.

$$\hat{P}_y = \int \hat{V}_y dt \tag{13}$$

where $\hat{P}_x$ and $\hat{P}_y$ denote the corrected positions along the x orientation and the y orientation in the earth frame, respectively.

2.3 Gain Calculation

Each of the stabilization algorithms utilizes a gain parameter to scale the error signal. The performance of the stabilization algorithm depends on the value of the gain parameter. For the angle stabilization, the gain K_θ is analogous to the erection rate for a gyroscope. It is used to mitigate the error in the measured angle which results from the tilt of the angular rate sensor so the true angle can be estimated in the earth frame.

The optimal value for the gain depends on the noise characteristics of the sensors, so it can be determined only in the minimum mean square error (MMSE) sense. The method for determining the optimal value for the gain parameter is the same for each of the iterative integration correction algorithms. This is explained here using the velocity stabilization algorithm as an example. First, the case in which there is a single non-adapting gain parameter, termed "Fixed K," is considered. Then, the case in which the gain parameter is permitted to vary depending on the characteristics of the sensor motion, termed "Adaptive K," is considered.

For the fixed gain case, the optimal gain parameter is calculated as

$$K_{opt} = \min_K E\left[\left(\hat{V}_i - V_i\right)^2\right] = \min_K \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N}\left(\hat{V}_i - V_i\right)^2, \tag{14}$$

where $\hat{V}_i$, V_i, and N are the velocity estimate, the ground truth corresponding to $\hat{V}_i$, and the number of samples, respectively.

This approach, which uses a single gain parameter for all time results in an algorithm that is straight forward to implement. However, the characteristics of the sensor errors are a function of the sensor motion, so it is anticipated that there would be an advantage to allowing the gain parameter to vary with the sensor motion. In this case, the sensor motion can be partitioned into several distinct segments, where each segment is associated with a particular type of motion, and a gain parameter is calculated for each state. Examples of the partition of a path into segments are shown in Figs. 5 and 6 for the x-coordinate and the y-coordinate, respectively. These figures illustrate the results of partitioning the sensor path into three states; turning (blue circle), large acceleration (green asterisk), and small acceleration (red

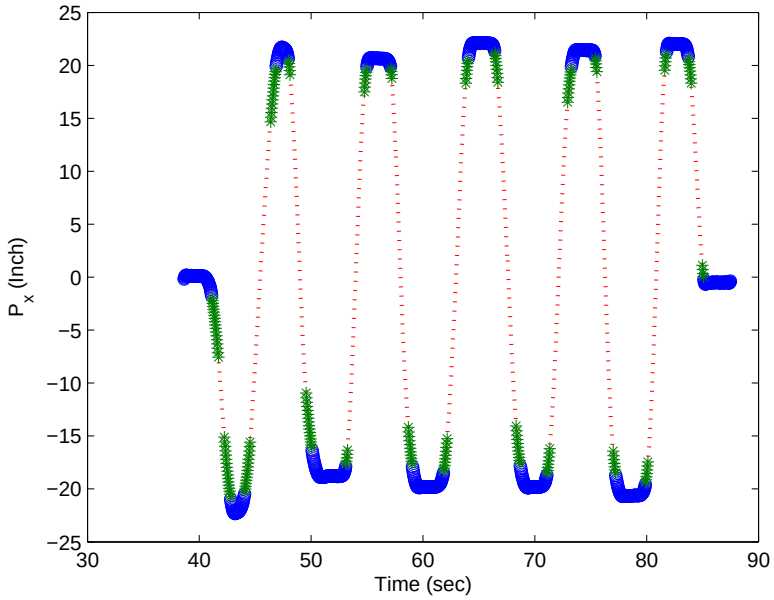


Figure 5: Example of the partition of the path along the *x* orientation.

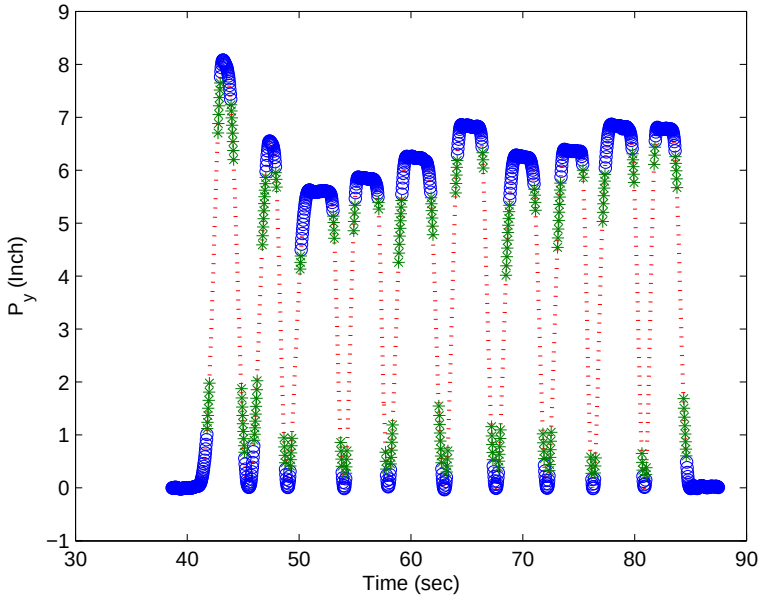


Figure 6: Example of the partition of the path along the *y* orientation.

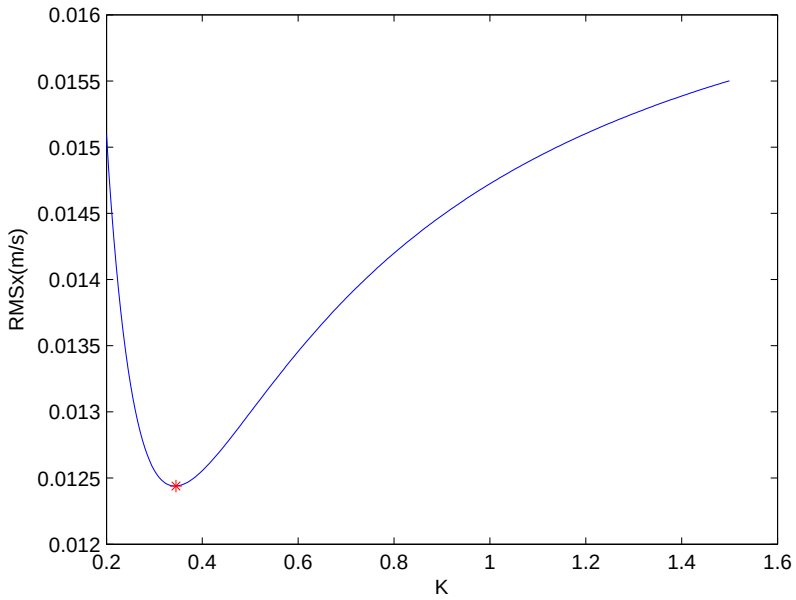


Figure 7: Example illustrating choosing K_x.

dot). This partition was done manually; automatic partitioning of the path into states is the subject of current work.

Since each state of the sensor occupies a block of samples, the calculated K for the t^{th} block of samples can be expressed as

$$\hat{K}_{opt}^{t} = \min_{K} \frac{1}{N_t} \sum_{i=1}^{N_t} \left(\hat{V}_i - V_i\right)^2 \tag{15}$$

where N_t is the number of samples in the t^{th} block, $\sum_{t=1}^{T} N_t = N$, and T denotes the number of blocks. Here we assume N_t is large enough to obtain a reasonable estimate of the gain parameter. We also assume the manual segmentation of the path into states is accurate. Figures 7 and 8 show the root of the mean square error (RMSE) versus the gain K. The red asterisk denotes the optimal K, which is used in the following analysis of the measured data.

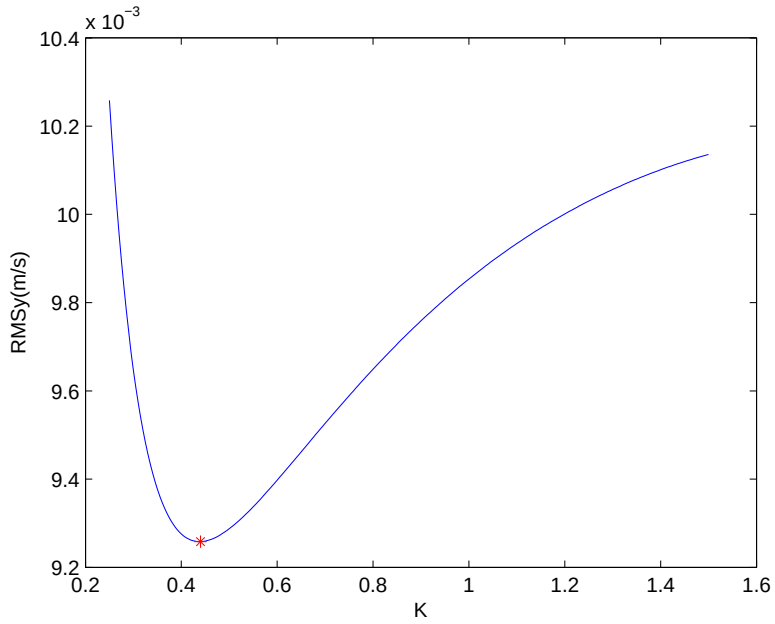


Figure 8: Example illustrating choosing K_y.

2.4 Experimental Results

2.4.1 Sweeping Data

The sweeping measurements consist of the EMI sensor being swept in an arc to simulate the motion an operator may make with it. The ground truth is taken from the video capture. The measured angular rates and accelerations are low-pass filtered for D/A conversion, and then any bias in the measurements is removed.

Figure 9 illustrates the velocities along the x axis in the earth frame after stabilization by the iterative integration correction algorithm. The velocity along the y axis in the earth frame after correction by the proposed scheme is shown in Fig. 10. The stabilized velocities closely follow the ground truth, indicating that the proposed algorithm with either a fixed gain or an adaptive gain effectively avoids the pitfalls associated with direct integration of the measured accelerations.

The estimated positions obtained by integrating the stabilized velocities are shown in Figs. 11 and 12. These results show that the adaptive gain provides a much better estimate of the sensor position than the fixed gain. Examining the error in the position estimates as a function of time, as shown in Figs. 13 and 14, illustrates that the fixed gain algorithm is prone to error accumulation, while the adaptive gain algorithm does not result in errors that increase with time. However, the remaining error after applying the algorithm with an adaptive gain does appear to be related to the nature of the sensor motion.

2.4.2 Linear Data

The linear measurements consist of the sensor moving back and forth in a straight line. Again, the ground truth is taken from the video capture. The measured angular rates and accelerations are low-pass filtered for D/A conversion, and then any bias in the measurements is removed.

Figures 15 and 16 illustrate the stabilized velocities in the x and y directions in the earth frame. The estimated positions obtained by integrating the stabilized velocities are shown in Figs. 17 and 18 and the errors follow in Figs. 19 and 20. Although the results obtained with the iterative integration correction algorithms are better than the baseline, and, further, the algorithm with the adaptive gain performs better than the algorithm with the fixed gain, the position estimates for the linear data are not as good as for the sweeping data. The source of the errors, and approaches to reduce them are the subject of ongoing research.

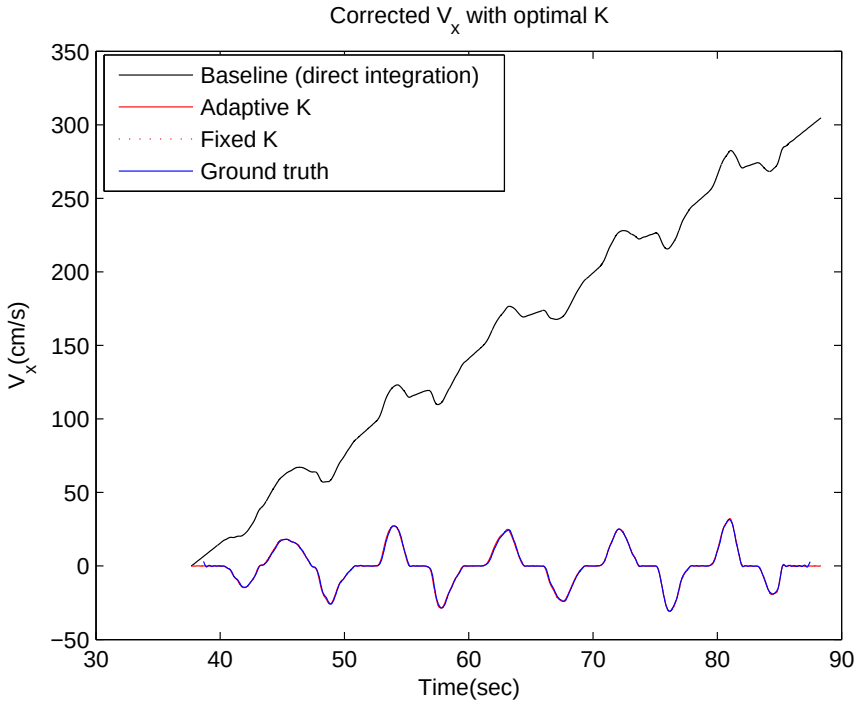


Figure 9: Stabilized velocities along the *x* axis with adaptive K and fixed K algorithms.

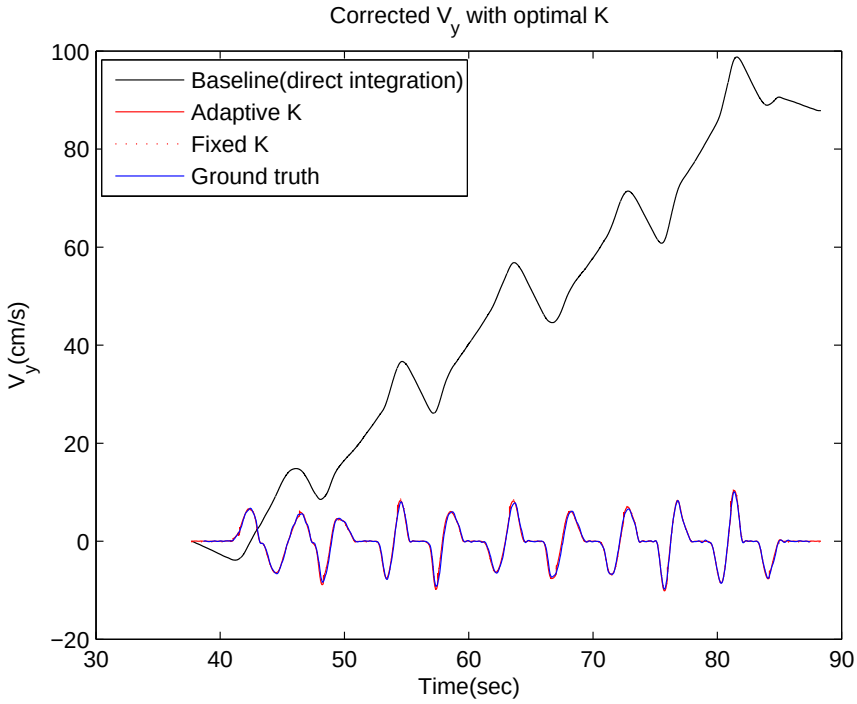


Figure 10: Stabilized velocities along the *y* axis with adaptive K and fixed K algorithms.

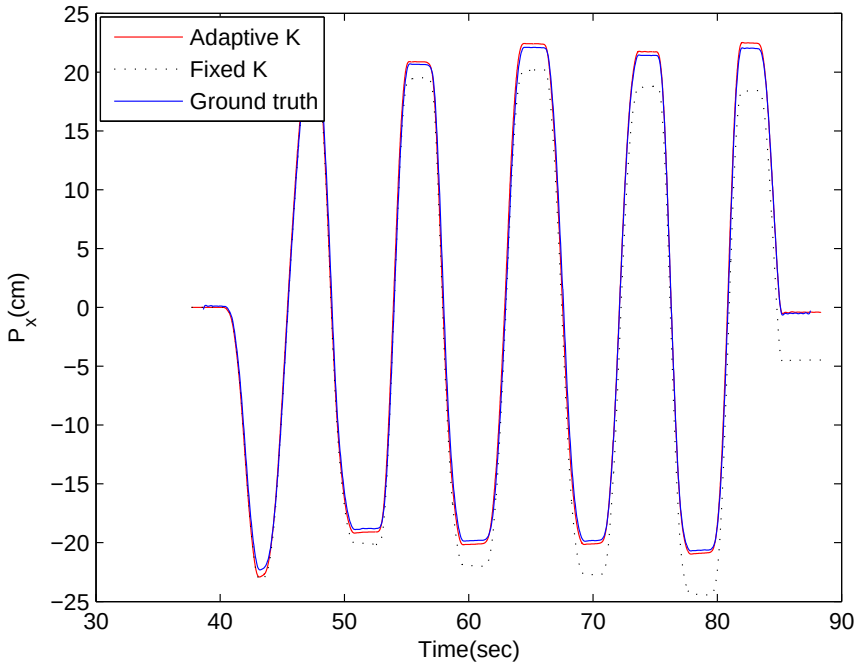


Figure 11: Stabilized positions along the *x* axis with adaptive K and fixed K algorithms.

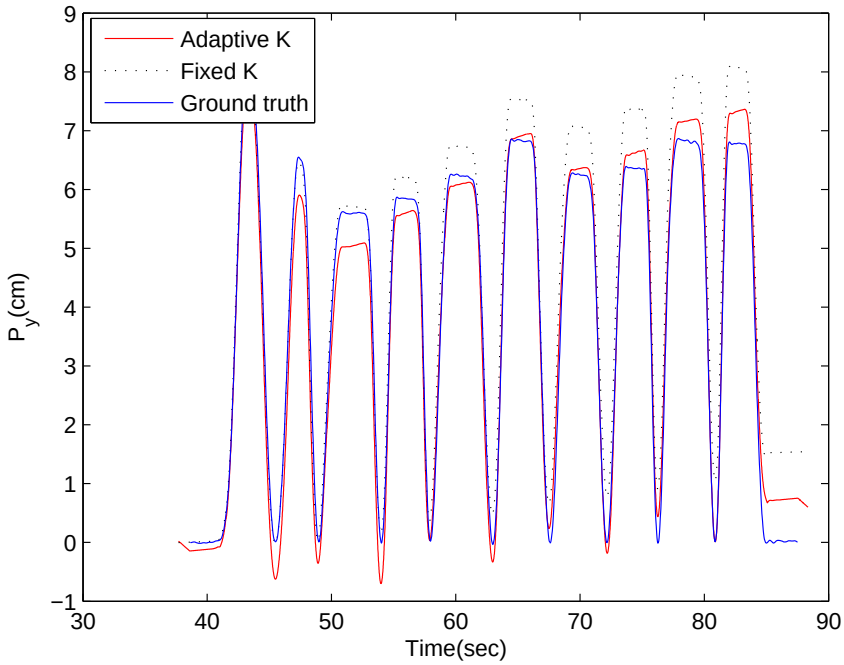


Figure 12: Stabilized positions along the *y* axis with adaptive K and fixed K algorithms.

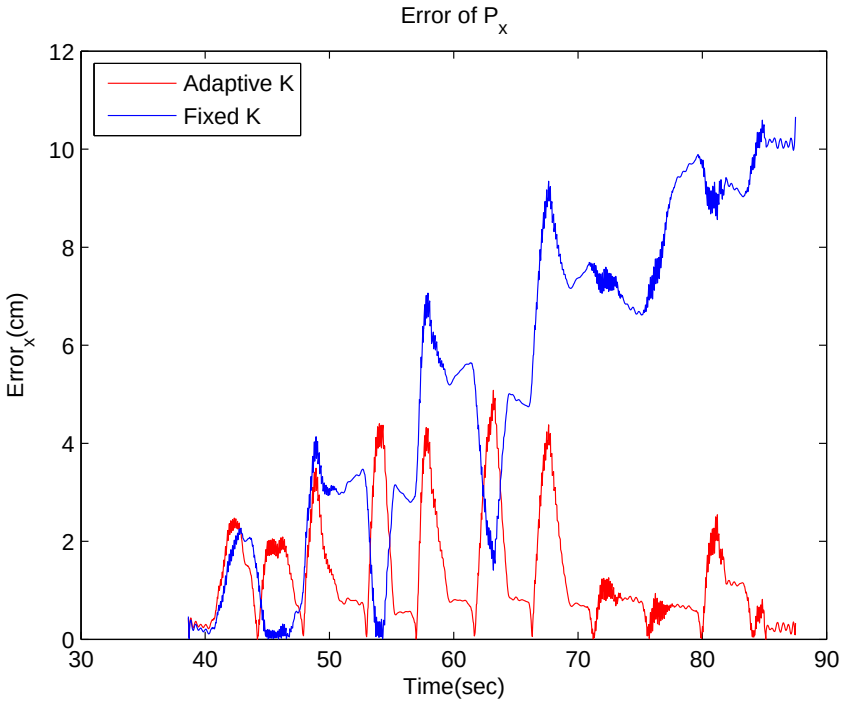


Figure 13: Errors in stabilized positions along the *x* axis with adaptive K and fixed K algorithms.

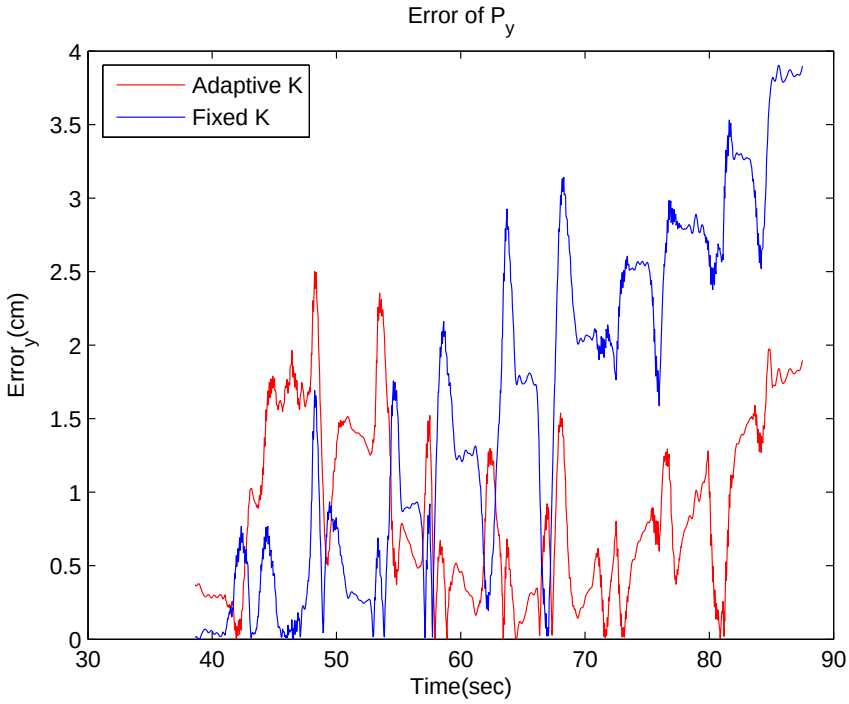


Figure 14: Errors in stabilized positions along the *y* axis with adaptive K and fixed K algorithms.

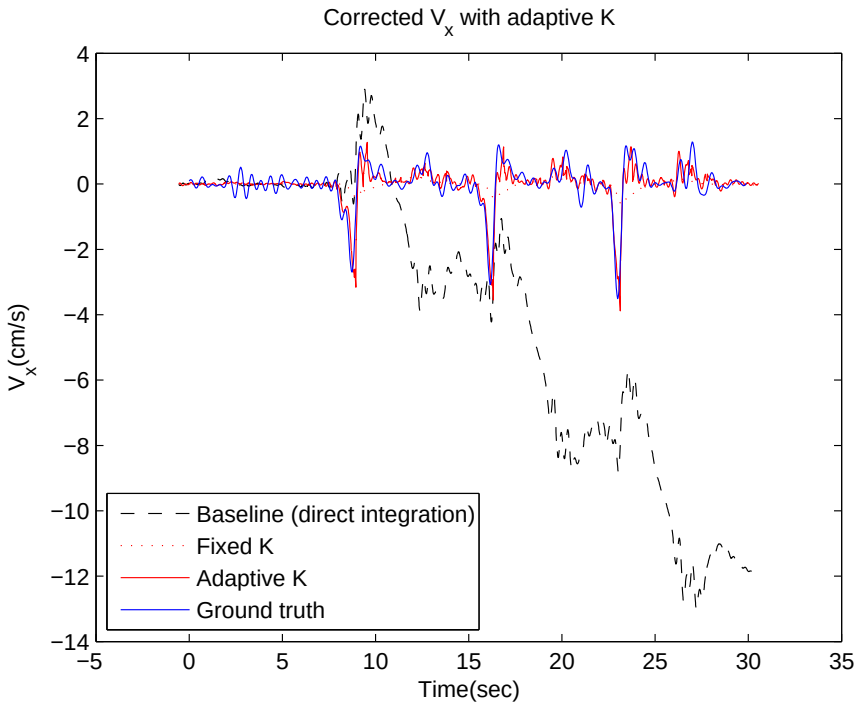


Figure 15: Calculated velocity along the *x* axis based on the linear data.

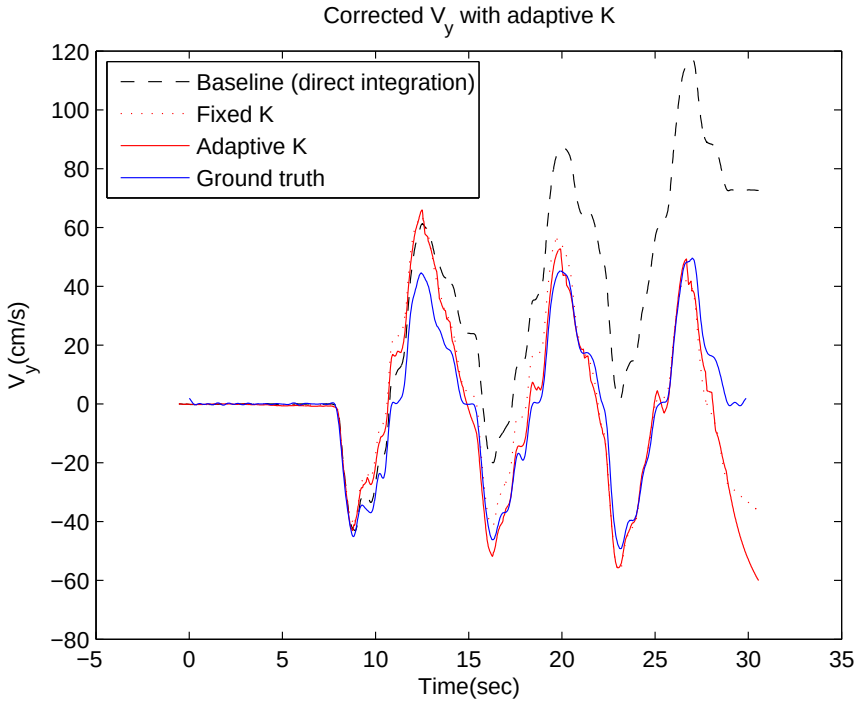


Figure 16: Calculated velocity along the *y* axis based on the linear data.

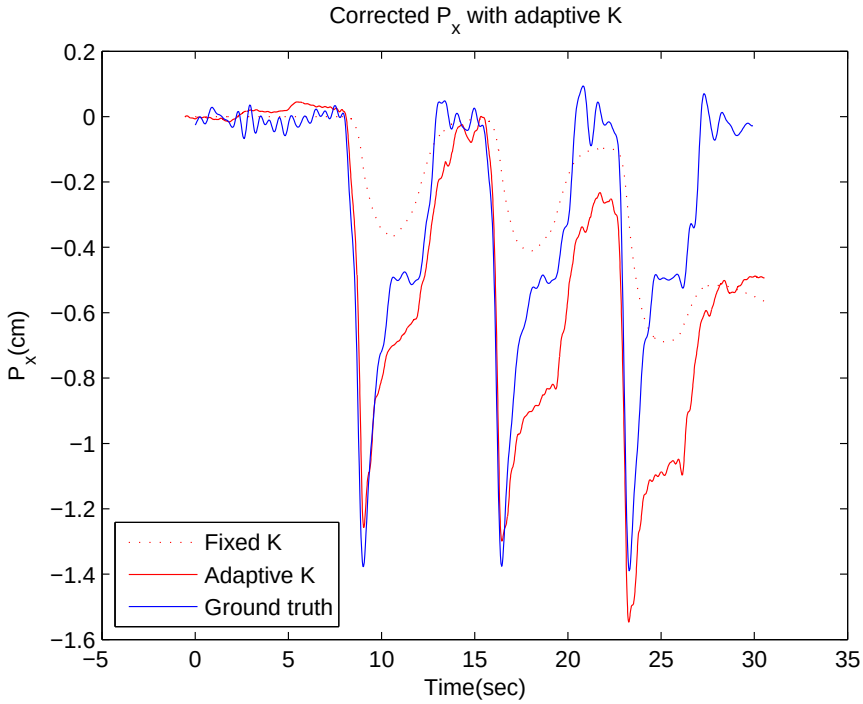


Figure 17: Calculated position along the *x* axis based on the linear data.

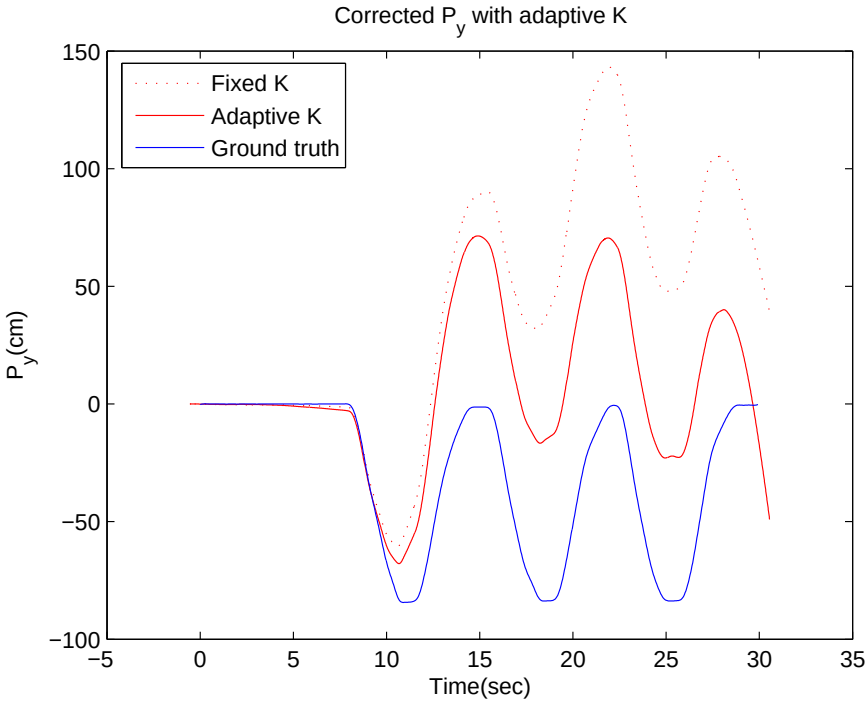


Figure 18: Calculated position along the *y* axis based on the linear data.

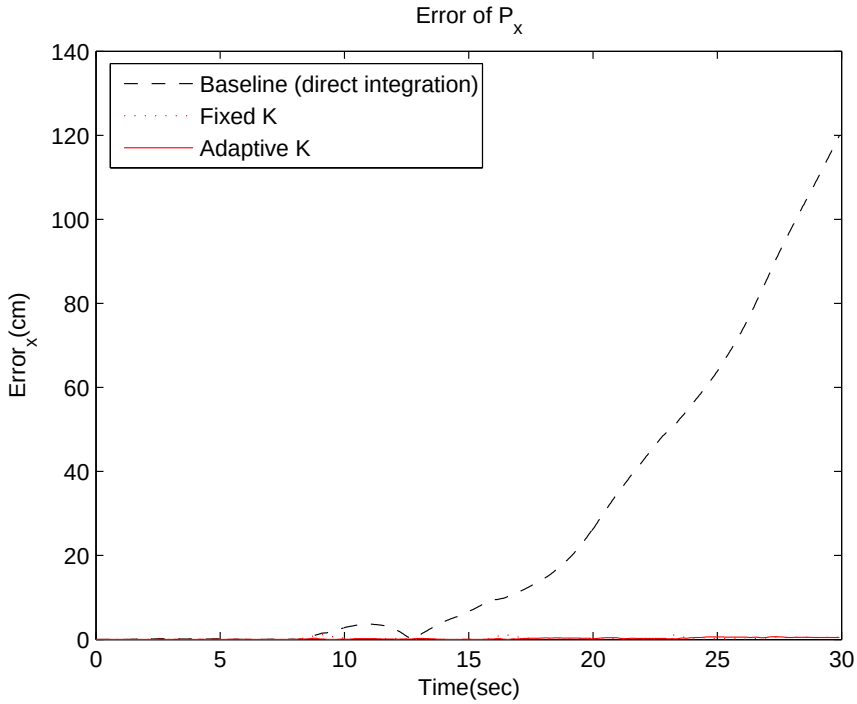


Figure 19: Estimation error of the position along the *x* axis based on the linear data.

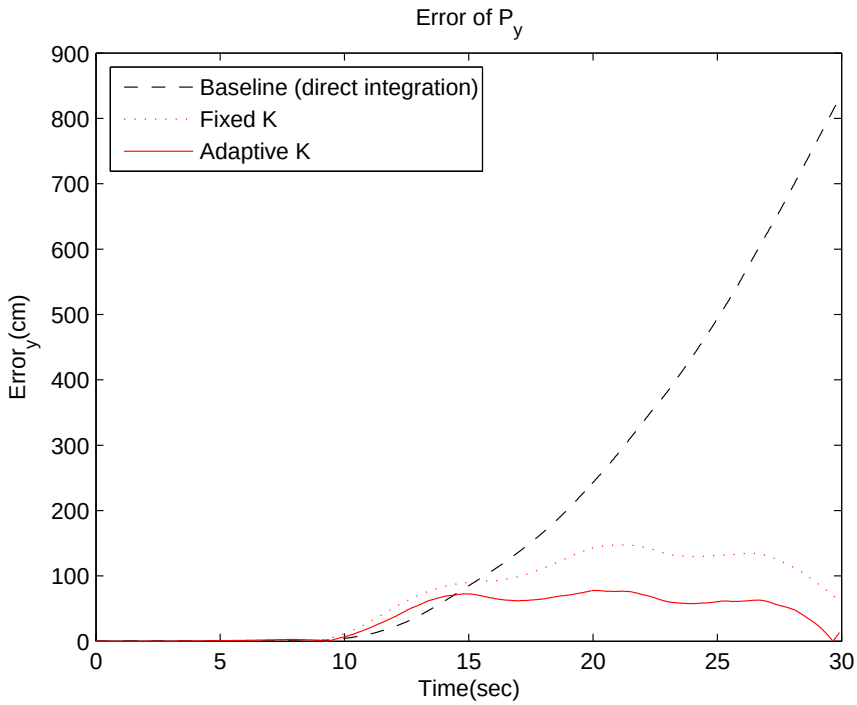


Figure 20: Estimation error of the position along the *y* axis based on the linear data.

2.5 Summary

The iterative integration correction algorithm is an effective approach to improving the sensor position estimates, particularly when the gain parameter is permitted to adapt to the nature of the sensor motion. The position errors for the sweeping data are reduced to less than 5 cm. This approach has the benefit of not making any *a priori* assumptions about the sensor motion; it is applicable to any sensor motion path. However, it does assume that the sensor arm is pivoting about the point where the IMU is attached to it, which may be an inaccurate assumption.

3 Iterative Integration for Position Calculation

In this section, the iterative integration correction algorithm for calculating the attitude (angles), velocities, and positions of the hand-held UXO sensor system is presented. The proposed method proceeds in three steps:

Step 1: Stabilize the pitch and roll angles and use them to remove the effect of gravity to find the true accelerations in the body frame of the 2-D plane.

Step 2: Calculate the velocities and positions by using the iterative integration correction algorithm.

Step 3: Based on the white noise assumption of the measurement noise, calculate the fixed and adaptive gains for the iterative integration correction algorithm.

3.1 Sensor Measurement Error Models

The accelerations, $\hat{a}$, and angular rates, $\hat{\omega}$, measured by the IMU in the body frame can be expressed as

$$\hat{a}_x = a_x + b_{ax} + n_{ax} \quad (16)$$
$$\hat{a}_y = a_y + b_{ay} + n_{ay} \quad (17)$$
$$\hat{a}_z = a_z + b_{az} + n_{az} \quad (18)$$

and

$$\hat{\omega}_x = \omega_x + b_{\omega x} + n_{\omega x} \quad (19)$$
$$\hat{\omega}_y = \omega_y + b_{\omega y} + n_{\omega y} \quad (20)$$
$$\hat{\omega}_z = \omega_z + b_{\omega z} + n_{\omega z}, \quad (21)$$

where the subscripts x, y, and z denote the coordinate axis, a represents the true acceleration, ω represents the true angular rates, b represents the bias in the measurement, and n represents the measurement noise, which is modeled as a white Guassian process.

Integration of measured accelerations and angular rates over time yields velocities and angles, and double integration of accelerations over time produces the positions of the hand-held UXO sensor system. Thus, for coordinate direction i the estimated velocity, $\hat{V}_i$, position, $\hat{P}_i$, and angle, $\hat{\theta}_i$, are

$$\hat{V}_i = V_i + \int b_{ai} dt + \int n_{ai} dt \quad (22)$$
$$\hat{P}_i = P_i + \int\int b_{ai} dsdt + \int\int n_{ai} dsdt \quad (23)$$
$$\hat{\theta}_i = \theta_i + \int b_{\omega i} dt + \int n_{\omega i} dt, \quad (24)$$

where V_i, P_i, and θ_i are the true velocity, positions, and angle along the coordinate direction i. (33)-(25) indicate that the bias and sensor noise are the primary causes of divergence of the position estimates. The bias affects the velocity and angle linearly with time, and the positions quadratically. Moreover, the integration of the noise causes the variance of the estimate to increase proportional to the integration time,

$$\sigma_n^2 = t. \quad (25)$$

This suggests the the integration of the noise creates a boundlessly growing error in the velocities, positions, and angles determined by integrating the IMU sensor measurements.

3.2 Stabilized Angle Calculation

To obtain an accurate estimate of velocity, we must first correctly estimate the tilt angles: pitch and roll, and then combine these corrected tilt angles and the measurement accelerations to remove the effect of gravity and therefore determine the true accelerations. The corrected accelerations allow us to accurately calculate the velocities with an efficient method. Since the method for pitch angle estimation is the same as for roll angle estimation, we only focus on the estimate of the pitch angle in this subsection.

Since a tilt sensor is capable of providing accurate angle measurements over long time periods while the angular rate sensor can accurately sense the the angular velocity over short time periods, we utilize the angular rate sensor to record angle changes on short time scales and use the accelerometer as a tilt sensor to sense the tilt angles, and force the angular rate sensor derived tilt angle to slowly match the accelerometer angles on long time scales. The algorithm for the pitch angle θ_x estimation is described in Fig. 1 in Crossbow's Application Note, "Measurement of a Vehicle's Dynamic Motion," where Δt_i denotes the sample separation, $e_{\theta_x,i}$ is the error signal and used to correct the raw angle $\hat{\theta}_{x,i} = \bar{\theta}_{x,i} + \hat{w}_{y,i}\Delta t_i$, $\hat{\theta}_{a_x,i}$ is the reference pitch angle and K_θ is an adjustable gain parameter. Note that the angular rate sensor sensing the angular velocity around y orientation gives the measurement of the pitch angle changes, which is along x orientation. It is straightforward to use a similar method to obtain the corrected roll angle $\bar{\theta}_y$.

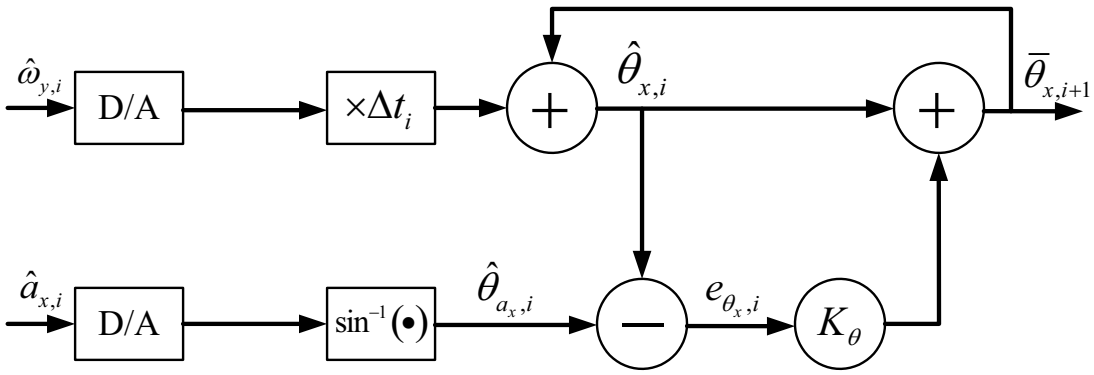


Figure 21: Iterative integration correction algorithm for pitch angle calculation.

Consider the geometry shown in Fig. 22 where the body frame and the earth frame are confined to the same 2-D plane, where R denotes the length of the arm of the hand-held UXO sensor system, xoy is the coordinate system of the body frame and XOY is the coordinate system of the earth frame. After calculating the stabilized tilt angles, *i.e.* $\bar{\theta}_x$ and $\bar{\theta}_y$, we use them to correct the accelerations and thereby obtain the true accelerations that are level relative to the earth. Thus, the corrected accelerations can be written as

$$\bar{a}_x = \hat{a}_x cos(\bar{\theta}_x) \tag{26}$$

$$\bar{a}_y = \hat{a}_y cos(\bar{\theta}_y) \tag{27}$$

where $\bar{\theta}_x$ and $\bar{\theta}_y$ are the stabilized pitch and roll angles, respectively.

3.3 Velocity Stabilization and Position Calculation

In the following discussion, we only consider the body frame shown in Fig. 22. After computing the stabilized velocities in the body frame, we transform them to the earth frame by using the yaw angle

and finally obtain the true position by directly integrating the corrected velocities in the earth frame over time.

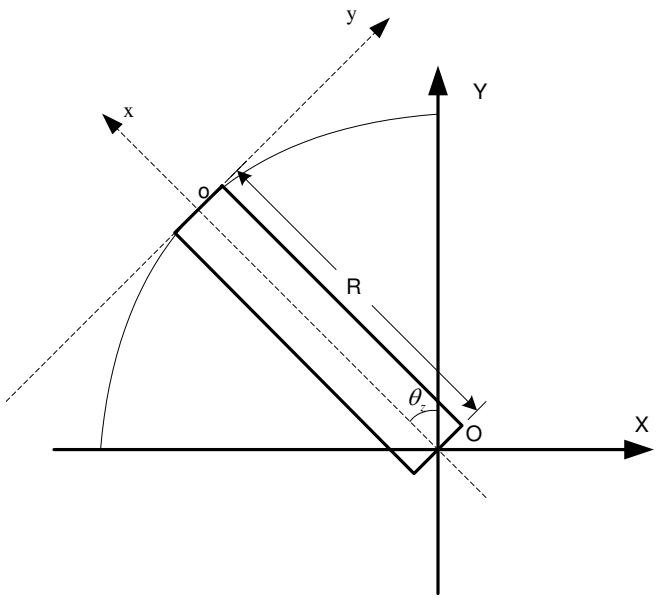


Figure 22: Geometry of earth frame and body frame. *xoy* is the coordinate system of the body frame and *XOY* is the coordinate system of the earth frame.

Noting that both the tangential velocity of the EMI sensor and the velocity of the IMU along the y orientation equal the true velocity along y axes in the unbiased and noise-free case and considering (32) and (33), we have the following relations

$$\begin{aligned} \hat{V}_t &= \hat{w} \cdot R = w \cdot R + b_{wz} \cdot R + n_{wz} \cdot R \\ &= V_t + b_{wz} \cdot R + n_{wz} \cdot R \\ &= V_y + b_{wz} \cdot R + n_{wz} \cdot R \end{aligned} \tag{28}$$

$$\hat{V}_y = V_y + \int b_{ay} dt + \int n_{ay} dt \tag{29}$$

where $V_y = \int a_y dt$ denotes the true velocity along y direction in the body frame, $\hat{V}_t = \hat{w}_z \cdot R$ represents the estimate of the tangential velocity $V_t = w_z \cdot R$ and R is the length of the arm of the handheld UXO sensor system, $\hat{V}_y$ is the calculated velocity based on the acceleration measurement of the IMU along y orientation. Comparing these two calculated velocities leads to an error signal $e_{V_y} = \hat{V}_t - \hat{V}_y$. It is notable that, on long time scales, the accelerometer is quite accurate to sense the acceleration; on short time scales, however, the angular rate sensor is more sensitive than the accelerometer to the changes in angular rate and therefore can provide accurate measurement of the tangential velocity in the y direction. Therefore, we should mainly utilize the acceleration to calculate the velocity over long times and exploit the short term stability of the angular rate sensor to keep the accelerometer drift in check. Thus this error signal can be used to stabilize the raw velocity $\hat{V}_y$. This is illustrated in Fig. 23, where Δt_i denotes the sample separation and $\bar{V}_{y,i+1}$ is the corrected velocity along y orientation. Accordingly, the calculated incremental velocity at the ith iteration can be expressed as:

$$\bar{V}_{y,i+1} = (1 - K_y)\left(\bar{V}_{y,i} + \bar{a}_{y,i}\Delta t_i\right) + K_y \hat{V}_{t,i} \tag{30}$$

where $i = 1, 2, \cdots, N$ and N represents the number of snapshots.

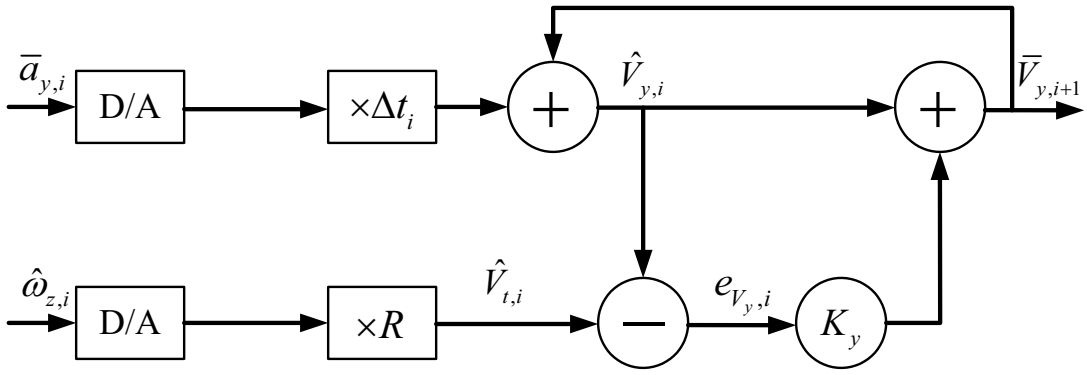


Figure 23: Iterative integration correction algorithm for velocity stabilization along y axes.

Notice that the calculation of the tangential velocity V_t does not involve the integration method, which indicates that there is no accumulated error in the process of calculating the tangential velocity. This therefore can be helpful to improve the estimation performance of the velocity along y orientation.

To obtain the corrected velocity along x orientation, we must find another velocity resulting from the angular rate sensor to stabilize the velocity along x orientation that is calculated by the measurement of the accelerometer. Note that the gravity angles are computed from the accelerations:

$$\theta_x = \sin^{-1}(a_x). \tag{31}$$

As is well known, the stabilized pitch angle $\bar{\theta}_x$ is responsive to the actual rotation of the hand-held UXO sensor system, and relatively insensitive to its linear acceleration. Therefore, after stabilized, the pitch angle can be used to determine the acceleration:

$$a_\theta = \sin(\bar{\theta}_x) \tag{32}$$

which is quite sensitive to angular rate and relatively insensitive the linear acceleration, where $\bar{\theta}_x$ denotes the stabilized pitch angle that can be written as a function of $\hat{w}_y$ as well as $\hat{a}_x$ in discrete form, namely

$$\bar{\theta}_{x,i+1} = (1 - K_\theta)\left(\bar{\theta}_{x,i} + \hat{w}_{y,i}\Delta t_i\right) + K_\theta \sin^{-1}(\hat{a}_x) \tag{33}$$

where K_θ is the adjustable gain. The corresponding block diagram is shown in Fig. 21.

In the sequel, the acceleration a_θ is more accurate than $\hat{a}_x$ within short time period since the gyroscope can provide a more accurate measurement than the accelerometer on a short time scale. Thereby, we use a_θ that mainly relies on the angular rate sensor to calculate the velocity on short time scale and use $\hat{a}_x$ that is stable over long time period to compute the velocity on long time scale, and finally producing the velocity calculation $\bar{V}_x$ that is stable over both short and long times. This relation used to stabilize the velocity along x orientation can be expressed as

$$\Delta V_{x,i} = \bar{a}_{x,i}\Delta t_i = a_{x,i}\Delta t_i + b_{ax,i}\Delta t_i + n_{ax,i}\Delta t_i \tag{34}$$

$$\Delta V_{\theta,i} = a_{\theta,i}\Delta t_i = a_{x,i}\Delta t_i + b_{\theta,i}\Delta t_i + n_{\theta,i}\Delta t_i \tag{35}$$

where $b_{\theta,i}$ and $n_{\theta,i}$ are the bias and noise of the calculated pitch angle. The iterative integration correction algorithm for velocity along the x orientation is shown in Fig. 24, in which ΔV_θ denotes the

reference incremental velocity resulting from the stabilized angle $\bar{\theta}_x$, $e_{V_x} = \hat{V}_x - \Delta V_\theta$ is the error signal and $\bar{V}_{x,i+1}$ is the stabilized velocity along x orientation. Therefore, the incremental velocity along x orientation is computed as:

$$\bar{V}_{x,i+1} = (1 - K_x)\left(\bar{V}_{x,i} + \bar{a}_{x,i}\Delta t_i\right) + K_x\left(a_{\theta,i}\Delta t_i\right). \tag{36}$$

Because the stabilized pitch angle is only capable of determining the acceleration rather than velocity, only the incremental velocities derived from the stabilized pitch angle and the acceleration are compared to produce the error signal e_{V_x}, the correction ability of this scheme is not comparable to that depicted in Fig. 23. Fortunately, the operator uses the hand-held UXO detector by the way of swinging it side to side to detect the potential UXO targets in a relatively large region. Therefore, the linear motion of the hand-held UXO sensor system is seldom encountered. Furthermore, in the final calculation of the position of the hand-held UXO sensor system, we also consider the measurement error of the yaw angle and the calculation error due to the integration of the stabilized velocities over time to compute the fixed and adaptive gains for the correction loop of the iterative integration algorithm. Thus the calculation error of positions of the hand-held UXO sensor system can be reduced as low as possible.

To obtain the true velocities in the earth frame, we must convert the stabilized velocities from the body frame (xoy) to the earth frame (XOY) by exploiting the yaw angle $\hat{\theta}_z$. The velocities in the earth frame can be written as

$$\hat{V}_X(K_x, K_y) = \bar{V}_y(K_y)\cos(\hat{\theta}_z) - \bar{V}_x(K_x)\sin(\hat{\theta}_z) \tag{37}$$

$$\hat{V}_Y(K_x, K_y) = \bar{V}_y(K_y)\sin(\hat{\theta}_z) + \bar{V}_x(K_x)\cos(\hat{\theta}_z) \tag{38}$$

where $\hat{V}_X(K_x, K_y)$ and $\hat{V}_Y(K_x, K_y)$ are a function of the adjustable gain K_x and K_y. The relationship between the body frame and the earth frame is illustrated in Fig. 22. The stabilized positions in the earth frame can be achieved by the direct integration of the corrected velocities over time:

$$\hat{P}_X(K_x, K_y) = \int \hat{V}_X(K_x, K_y)\, dt \tag{39}$$

$$\hat{P}_Y(K_x, K_y) = \int \hat{V}_Y(K_x, K_y)\, dt \tag{40}$$

where $\hat{P}_X$ and $\hat{P}_Y$ denote the corrected positions along X and Y orientations in the earth frame, respectively.

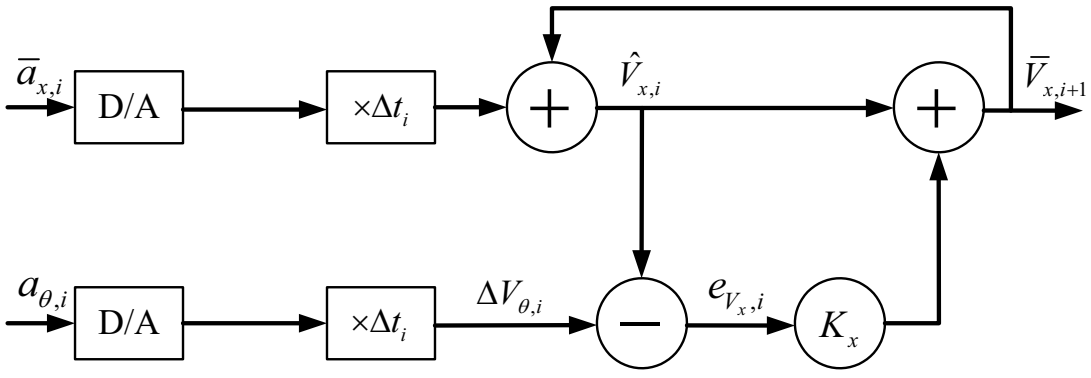


Figure 24: Iterative integration correction algorithm for velocity stabilization along x axes.

3.4 Gain Calculation

3.4.1 Fixed Gain Calculation

In the case of angle stabilization, the gain K_θ is the erection rate. Just as its name suggests, the gain is used to erect out the angle errors by moving the estimated angle more toward the angle measurements derived from the accelerometer measurements. In this section we only focus on the calculation of the gain for velocity stabilization. Because our final goal is to obtain an accurate position estimate of the UXO sensor system, for the fixed gain case, we calculate the optimal gains $K_{x,opt}^{(X)}$ and $K_{y,opt}^{(X)}$ along X orientation as

$$\begin{aligned}\left\{K_{x,opt}^{(X)}, K_{y,opt}^{(X)}\right\} &= \arg\min_{K_x,K_y} E\left[\hat{P}_X\left(K_x,K_y\right)-P_X\right]^2 \\ &= \arg\min_{K_x,K_y}\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\left[\hat{P}_{X,i}\left(K_x,K_y\right)-P_{X,i}\right]^2 \end{aligned} \tag{41}$$

and the optimal gains $K_{x,opt}^{(Y)}$ and $K_{y,opt}^{(Y)}$ along Y orientation as

$$\begin{aligned}\left\{K_{x,opt}^{(Y)}, K_{y,opt}^{(Y)}\right\} &= \arg\min_{K_x,K_y} E\left[\hat{P}_Y\left(K_x,K_y\right)-P_Y\right]^2 \\ &= \arg\min_{K_x,K_y}\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\left[\hat{P}_{Y,i}\left(K_x,K_y\right)-P_{Y,i}\right]^2. \end{aligned} \tag{42}$$

Note that the optimal gains are calculated only in the training phase, and therefore will not increase the computational burden of positioning the hand-held UXO sensor system in practical applications.

3.4.2 Adaptive Gain Calculation

Obviously, there is no one optimal gain for all types of sensor motion. Therefore, it is necessary to compute the optimal gain for different types of motion to achieve more robust estimates of positions. While [1] proposed the adaptive T-setting method to switch between two different gains for the computation of angles, it did not present an efficient method to calculate the optimal gain. Moreover, the calculations of position of vehicles are more complicated than the computation of attitude since the former involves the stabilized angle calculation, acceleration correction as well as double integrations of the corrected acceleration. So, even a small measurement error in the IMU measurements will result in a very large calculation error in the position estimation.

To calculate the optimal gains in the training phase, we assume that the hand-held UXO sensor system has two different states, namely abrupt or fast dynamic motion and smooth or slow dynamic motion: the accelerations larger than the threshold γ_a or the angular rates larger than γ_ω are deemed to be in the *fast* dynamic motion state, otherwise, the accelerations and angular rates are considered to be in the *slow* dynamic motion state. In the sequel, from (15) and (21) it follows that the incremental velocities along x and y orientations can be expressed as

$$\bar{V}_{x,i+1} = \begin{cases} \left(1-K_x^s\right)\left(\bar{V}_{x,i}+\bar{a}_{x,i}\Delta t_i\right)+K_x^s\left(a_{\theta,i}\Delta t_i\right), & \bar{a}_{x,i}<\gamma_a, \hat{\omega}_{y,i}<\gamma_\omega; \\ \left(1-K_x^f\right)\left(\bar{V}_{x,i}+\bar{a}_{x,i}\Delta t_i\right)+K_x^f\left(a_{\theta,i}\Delta t_i\right), & \bar{a}_{x,i}\geq\gamma_a, \hat{\omega}_{y,i}\geq\gamma_\omega. \end{cases} \tag{43}$$

and

$$\bar{V}_{y,i+1} = \begin{cases} \left(1-K_y^s\right)\left(\bar{V}_{y,i}+\bar{a}_{y,i}\Delta t_i\right)+K_y^s\hat{V}_{t,i}, & \bar{a}_{y,i}<\gamma_a, \hat{\omega}_{z,i}<\gamma_\omega; \\ \left(1-K_y^f\right)\left(\bar{V}_{y,i}+\bar{a}_{y,i}\Delta t_i\right)+K_y^f\hat{V}_{t,i}, & \bar{a}_{y,i}\geq\gamma_a, \hat{\omega}_{z,i}\geq\gamma_\omega. \end{cases} \tag{44}$$

Thereby, after conversion from the body frame to the earth frame, the calculated velocity along the X orientation in the earth frame is given by

$$\hat{V}_X\left(K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}\right)=\bar{V}_y\left(K_y^{(X),s},K_y^{(X),f}\right)\cos(\hat{\theta}_z)-\bar{V}_x\left(K_x^{(X),s},K_x^{(X),f}\right)\sin(\hat{\theta}_z) \tag{45}$$

and the computed velocity along the Y orientation in the earth frame is written as

$$\hat{V}_Y\left(K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}\right)=\bar{V}_y\left(K_y^{(Y),s},K_y^{(Y),f}\right)\sin(\hat{\theta}_z)+\bar{V}_x\left(K_x^{(Y),s},K_x^{(Y),f}\right)\cos(\hat{\theta}_z). \tag{46}$$

Thus, the optimal gains along X and Y orientations can be found by

$$\begin{aligned}
&\left\{K_{x,opt}^{(X),s},K_{x,opt}^{(X),f},K_{y,opt}^{(X),s},K_{y,opt}^{(X),f}\right\}\\
=\;&\arg\min_{K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}} E\left[\hat{P}_X\left(K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}\right)-P_X\right]^2\\
=\;&\arg\min_{K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}}\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\left[\hat{P}_{X,i}\left(K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}\right)-P_{X,i}\right]^2,
\end{aligned} \tag{47}$$

and

$$\begin{aligned}
&\left\{K_{x,opt}^{(Y),s},K_{x,opt}^{(Y),f},K_{y,opt}^{(Y),s},K_{y,opt}^{(Y),f}\right\}\\
=\;&\arg\min_{K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}} E\left[\hat{P}_Y\left(K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}\right)-P_Y\right]^2\\
=\;&\arg\min_{K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}}\lim_{N\to\infty}\frac{1}{N}\sum_{i=1}^{N}\left[\hat{P}_{Y,i}\left(K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}\right)-P_{Y,i}\right]^2
\end{aligned} \tag{48}$$

where

$$\hat{P}_X\left(K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}\right)=\int\hat{V}_X\left(K_x^{(X),s},K_x^{(X),f},K_y^{(X),s},K_y^{(X),f}\right)dt \tag{49}$$

and

$$\hat{P}_Y\left(K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}\right)=\int\hat{V}_Y\left(K_x^{(Y),s},K_x^{(Y),f},K_y^{(Y),s},K_y^{(Y),f}\right)dt. \tag{50}$$

After finding the optimal adaptive gains, we insert them into (28) and (29) to attain the stabilized velocity calculations and then find the corrected position calculations by employing (30), (31), (34) and (35).

3.5 Experimental Results

3.5.1 Interpolation of Ground Truth Data

To apply the iterative integration correction technique to practical applications, we must compute the gains in the training phase because the ground truth is unavailable in practical implementations. However, the ground truth is recorded by video tracks and its sample rate is generally not equal to that of the measurement data, causing the number of samples in the ground truth to be different from the number in the measurement data. In addition, the sample times of the measurement data are not aligned with those of the video tracks either. These non-ideal conditions will lead to the misalignment errors the gains are calculated. We here apply the interpolation method to ground truth and align the sample time of the ground truth with that of the measurement data. Hence we can obtain the same number of the samples of the ground truth as the measurement data and reduce the misalignment errors.

3.5.2 Position Calculation

The measurements consist of the EMI being swept in an arc to simulate the motion an operator may make with it. The ground truth is taken from the video capture. The measured angular rates and accelerations are low-pass filtered for D/A conversion, and then any bias in the measurements is subtracted.

Fixed Gain Case In this case, the optimal gains are calculated by (26) and (27). Fig. 25 shows the root mean square error (RMSE) of the calculated positions of the hand-held UXO sensor along X orientation varying with the gains K_x and K_y. The top and bottom panels of Fig. 26 illustrate the RMSE versus K_x and K_y, respectively. The star points denote the locations of the optimal gains. It is easy to see from Fig. 25 and Fig. 26 that the gains do affect the estimation performance of the proposed method. On the other hand, the RMSE is not very sensitive to the change of gains in a certain wide region around the optimal gains, thereby allowing the optimal gains to be computed in the training phase and used to calculate the position in practical applications.

Fig. 27 depicts the RMSE of computing the position of the hand-held UXO sensor system along Y orientation versus the gains K_x and K_y. The top and bottom panels of Fig. 28 display the RMSE versus K_x and K_y, respectively. The star points denote the locations of the optimal gains. From Fig. 28 we can observe that the RMSE increases sightly only when the gains are not close to their optimal values. This implies that the gains can be determined prior to being used to calculate the position of the hand-held UXO sensor system in actual applications.

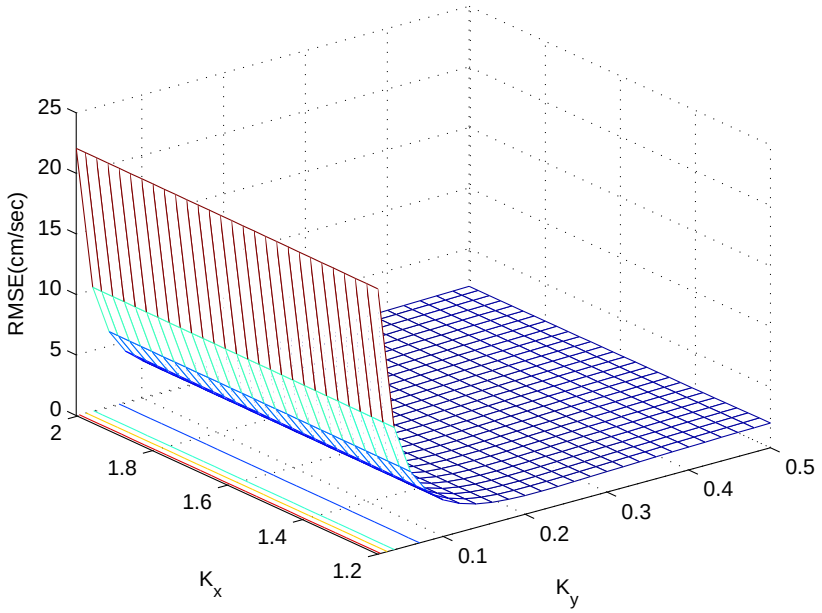


Figure 25: Root of mean square error versus gains K_x and K_y for calculating position of hand-held UXO sensor system along X orientation.

For comparison purpose, we firstly give the results of the conventional numerical integration method as baseline in Fig. 29 and Fig. 30. Since the calculation errors of this technique for the computation of velocity and position are very large, they are not plotted here. It is easy to see from Fig. 29 and Fig. 30

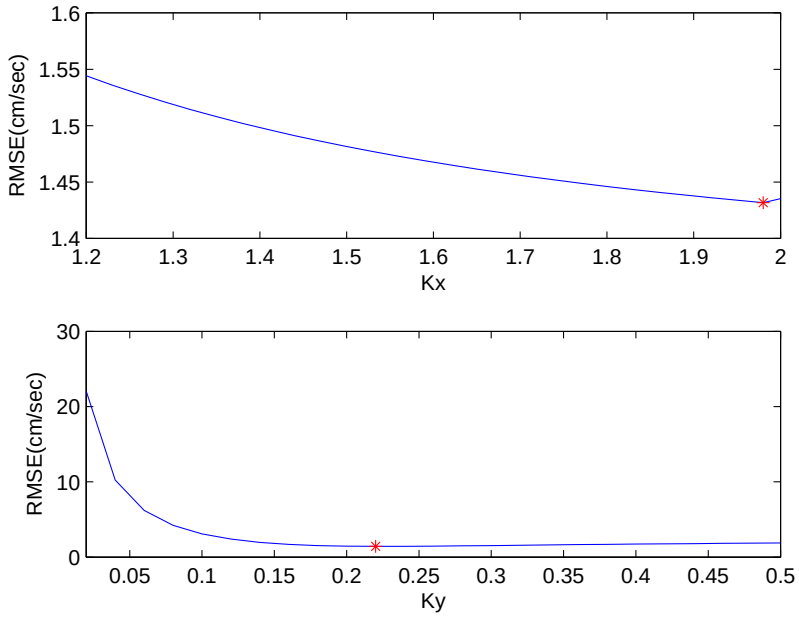


Figure 26: Top panel: Root of mean square error versus gain K_x for calculating position of hand-held UXO sensor system along X orientation. Bottom panel: Root of mean square error versus gain K_y for calculating position of hand-held UXO sensor system along X orientation.

that the conventional integration method fails to provide an efficient estimate for the positions of the hand-held UXO sensor system due to the measurement errors of sensors. Fig. 31 illustrates the velocity calculated by the iterative integration correction algorithm along X axes in the earth frame. The ground truth of the velocity V_X is also given for comparison. It is shown in Fig. 31 that the calculated velocity is very close to its ground truth. The calculation error of the velocity V_X is plotted in Fig. 32. It can be observed from Fig. 32 that the calculation error of the velocity is quite small compared to that of the baseline. Directly integrating the calculated velocity over time yields the estimate of position, which is shown in Fig. 33. For comparison, the ground truth is also plotted. From Fig. 33 we can see that the computed position is nearly completely equal to its ground truth P_X. The hand-held UXO sensor system is not very close to its truth track only when it moves in one single direction in the body frame. This lies in the fact that in this case the hand-held UXO sensor system move straightly only in the x direction. The calculation accuracy of the proposed method thereby only relies on the scheme for position calculation along x orientation, and does not improve by the scheme along y orientation. The calculation error of the position P_X is given in Fig. 34. We can see from Fig. 34 that accumulated errors are nearly completely removed and the estimation error is quite small compared to the baseline.

The velocity along the Y axis in the earth frame after correction by the proposed scheme is plotted in Fig. 35, and its calculation error is given in Fig. 36. It can be observed that the new scheme works very well and the error of the estimated velocity is small compared to the baseline. After integrating the estimated velocity V_Y over time, we readily obtain the estimated position of the hand-held UXO sensor system along Y orientation. The estimated positions along the Y axes is shown in Fig. 37. The estimation error associated with the estimated position is illustrated in Fig. 38. From Fig. 37 and Fig. 38,

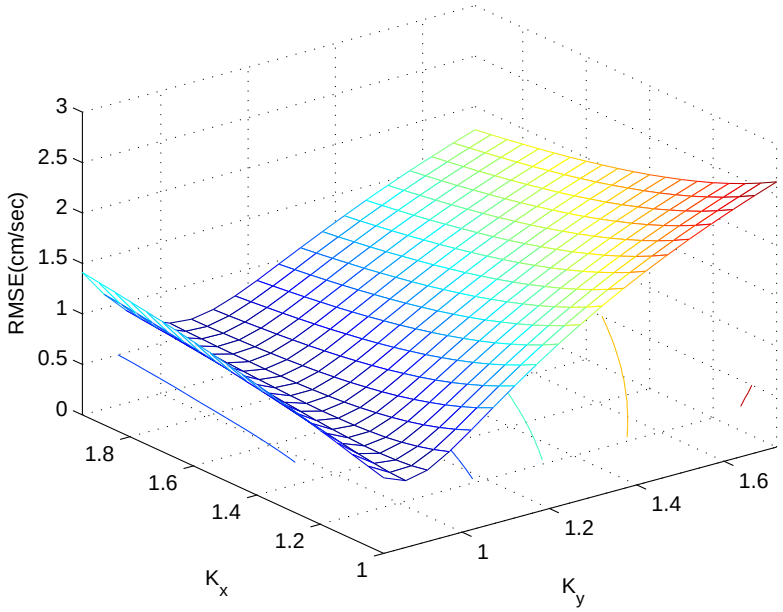


Figure 27: Root of mean square error versus gains K_x and K_y for calculating position of hand-held UXO sensor system along Y orientation.

it is easy to see that the new scheme can achieve a quite accurate estimate of the position P_Y.

Adaptive Gain Case Obviously, there is no one optimal gain for all cases. It is therefore necessary to find the corresponding optimal gains for different cases. The adaptive optimal gains are calculated by (38) and (39). In this experiment, we assume the threshold for acceleration is $\gamma_a = 3inch/sec$ and the threshold for angular rate is $\gamma_\omega = 0.2rad/sec$. In fact, it is quite difficult to determine the thresholds for acceleration and the angular rate since they rely on the training data and the actual application. Therefore, the adaptive gains presented in this experiment are optimal only for the given thresholds on acceleration and angular rate. Fortunately, we can calculate the optimal gains based on the knowledge of the actual implementation and data collected in the training phase prior to utilizing them in actual applications.

Fig. 39 and Fig. 40 display the estimated velocity and its error along X in the earth frame. The ground truth is also plotted for comparison. As is expected, the computed velocity is very close to its ground truth. This therefore indicates that the proposed method is efficient for the calculation of velocity. It is shown in Fig. 40 that the estimation error is much lower than that of the baseline shown in Fig. 30. The calculated position is given in Fig. 41 and its calculation error is plotted in Fig. 42. From Fig. 41 we can observe that the hand-held UXO sensor system is very close to its truth track except when the hand-held UXO sensor system moves linearly along x orientation. Actually, when moving in a single direction, the hand-held UXO sensor system merely has the linear motion. Therefore, the proposed method completely dependents on the calculation along x orientation in the body frame to determine the position rather than the calculations from both x and y orientations. Thus, the calculation accuracy of the proposed method in this case is not as high as the case when the hand-held UXO sensor system swings from side to side. Fig. 42 implies that the calculation error of the proposed method for the position

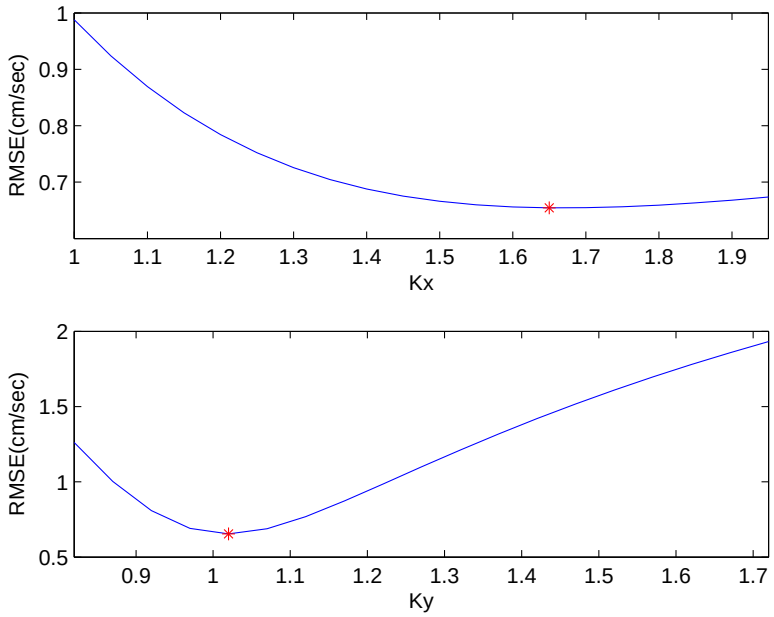


Figure 28: Top panel: Root of mean square error versus gain K_x for calculating position of hand-held UXO sensor system along Y orientation. Bottom panel: Root of mean square error versus gain K_y for calculating position of hand-held UXO sensor system along Y orientation.

along X orientation is very small and lower than that of the fixed gain case. The calculation error of the position P_X is only around 2.5cm. Comparing Fig. 34 with Fig. 42, we can find that the proposed method is more accurate and robust in the adaptive gain case than in the fixed gain case because we use two different gains for the fast and slow dynamic motions to calculate the velocities and positions and thereby can reduce the accumulated error. Note that the hand-held UXO sensor system move quite smoothly in this experiment, the improvement of the adaptive gain scheme over the fixed gains scheme is thereby not very clear.

The stabilized velocity and its calculation error along Y orientation are shown in Fig. 43 and Fig. 44, respectively. As is clearly depicted in Fig. 43, the calculated velocity is very close to its ground truth. And the calculation error of the position P_Y shown in Fig. 44 is very small compared to the baseline. The calculated position and its computed error along Y orientation are plotted in Fig. 45 and Fig. 46, respectively. For comparison purpose, we also give the ground truth of the position P_Y. Fig. 45 shows that the hand-held UXO sensor system is very close to its truth track as it swings from side to side. However, when the hand-held UXO sensor system is only in a linear motion, the calculated position is not as accurate as the case of the nonlinear dynamic motion. This is due to the fact that the proposed method only relies on the scheme for the calculation of position along x orientation, and does not benefit from the scheme for the calculation of position along y orientation. When the hand-held UXO sensor system is in a swinging dynamic motion, however, the performance of the proposed method can be improved by the two schemes for position calculation and thus achieve a more accurate estimation. Fig. 46 shows that the calculation error of the estimated position P_Y is less than 1.8cm. Comparing Fig. 38 with Fig. 46, we also can see that the proposed method is more accurate and robust in the adaptive gain case than in the fixed gain case. Again, the improvement of the proposed method in

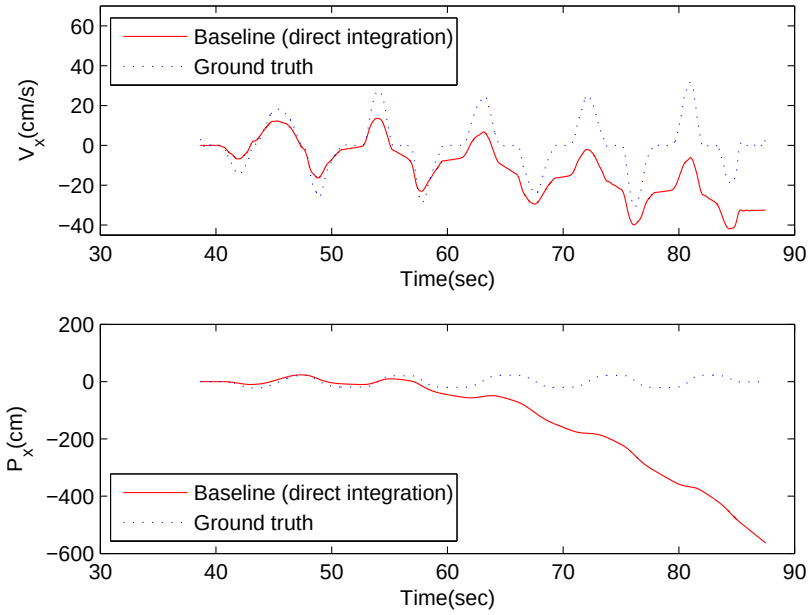


Figure 29: Calculated velocity and position along X orientation directly using the conventional integration method.

estimation performance for the adaptive gain case is due to the fact we find the more accurate gains for the fast and slow dynamic motions of the hand-held UXO sensor system. Because the motion of the hand-held UXO sensor system is relatively smooth in this experiment, the improved estimation performance of the adaptive scheme is not dramatically better than that of the fixed gain scheme.

3.6 Summary

A correction algorithm has been developed to accurately calculate the position of the hand-held UXO sensor system in this report. First, the tilt angles are computed and used to remove the effect of gravity from the acceleration measurements to obtain the true accelerations that are level relative to the earth. Utilizing the fact that the tangential velocity equals the velocity along the y orientation in the body frame and the acceleration derived from the corrected pitch angle equals that recorded by the accelerometer along the x axes in the body frame, we develop two iterative integration correction algorithms to calculate the velocities. The stabilized velocities are converted from the body frame to the earth frame by exploiting the yaw angle. Direct integration of the stabilized velocities in the earth frame leads to the corrected positions. To obtain the accurate estimate of positions, we present algorithms to calculate the fixed and adaptive gains based on the assumption of white noise. Experimental results from measured sensor data imply that the proposed method is capable of positioning the hand-held UXO sensor system. The estimation errors of positions are around 3cm along the X orientation and 2cm along the Y orientation for the fixed gains case and around 2.5cm along the X orientation and 1.8cm along the Y orientation for the adaptive gain case. Thus, the proposed method can provide the accurate calculations for the positions of the hand-held UXO sensor system.

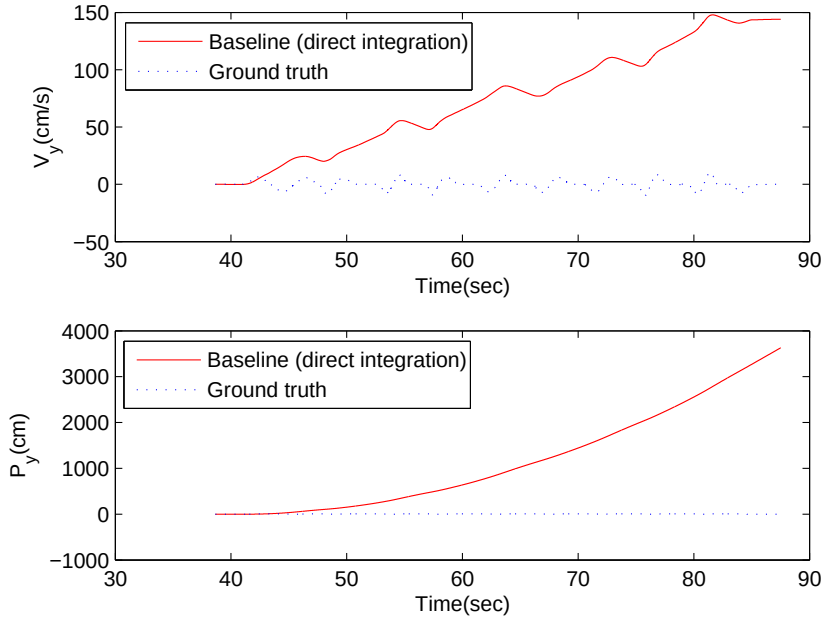


Figure 30: Calculated velocity and position along Y orientation directly using the conventional integration method.

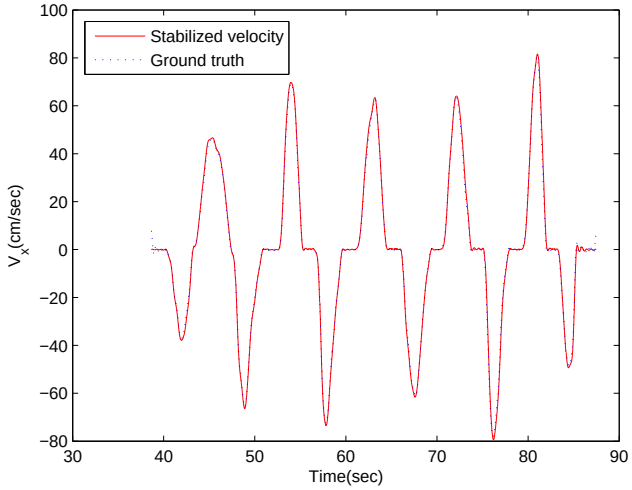


Figure 31: Stabilized velocity along X orientation with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$.

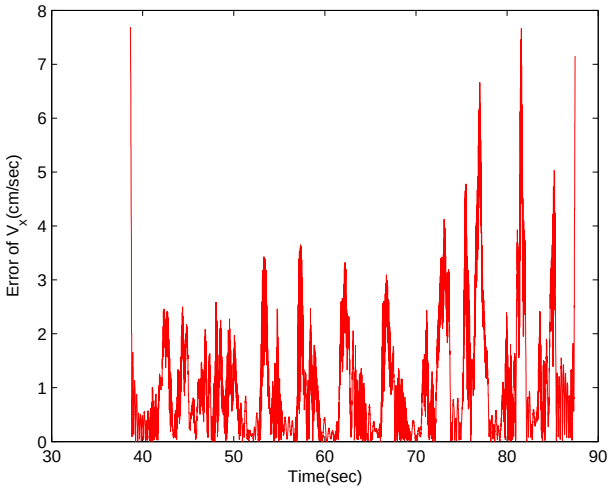


Figure 32: Calculated error of stabilized velocity along X orientation with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$.

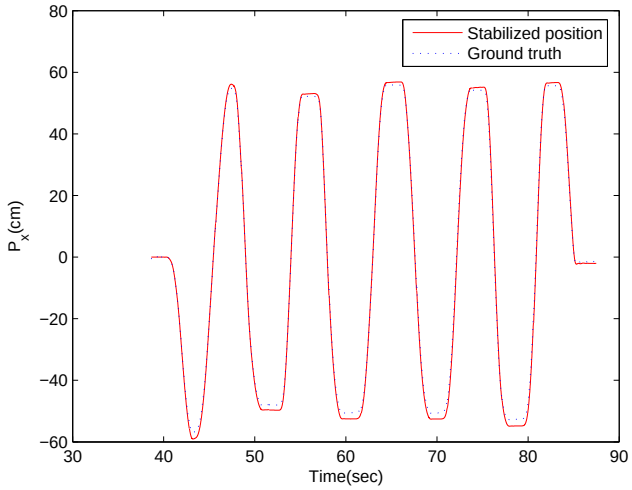


Figure 33: Stabilized position along X axes with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$.

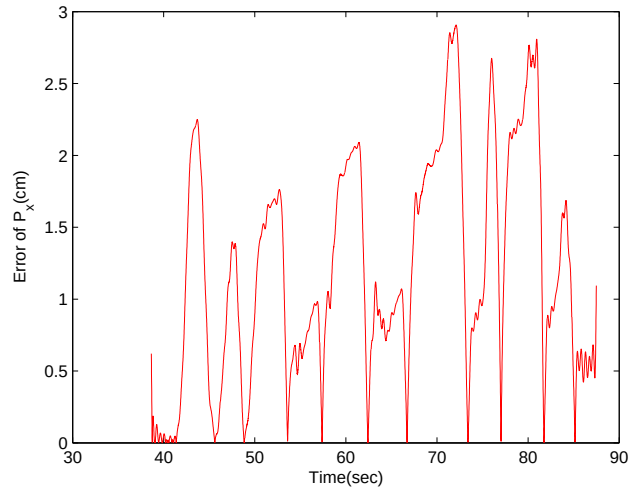


Figure 34: Calculated error of stabilized position along X axes with fixed optimal gains $K_x = 1.9800$ and $K_y = 0.2800$.

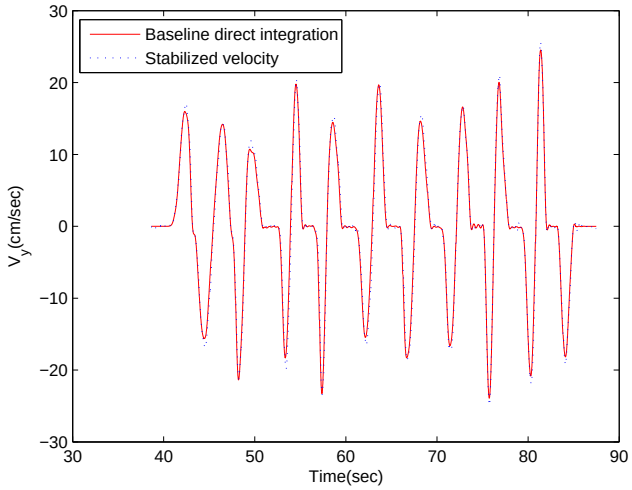


Figure 35: Stabilized velocity along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$.

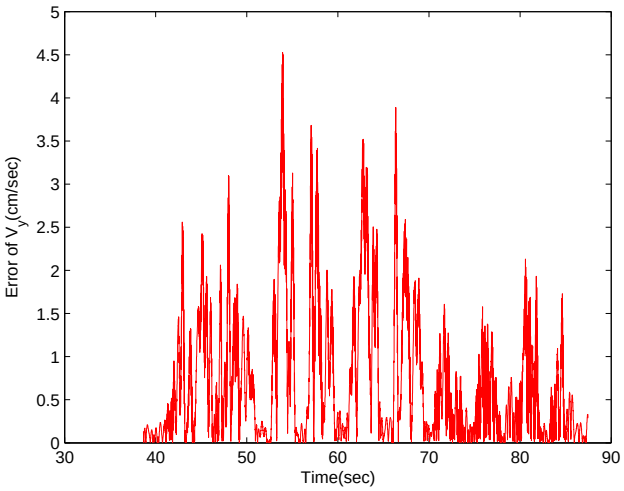


Figure 36: Calculated error of stabilized velocity along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$.

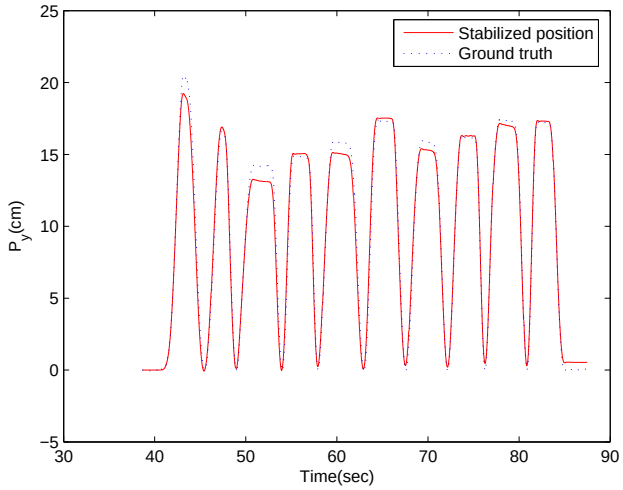


Figure 37: Stabilized position along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$.

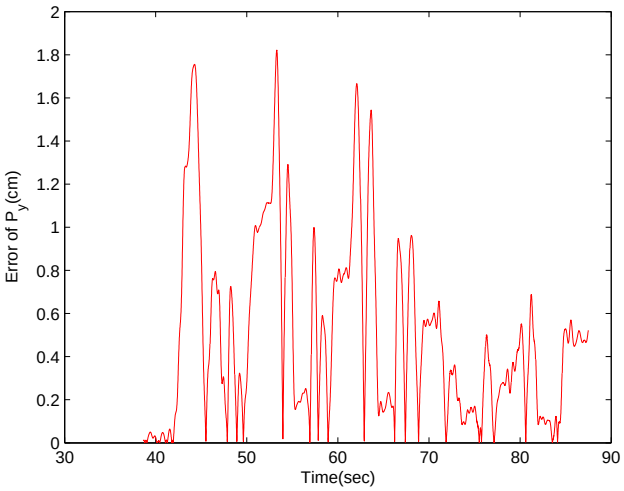


Figure 38: Calculated error of stabilized position along Y axes with fixed optimal gains $K_x = 1.9000$ and $K_y = 1.8500$.

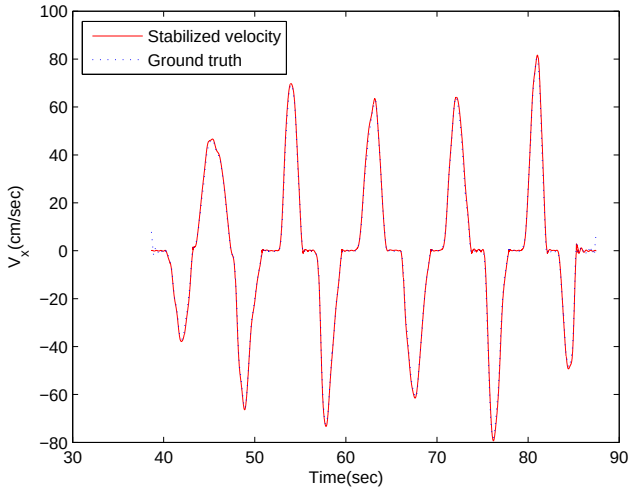


Figure 39: Stabilized velocity along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$.

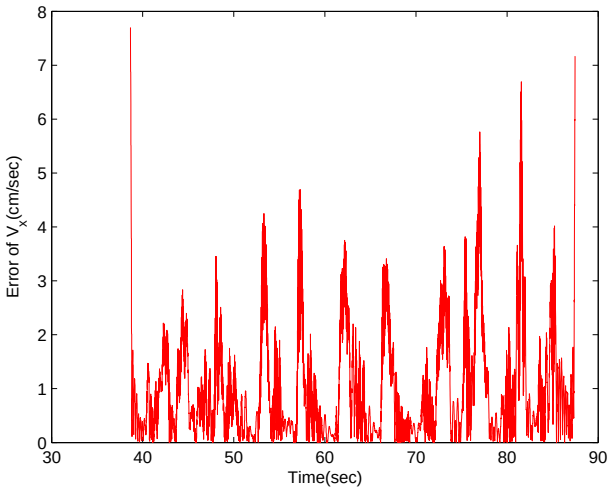


Figure 40: Calculated error of stabilized velocity along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$.

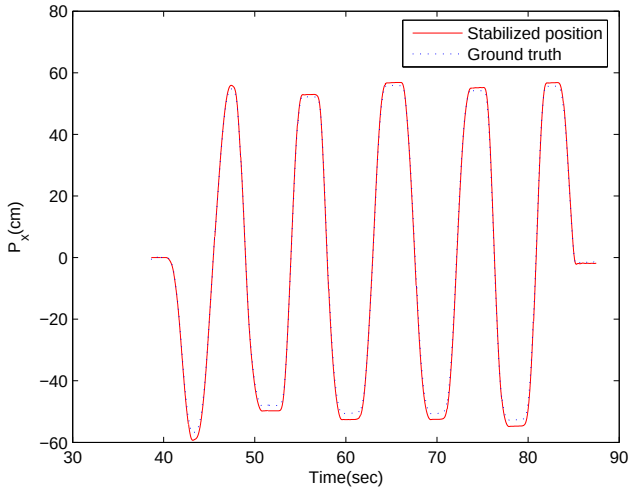


Figure 41: Stabilized position along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$.

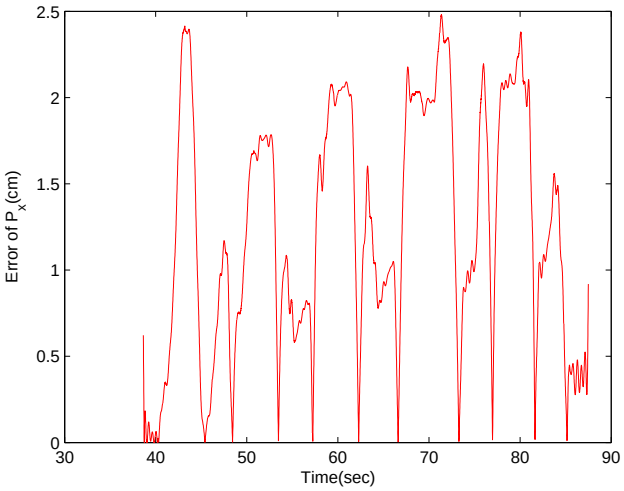


Figure 42: Calculated error of stabilized position along X axes with adaptive optimal gains $K_x^{(X),s} = 1.9950$, $K_x^{(X),f} = 1.5800$, $K_y^{(X),s} = 0.0950$ and $K_y^{(X),f} = 0.2600$.

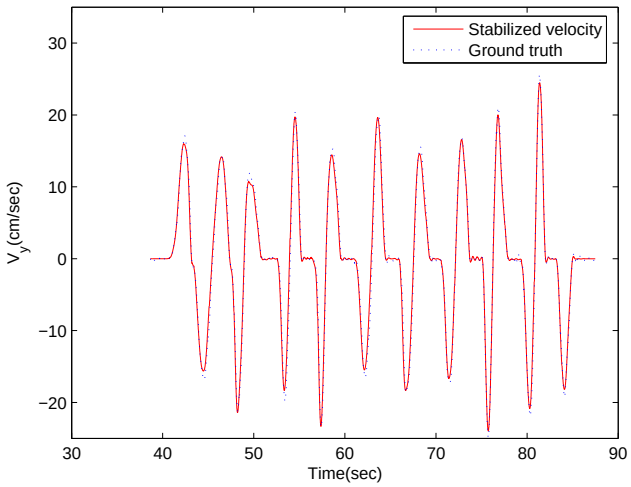


Figure 43: Stabilized velocity along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$.

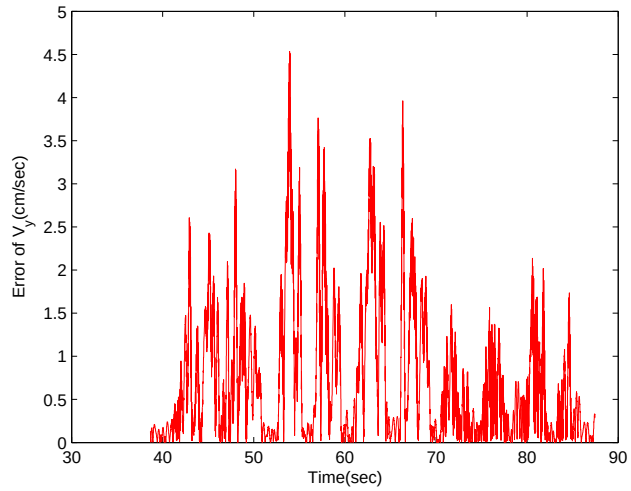


Figure 44: Calculated error of stabilized velocity along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$.

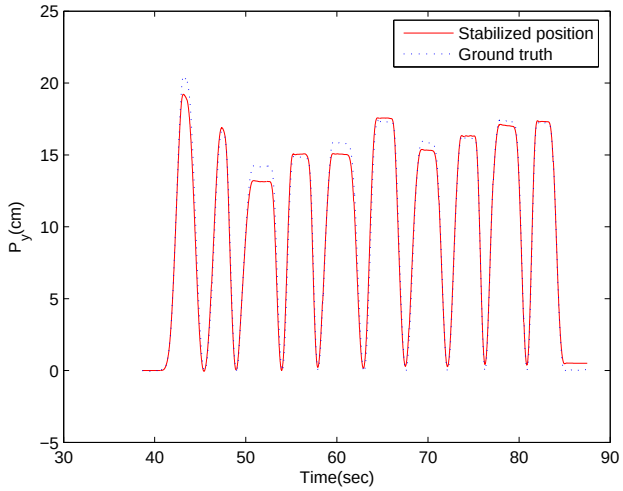


Figure 45: Stabilized position along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$.

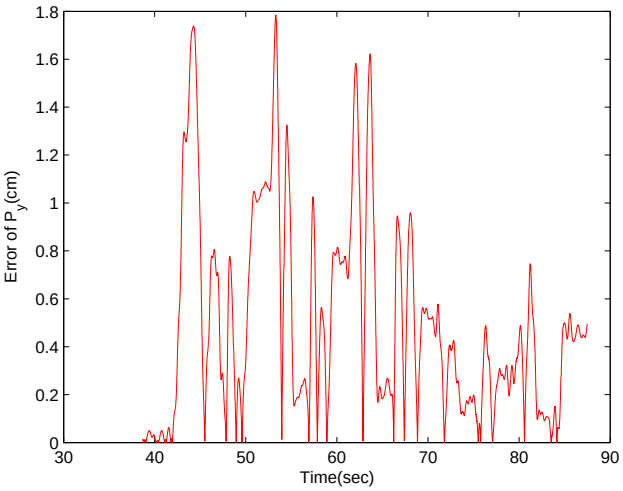


Figure 46: Calculated error of stabilized position along Y axes with adaptive optimal gains $K_x^{(Y),s} = 0.5000$, $K_x^{(Y),f} = 1.9900$, $K_y^{(Y),s} = 1.9900$ and $K_y^{(Y),f} = 1.9000$.

4 Positioning of Handheld UXO Sensor via ZUPT Algorithm

Creating zero-velocity-update points in the data stream is a topic of interest in the applications of the ZUPT algorithm. In general, the zero-velocity update is realized by simply bringing the target being tracked to a complete stop. However, since the motion of the UXO sensor is not prescribed, there is no efficient method to know the zero-velocity points in the positioning of the handheld UXO sensor. Fortunately, the IMU measurements provide information regarding the zero-velocity points. The yaw angular rate sensor provides a more reliable indicator than the accelerometer for the nonlinear motion case, and the accelerometer provides a more reliable indicator than the angular rate sensor when the IMU is moved linearly. Therefore, we can use the IMU measurements to determine the zero-velocity points and apply the ZUPT algorithm to calculate the velocities of the handheld UXO sensor.

4.1 Velocity calculation

For sweeping motion, when the yaw angular rate equals zero, the velocities should also be equal to zero. Fig. 47 gives the comparison of zero velocity points along X and Y orientations. From Fig. 47 we can observe that the yaw angular rate can provide accurate estimates of the zero velocity points for sweeping (nonlinear) motion in X and Y directions. Moreover, we also can use the measurement of the yaw angular rate to determine which motion the handheld UXO sensor is in, namely nonlinear or linear motion. More specifically, the yaw angular rate is very small for the linear motion but quite large for the nonlinear motion. The yaw angular rates for linear and nonlinear motions are shown in Fig. 48 and Fig. 49. It is easy to see in Fig. 49 that the yaw angular rate is very large for nonlinear motion. However, for linear motion, the yaw angular rate is quite small, as shown in Fig. 48.

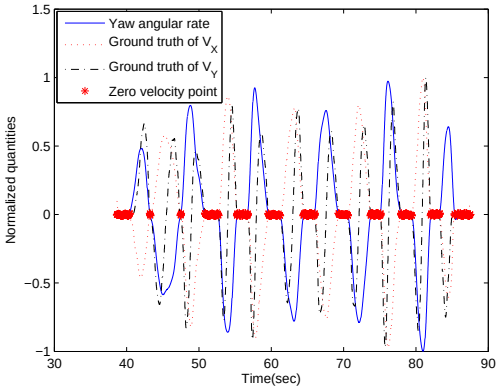


Figure 47: Comparison of zero velocity points in X and Y directions. Sweeping motion.

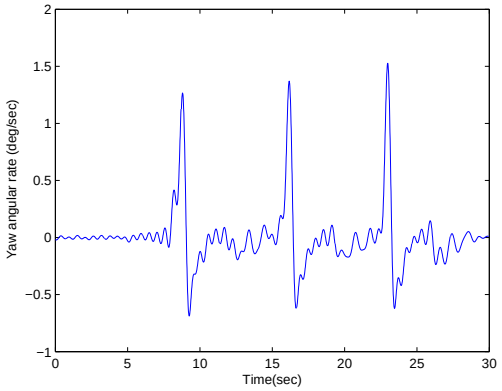


Figure 48: Yaw angular rate of linear motion of handheld UXO sensor.

With the knowledge of zero velocity points given by the yaw angular rate, we can apply the ZUPT algorithm to the estimated accelerations to obtain the velocity estimates along X and Y orientations. However, the accumulated error of velocity due to the error of acceleration in the non-zero velocity points will be still quite large. To remove the accumulated error in the non-zero velocity points, a smoothing method is employed here. Assume that a raw acceleration after removing the deterministic

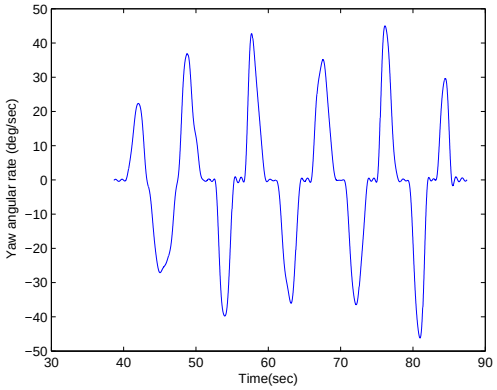


Figure 49: Yaw angular rate of nonlinear motion of handheld UXO sensor.

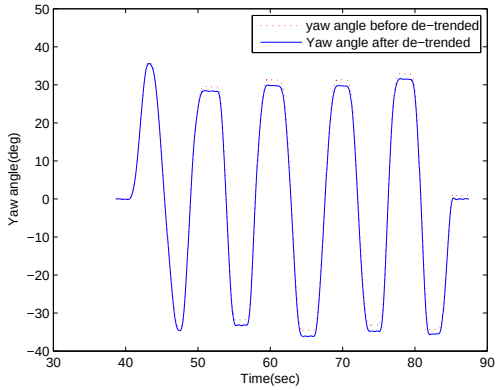


Figure 50: Yaw angle calculation before de-trended and after de-trended.

bias offset, say $\hat{a}_x$, can be written as

$$\hat{a}_x = a_x + n_{ax}. \tag{51}$$

Integrating this acceleration over the period $[\tau_1, \tau_2]$ results in a velocity estimate:

$$\hat{V}_x(\tau_2) = V_x(\tau_2) + \int_{\tau_1}^{\tau_2} n_{ax}(t)dt \tag{52}$$

Suppose $V_x(t) = 0$, for a fixed time point $t = \tau_1, \tau_2$ (zero-velocity point). Therefore, we have

$$\hat{V}_x(\tau_2) = \int_{\tau_1}^{\tau_2} n_{ax}(t)dt \triangleq \bar{n}_{ax} \triangle\tau \tag{53}$$

where $\triangle\tau = \tau_2 - \tau_1$ and $\bar{n}_{ax}$ is a constant value. From (4), it follows that

$$\bar{n}_{ax} = \frac{\hat{V}_x(\tau_2)}{\triangle\ \tau} \tag{54}$$

Thus, the smoothed acceleration is given as

$$\bar{a}_x = a_x + n_{ax} - \bar{n}_{ax} \tag{55}$$

and the calculated velocity is expressed as

$$\begin{aligned} \hat{V}_x(t) &= \int_{\tau_1}^{t} \bar{a}_x(s)ds \\ &= V_x(t) + \int_{\tau_1}^{t} n_{ax}(s)ds - \bar{n}_{ax}\cdot(t-\tau_1), \quad t = [\tau_1, \tau_2] \end{aligned} \tag{56}$$

From (4) and (7), we see that when $t = \tau_1, \tau_2$, $\hat{V}_x(t) = 0$. When $\tau_1 < t < \tau_2$, $\hat{V}_x(t) = V_x(t) + e_x(t)$, where $e_x(t) = \int_{\tau_1}^{t} n_{ax}(s)ds - \bar{n}_{ax}\cdot(t-\tau_1)$. Obviously, $e_x(t)$ is a small value for a reasonably short time t.

To obtain the true velocity in the earth frame, we must transform the calculated velocity from the body frame to the earth frame. This transformation requires the yaw angle, which is obtained by integrating the yaw angular rate over time. Though the yaw angular rate is quite accurate for sweeping

motion, the calculation of the yaw angle may be not accurate enough for an integration over a long period. Normally, after low-pass filtering, the measured rate still needs to be de-trended to avoid a large drift error in the yaw angle calculation. Fig. 50 illustrates the yaw angle calculations before de-trending and after de-trending. From Fig. 50 we can see that there is not a large drift error in the yaw angle calculation after de-trending.

After calculating the yaw angle, we can use it to transform the velocities from the body frame to the earth frame. The relation between the body frame and the earth frame is given in Fig. 51. In the following, the velocities in the earth frame can be written as

$$\hat{V}_X = \hat{V}_y \cos(\hat{\theta}_z) - \hat{V}_x \sin(\hat{\theta}_z) \quad (57)$$

$$\hat{V}_Y = \hat{V}_y \sin(\hat{\theta}_z) + \hat{V}_x \cos(\hat{\theta}_z) \quad (58)$$

where $\hat{\theta}_z = \int \hat{w}_z dt$ denotes the calculated yaw angle, $\hat{V}_x$ and $\hat{V}_y$ are the estimated velocities in the body frame, $\hat{V}_X$ and $\hat{V}_Y$ are the estimated velocities in the earth frame.

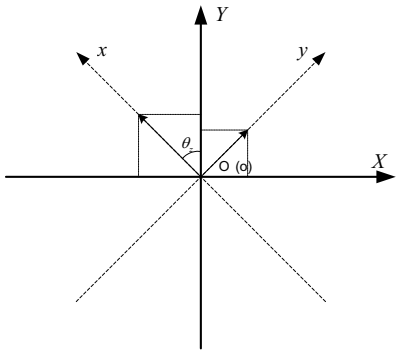


Figure 51: Geometry of earth frame and body frame. *xoy* is the coordinate system of the body frame and *XOY* is the coordinate system of the earth frame.

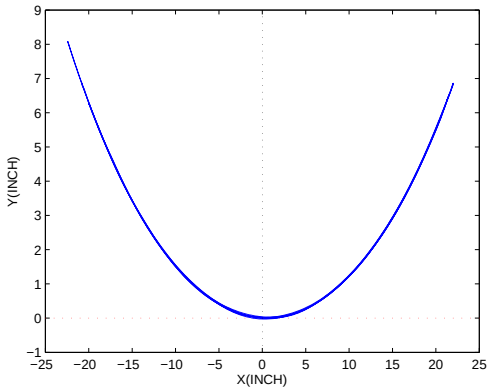


Figure 52: Example of zero-position-update points. The solid lines are the actual track of the sensor. The dash lines are the coordinate axes of the earth frame.

4.2 Position calculation

Obviously, directly integrating the estimated velocities over time will produce position estimates. However, if the calculated velocities are not accurate enough, the final position error will be still quite large. In addition to the zero-velocity points, there are also many zero-position points at which the x-coordinate is equal to 0 in our applications. An example of zero-position points is shown in Fig. 52. To further reduce the position estimation error, we use the zero-position-update point to improve the performance of positioning the handheld UXO sensor. Notice that when the yaw angle is zero, the positions of the handheld UXO sensor are also exactly equal to zero for the sweeping motion in both X and Y directions. Thereby, if the yaw angle can be accurately calculated, the zero-position points can also be accurately determined. In the following, the smoothing method presented in the previous subsection for velocity calculation can also be employed here to enhance the position calculation. Thus, we first utilize the

calculated yaw angle to determine the zero-position points, then use the smoothing method to accurately calculate the position of the handheld UXO sensor. The scheme for calculating the position of the handheld UXO sensor is illustrated in Fig. 4.2.

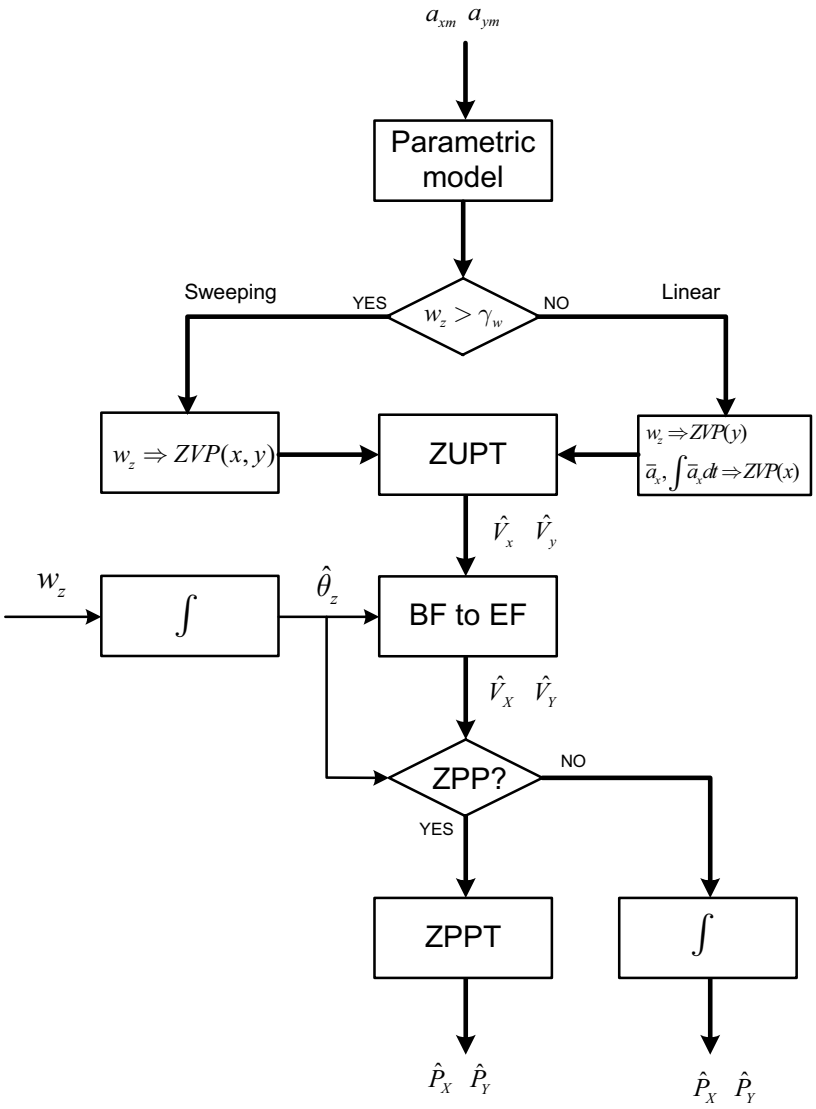


Figure 53: Scheme for positioning the handheld UXO sensor. ZVP denotes zero-velocity points. ZPP is zero-position points. ZPPT represents the zero-position-update algorithm.

4.3 Experimental Results

The nonlinear measurements consist of the handheld UXO sensor being swept in an arc, similar to a motion an operator may make with it. The linear measurements consists of the sensor moving back and forth in a straight line. The raw measurements of the IMU are low-pass filtered and D/A converted. The

deterministic bias offset is then removed from the measurements by calibration. Since the sample times for the ground truth and IMU measurements are not identical, the ground truth is interpolated to the sample times for the IMU measurements.

4.3.1 Results of sweeping motion

For comparison purpose, we use the results of the conventional numerical integration method as a baseline, which are shown in Fig. 54 and Fig. 55. It can be observed from Fig. 54 and Fig. 55 that the direct integration technique fails to correctly calculate velocities and positions, even though the deterministic bias offset has been removed from the measurements. The following results are calculated from the estimated accelerations, which are obtained by applying the estimated parametric model to the raw accelerations. The zero-velocity points are determined by searching the points in the yaw angular rate, which are nearly or completely equal to zero. Similarly, the zero-position points are obtained by searching the points of the yaw angle calculation that equal or are close to zero.

The calculated velocity in the X direction is shown Fig. 56. The ground truth is also plotted for comparison. From Fig. 56 we can see that the calculated velocity is very close to its ground truth, thereby indicating that the ZUPT algorithm can accurately compute the velocity from the estimated acceleration if the zero-velocity points are also accurately determined. However, the ZUPT algorithm fails to correctly calculate the velocity from the raw acceleration, even though the zero velocity points have been accurately obtained. This is clearly shown in Fig. 58. Note that the ZUPT algorithm will not yield any calculated error at the zero velocity points, but will produce the error at the nonzero velocity points. The velocities in the Y direction are plotted in Fig. 57 and Fig. 59, which are calculated from the estimated acceleration and the raw acceleration, respectively. It is indicated in Fig. 57 and Fig. 59 that the ZUPT algorithm can accurately calculate the velocity from the estimated acceleration but fail to correctly compute it from the raw acceleration, even though the zero-velocity points are well determined.

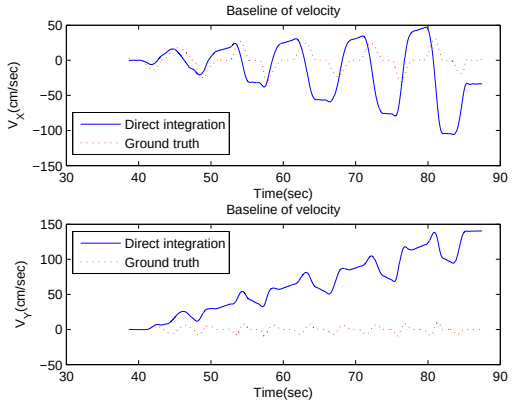


Figure 54: Velocity calculated from raw acceleration by the conventional numerical integration. Sweeping motion.

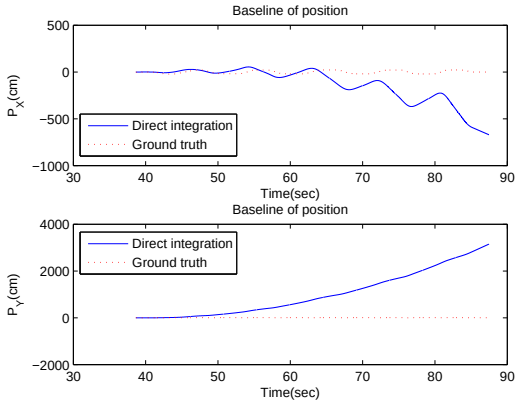


Figure 55: Position calculated from raw acceleration by the conventional numerical integration. Sweeping motion.

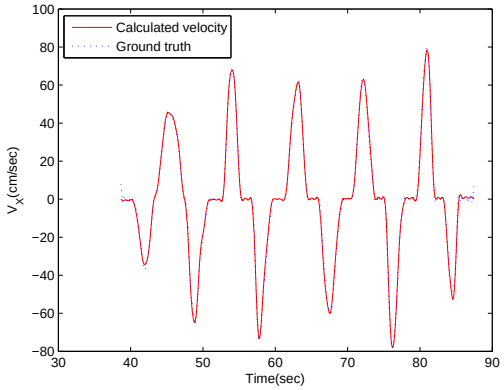


Figure 56: Calculated velocity along X orientation based on estimated acceleration. Sweeping motion.

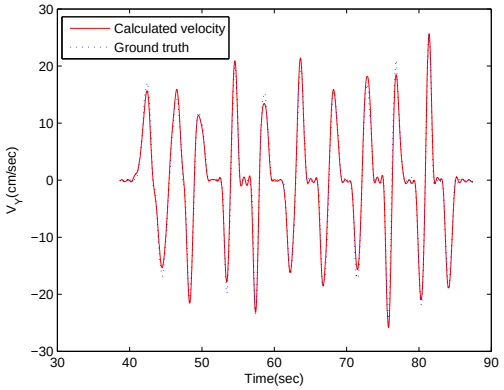


Figure 57: Calculated velocity along Y orientation based on estimated acceleration. Sweeping motion.

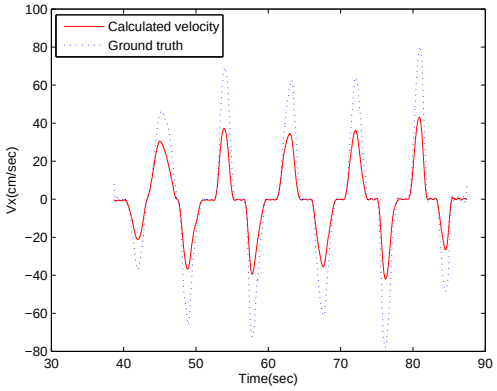


Figure 58: Calculated velocity along X orientation based on raw acceleration. Zero velocity points are accurately determined. Sweeping motion.

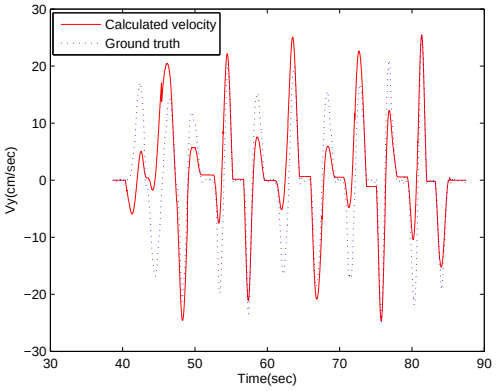


Figure 59: Calculated velocity along Y orientation based on raw acceleration. Zero velocity points are accurately determined. Sweeping motion.

To demonstrate the role of the zero-position-update method in improving the position calculation, we also give the results of directly integrating the estimated velocity. The position directly calculated by the numerical integration technique along the X orientation is plotted in Fig. 60, and its error is shown in Fig. 61. Meanwhile, the position calculated by the zero-position-update method is given in Fig. 62 and its calculation error is plotted in Fig. 63. Comparing Fig. 61 and Fig. 63 we can see that the results of the zero-position-update method are more accurate than that of the direct integration. The maximum error of the estimated position is only around 2.5cm in X direction for the zero-position-update method, which is much less than that of the direct integration technique. Fig. 64 and Fig. 65 show the results of the direct integration method. Fig. 66 and Fig. 67 give the position and calculation error of the zero-position-update method along Y orientation. From Fig. 65 and Fig. 67, we also can see that the zero-position-update method can further improve the accuracy of positioning the handheld UXO sensor in the Y direction. The maximum error of position calculation is less than 2cm.

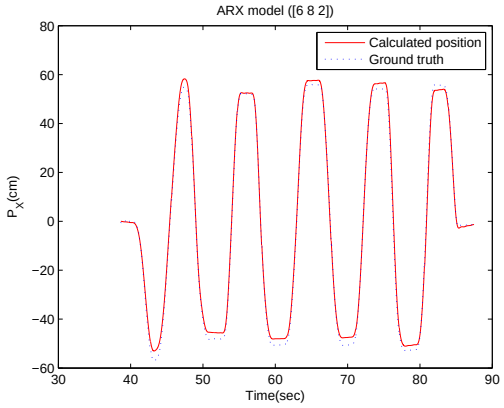


Figure 60: Position along X orientation calculated from the estimated velocity by direct integration method. Sweeping motion.

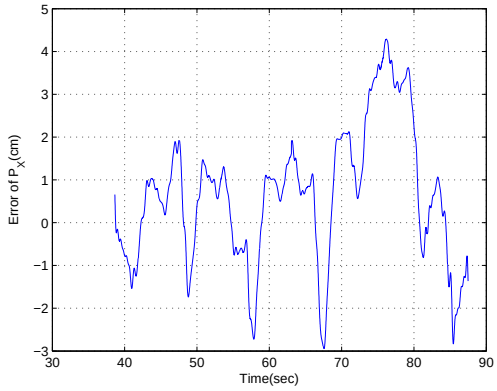


Figure 61: Position error along X orientation calculated from the estimated velocity by direct integration method. Sweeping motion.

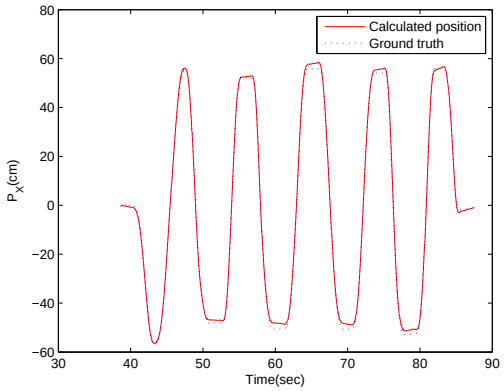


Figure 62: Position along X orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion.

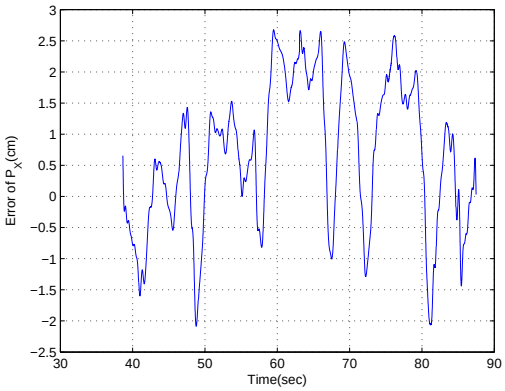


Figure 63: Position error along X orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion.

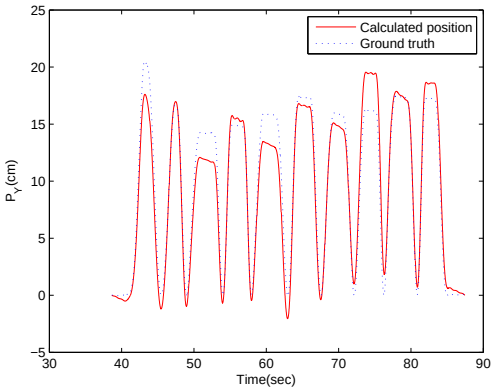


Figure 64: Position along Y orientation calculated from the estimated velocity by direct integration method. Sweeping motion.

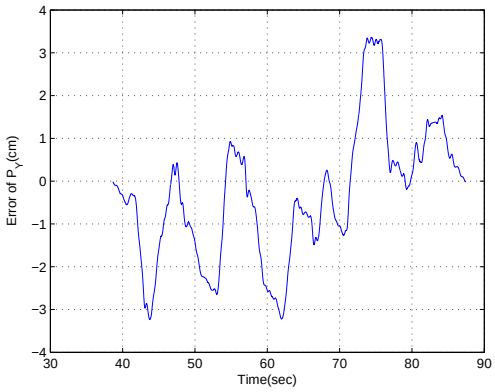


Figure 65: Position error along Y orientation calculated from the estimated velocity by direct integration method. Sweeping motion.

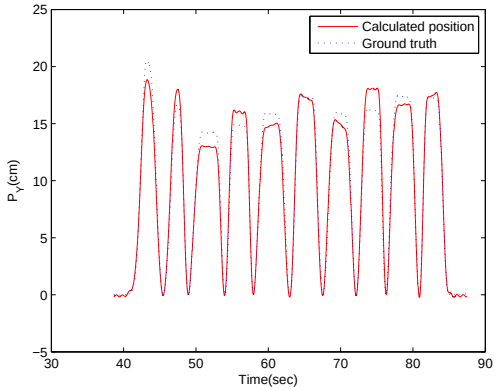


Figure 66: Position along Y orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion.

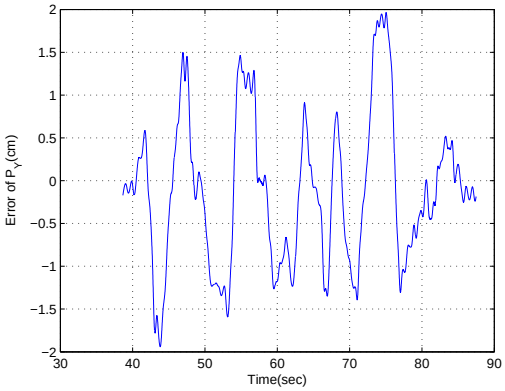


Figure 67: Position error along Y orientation calculated from the estimated velocity by zero-position-update method. Sweeping motion.

4.3.2 Results of linear motion

The sweeping motion creates a scenario in which the angular rate sensor has a quantity to measure. Thus, the angular rate data can provide accurate information regarding the zero-velocity points. However, for linear motion, the yaw angular rate is very small and will not change with the linear motion of the handheld UXO sensor. Therefore, the zero-velocity points cannot be determined merely by the yaw angular rate. Fortunately, the accelerometer measures a significant quantity for linear motion. Therefore, we can determine the zero-velocity points by comparing the acceleration and its integration quantity. Ideally, The time points at which both the acceleration and its integration are equal to zero can be considered as the zero-velocity points. Since the acceleration is reasonably accurate for this purpose, its integration will not lead to a large drift error after de-trending. Thus, the zero-velocity points can be determined directly from the measured acceleration. The method for determining the zero-velocity points is illustrated in Fig. 4.4, in which the circle points are potential zero-velocity points obtained from the acceleration, the star points are the possible zero-velocity points determined by the calculated velocity. By comparing the circle points with the star points, we can determine the zero-velocity points accurately. On the other hand, the zero-velocity points along the Y orientation are still obtained by the yaw angular rate since angular rate sensor is more sensitive to the dynamic motion of the handheld UXO sensor than the accelerometer. Fig. 4.4 give the ground truth of linear motion. We can observe from Fig. 4.4 that there exist many zero-position points in the linear motion. Thereby, we can employ these zero-position points to further improve the accuracy of position calculation.

For comparison, we plot the baselines in Fig. 70 and Fig. 4.4, which are obtained by directly integrating the raw measurements over time, one time for velocity calculation and twice for position computation. It can be observed from Fig. 70 and Fig. 4.4 that the numerical integration method fails to correctly calculate the velocities and positions for the linear motion case. However, with accurate estimates of acceleration and zero-velocity points, the ZUPT algorithm can accurately compute the velocities along X and Y orientations, which are shown in Fig. 72 and Fig. 73. The angular rate is utilized to determine the zero-position points in the X direction. However, the zero-position points cannot be obtained from the angular rate along the Y orientation, and thereby the ZPPT method cannot be employed to compute the position P_Y. Fortunately, since the velocity can be accurately calculated from the accurate measured acceleration, directly integrating the estimated velocities V_Y over time leads to an accurate position estimation in Y direction. The estimated positions in X and Y directions are plotted in Fig. 74 and Fig. 75, respectively. Fig. 74 and Fig. 75 indicate that the positions can be accurately computed by the proposed scheme. Accordingly, the calculated position errors are given in Fig. 76 and Fig. 77. It is shown in Fig. 76 and Fig. 77 that the calculated position errors are quite small, only around 0.45cm in X direction and around 2.5cm along Y orientation.

4.4 Summary

This report presented an efficient method for positioning the handheld UXO sensor. This method proceeds in four steps: First, determining a dynamical system model between the measured acceleration and its ground truth in the training phase and then apply this model to the measurements to improve the acceleration estimates. Second, exploiting the yaw angular rate to determine the zero-velocity points for nonlinear motion, and the acceleration and its integration quantity to find the zero-velocity points for linear motion, and then implement the ZUPT algorithm to calculate the velocities. Third, the calculated velocities are transformed from the body frame to the earth frame. Finally, since there are many zero-position points in the nonlinear motion of the handheld UXO sensor, which can be determined by the yaw angle, similarly to the ZUPT algorithm, we employ these zero-position points to further improve

the position accuracy. Experimental results obtained with the measured laboratory quality data indicate that the method is capable of accurately positioning the handheld UXO sensor.

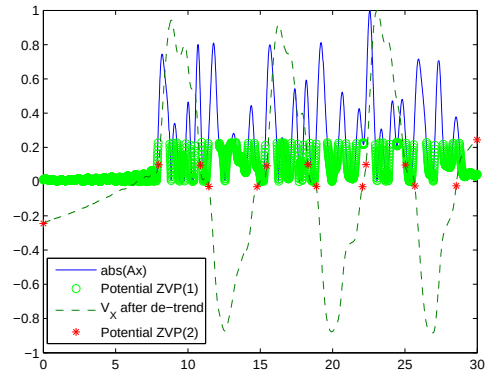


Figure 68: Searching for zero velocity points by relying on acceleration in linear motion. The zero velocity points are the green circle points that are between the red star points.

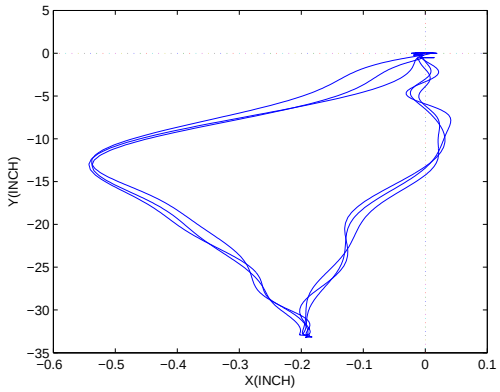


Figure 69: Example of zero-position updates for linear motion. The solid lines are the actual track of the sensor. The dash lines are the coordinate axes of the earth frame.

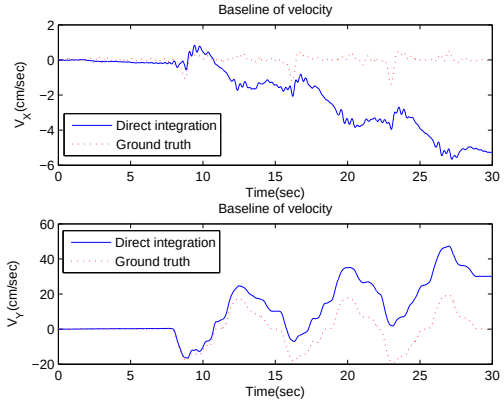


Figure 70: Velocity calculated from raw acceleration by the conventional numerical integration. Linear motion.

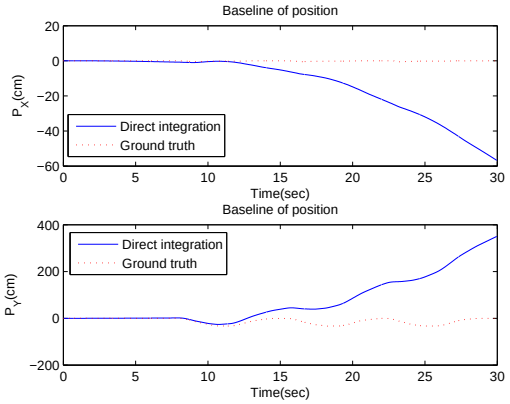


Figure 71: Position calculated from raw acceleration by the conventional numerical integration. Linear motion.

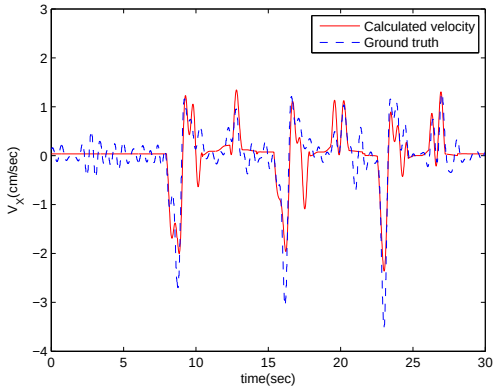


Figure 72: Calculated velocity along X orientation based on estimated acceleration. Linear motion.

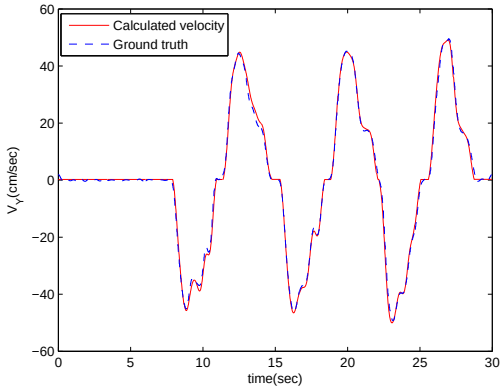


Figure 73: Calculated velocity along Y orientation based on estimated acceleration. Linear motion.

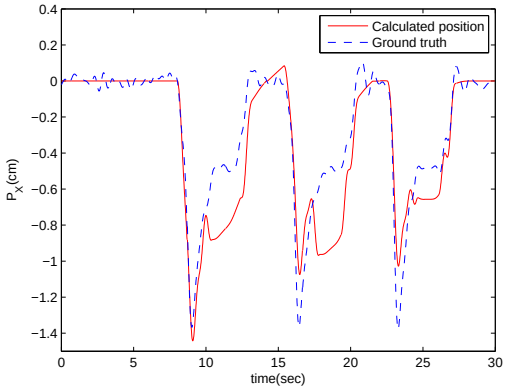


Figure 74: Calculated position along X orientation based on estimated acceleration. Linear motion.

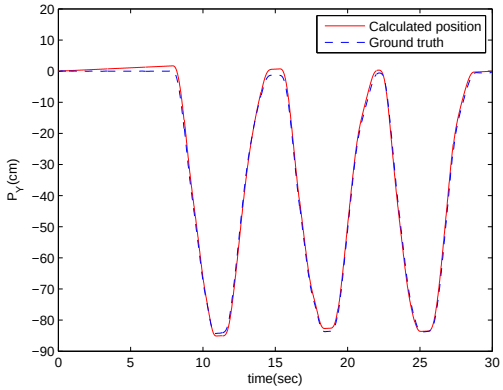


Figure 75: Calculated position along Y orientation based on estimated acceleration. Linear motion.

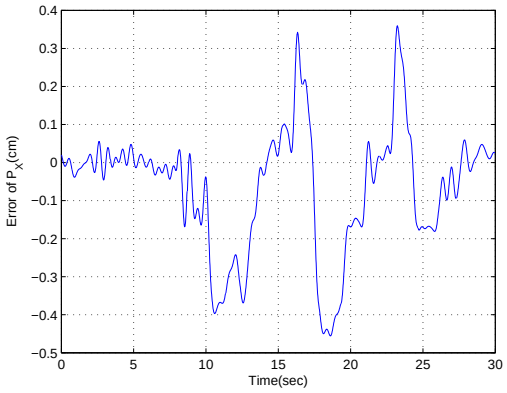


Figure 76: Calculated position error along X orientation based on estimated acceleration. Linear motion.

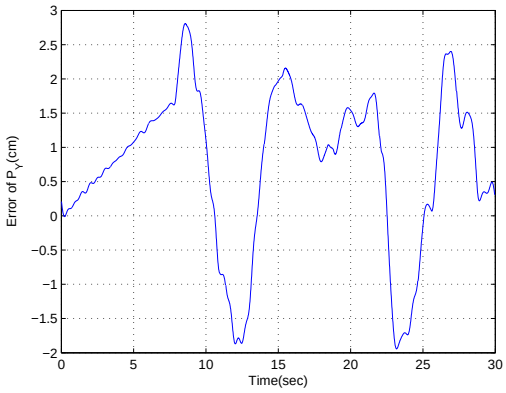


Figure 77: Calculated position error along Y orientation based on estimated acceleration. Linear motion.

5 System Modeling for Improved Acceleration Estimation

In this section, system identification techniques are applied to improve the acceleration estimates. As mentioned in the previous reports, the errors of the raw acceleration measurements generally consist of a deterministic bias offset and a stochastic term. The deterministic bias offset can be calculated and subtracted from the raw measurement. The random noise, however, cannot be removed or reduced by a calibration procedure. The accuracy of positioning the handheld UXO sensor is expected to be improved by improving the acceleration estimate.

Consider the system identification problem shown Fig. 5. In general, system identification methods are classified into two types: nonparametric identification methods and parametric identification methods. Nonparametric methods typically involve correlation analysis, which estimates a system's impulse response, and spectral analysis, and calculates a system's frequency response. Parametric methods generally include determining a suitable polynomial model relating the input to the output and estimating the model parameters. In this report, only parametric methods are considered.

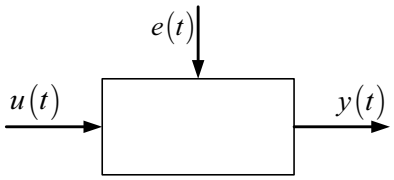


Figure 78: Black box structure. $u(t)$ is the input, $y(t)$ denotes the output and $e(t)$ represents the model error.

5.1 Parametric methods for system identification

The most general polynomial model is given by

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t) \tag{59}$$

where $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ are polynomials in which q denotes the time-domain shift operator, *e.g.* $q^{-1}u(t) = u(t-1)$. In practice, this structure is often too general and one or several polynomials are often set to zero or unity. The polynomials in 59 are expressed in the following long-hand and short-hand notations as

$$\begin{aligned}
A(q) &= 1 + \sum_{i=1}^{na} a_i q^{-i} = [1, a_1, \cdots, a_{na}] \\
B(q) &= b_0 + \sum_{i=1}^{nb} b_i q^{-i} = [1, b_1, \cdots, b_{nb}] \\
C(q) &= 1 + \sum_{i=1}^{nc} c_i q^{-i} = [1, c_1, \cdots, c_{nc}] \\
D(q) &= 1 + \sum_{i=1}^{nd} d_i q^{-i} = [1, d_1, \cdots, d_{nd}] \\
F(q) &= 1 + \sum_{i=1}^{nf} f_i q^{-i} = [1, f_1, \cdots, f_{nf}]
\end{aligned}$$

By fixing one or some polynomials above to zero or unity, the structures listed in Table 1 are obtained, in which ARMAX means autoregressive moving average with exogenous inputs, ARX means autoregressive with exogenous inputs, OE means output-error, ARMA means autoregressive moving average and AR means autoregressive. Among these structures, only ARX, ARMAX and OE models are suitable for this application. The ARX and ARMAX models have some advantages over the other models. The estimation of the ARX model is the most efficient of the polynomial estimation methods because it is the result of solving linear regression equations in analytic form. In addition, the solution is unique, meaning the solution always satisfies the global minimum of the loss function. Compared with the ARX model, the ARMAX model provides more flexibility for the stochastic dynamics. However, the ARMAX model and the OE model involve iterative, nonlinear optimization in the identification procedure. They require excessive computation time and the minimization can get stuck at a false local minimum, especially when the order is high and the signal-to-noise ratio is low. Therefore, we only focus on the ARX model in this report. The comparison between the ARX model and the ARMAX model will be given in the next subsection.

Model	Equations
ARMAX	$A(q)y(t) = B(q)u(t) + C(q)e(t)$
ARX	$A(q)y(t) = B(q)u(t) + e(t)$
OE	$F(q)y(t) = B(q)u(t) + F(q)e(t)$
ARMA	$A(q)y(t) = C(q)e(t)$
AR	$A(q)y(t) = e(t)$

Table 1: Polynomial models.

5.2 Model parameter estimation and its applications to acceleration enhancement

As mentioned previously, the input of the black box may be the acceleration, its difference and integration. Accordingly, the output should be their desired quantities. All the possible input/output combinations are considered and their results are compared to determine which case is the most suitable for this application.

These models can be easily determined by the system identification toolbox in Matlab. The order of the ARX model is determined by the Akaike information criterion (AIC), and the parameters of the ARX model are calculated in the least squares sense. Fig. 79 and Fig. 80 give the estimated accelerations for the case where the input of the black box is the measured acceleration while the output is the difference of its ground truth. In this report, some block of the ground truth, namely samples from 1000 to 4500, is used as the training data. The total number of samples in the measurements is 5700. The lower case $\{x, y, z\}$ denotes the axis directions of the body frame and the upper case $\{X, Y, Z\}$ represents the axis directions of the earth frame. From Fig. 79, it can be observed that the measured acceleration along the x orientation is nearly completely different from its ground truth. However, the estimate is very close to its ground truth. Especially for the acceleration along y orientation, its estimate is almost completely equal to its ground truth and much more accurate than its measurement. See Fig. 80, where [8 7 2] represents $na = 8$, $nb = 7$ and $nk = 2$, and nk denotes the delay from the input to the output. However, for the cases where the input is the measured acceleration while the output is true acceleration or velocity, the results are poorer than that of Fig. 79 and Fig. 80, which are plotted in Fig. 81 – Fig. 84. These results indicate that the first case is the most suitable among them, in which the measurement is the input while the difference of its ground truth is the output. This is due to the fact that in this case the dynamical system models the difference relation between the input and the output, thereby implying there is no

accumulated error from the input to the output. Thus, the model is more accurate to determine for this case than for the other cases.

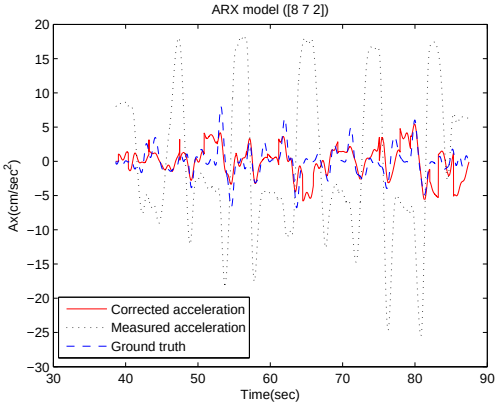


Figure 79: Estimated acceleration along x orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 8$, $nb = 7$ and $nk = 2$.

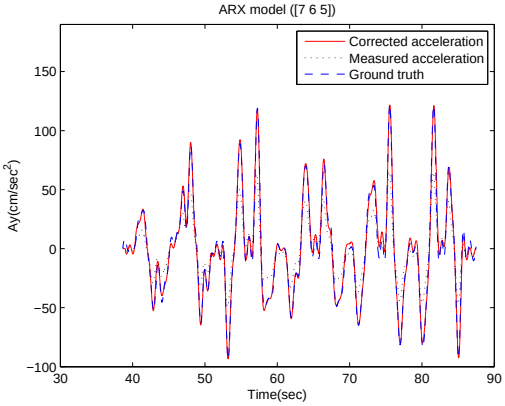


Figure 80: Estimated acceleration along y orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 7$, $nb = 6$ and $nk = 5$.

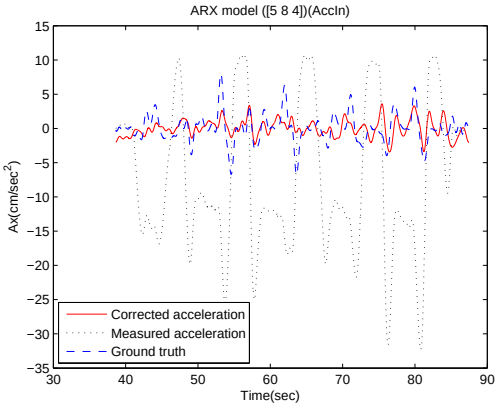


Figure 81: Estimated acceleration along x orientation. a_{meas} is input and a_{true} is output. $na = 6$, $nb = 1$ and $nk = 1$.

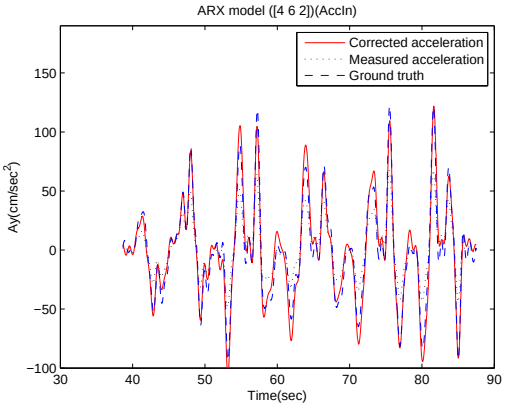


Figure 82: Estimated acceleration along y orientation. a_{meas} is input and a_{true} is output. $na = 4$, $nb = 6$ and $nk = 2$.

Normally, the ARMAX model is determined by trial and error. At first, we should employ correlation analysis to determine the delay from the input to the output of the dynamical system. Secondly, we obtain useful information about the model order by observing the the number of resonance peaks in the nonparametric frequency response function. Generally, the number of peaks in the magnitude response equals half the order of $A(q)$. Finally, we use the method of trial and error to find the suitable orders for the polynomials $B(q)$ and $C(q)$ by checking the autocorrelation function (ACF) and partial autocorrelation function (PACF). Fig. 85 gives the results of correlation analysis and spectral analysis.

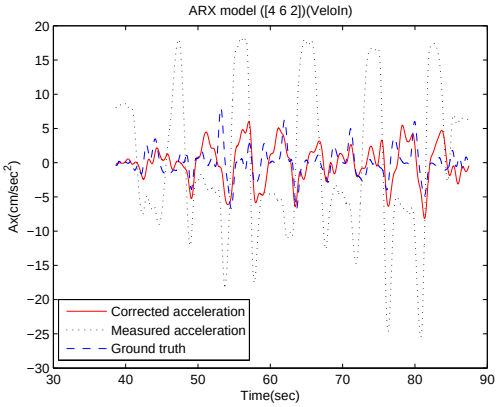


Figure 83: Estimated acceleration along x orientation. a_{meas} is input and v_{true} is output. $na = 4$, $nb = 6$ and $nk = 2$.

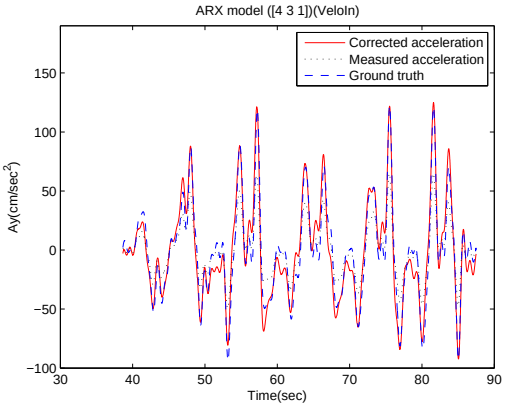
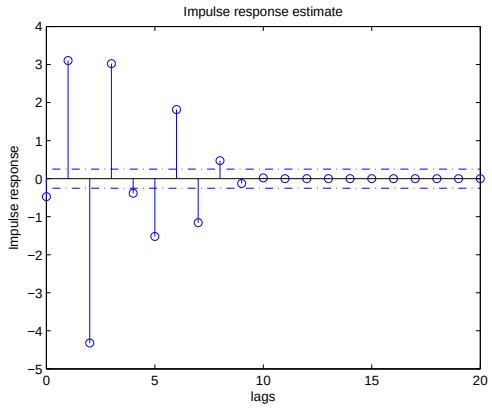


Figure 84: Estimated acceleration along y orientation. a_{meas} is input and v_{true} is output. $na = 4$, $nb = 3$ and $nk = 1$.

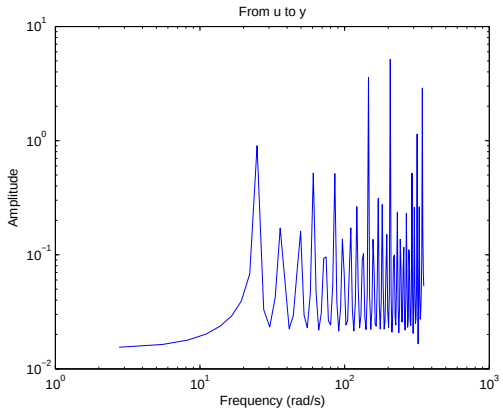
The estimated accelerations along x and y orientations are shown in Fig. 86 and Fig. 87, respectively. It is indicated in Fig. 86 and Fig. 87 that the accelerations also can be accurately estimated by correctly establishing a ARMAX model. However, by comparing Fig. 79 and Fig. 80 with Fig. 86 and Fig. 87, we can see the ARMAX model has no advantage over the ARX model in the estimation accuracy of acceleration. Furthermore, the determination of the order of the ARMAX model is more difficult than that of the ARX model.

5.3 Summary

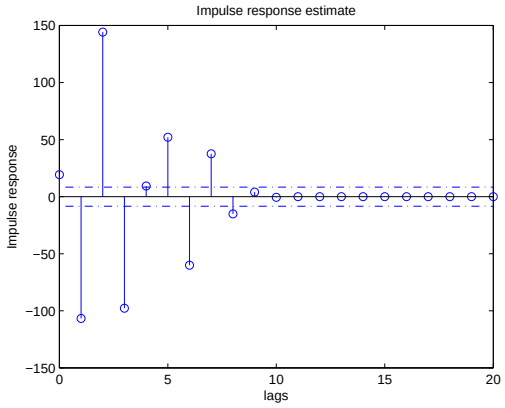
System identification has been explored as a method to mitigate IMU measurement errors. After determining a system model that relates the measured data to the ground truth, the effects of the IMU system can be reversed so that the original data is recovered. This is similar to identifying an unknown filter and then applying an inverse filter to obtain the original data. This approach shows promise for reducing IMU measurement errors



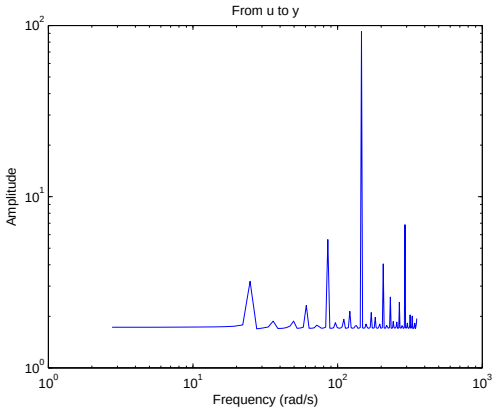
(a)*x* orientation.



(b)*x* orientation



(c)*y* orientation



(d)*y* orientation

Figure 85: Correlation analysis and spectral analysis for ARMAX model.

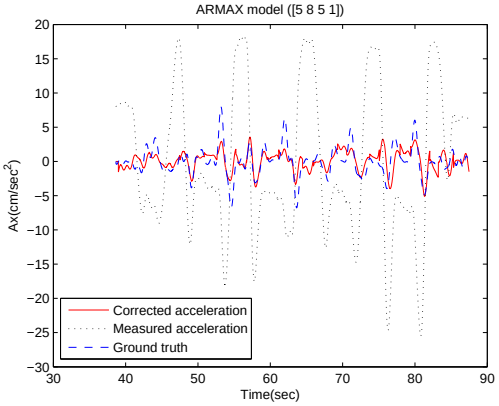


Figure 86: Estimated acceleration along x orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 5$, $nb = 8$, $nc = 5$ and $nk = 1$.

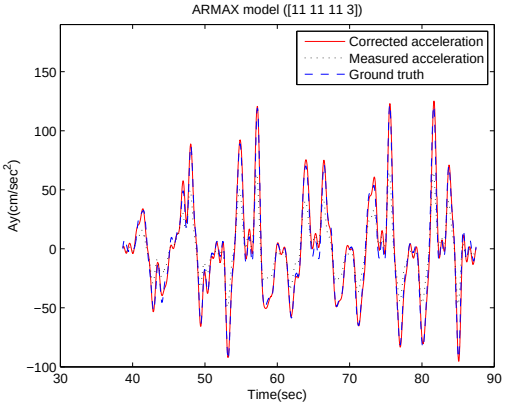


Figure 87: Estimated acceleration along y orientation. a_{meas} is input and $\frac{d}{dt}a_{true}$ is output. $na = 11$, $nb = 11$, $nc = 11$, and $nk = 3$.

6 Board Data Processing (July 2006)

The adaptive error mitigation and system identification algorithms are applied to the "Board Data" measured in July 2006. This data set, provided by AETC/SAIC, was collected with the ArcSecond system providing the ground truth. This is a departure from previous data sets where the ground truth was determined via video capture.

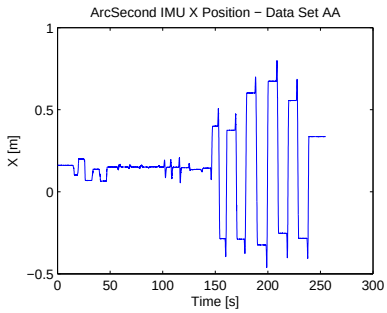
There are four individual data collections, labeled 'AA', 'BB', 'CC', and 'DD'. The ArcSecond ground truth for each data collection is shown in Figs. 88 though 91. With the exception of data collection 'AA', the IMU was generally moved in a sweeping motion consistent with handheld sensor operation. Data collection 'AA' consists of two segments. In the first segment, the IMU is subjected to angular rotations (pitch, roll, and yaw) while remaining approximately stationary in (x,y,z). In the second segment, the IMU is moved in a sweeping motion.

6.1 Adaptive Error Mitigation Algorithms

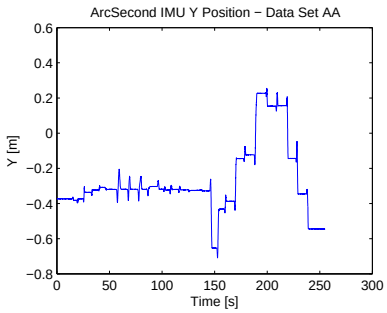
The adaptive error mitigation algorithms were applied to the measured IMU data. The raw accelerations after pre-processing (D/A conversion, bias removal, and trend removal) are shown in Fig. 92. The stabilized velocities and attitude angles (roll, pitch, yaw) follow in Figs. 93 and 94, respectively. The stabilized velocities are compared to the ground truth (in the Earth frame) in Figs. 95 though 98.

This data collection was not constrained as previous data sets were. Specifically, the sweeping motion was not constrained to follow an arc with a known radius about a fixed pivot point. Previous algorithms utilized knowledge of the constraints on the motion. Therefore, the stabilization algorithms had to be modified so they no longer relied on prior knowledge such as the radius of the sweeping arc. With this change, the stabilization algorithms are not different for each of the orientations. This change in the nature of the motion also removed the ability to use zero-position-points to improve the position estimates. It is still possible to use zero-velocity-points within the algorithms, however, under some mild assumptions. It is assumed that when the rates of change in all accelerations are very small for an extended length of time (on the order of several seconds), the IMU is stationary. The rate of change in the accelerations is utilized rather than the accelerations because it was observed that the accelerations were not always zero when the IMU was stationary, but they were consistently a steady value. It is possible for this assumption to be violated if the IMU is truly experiencing constant acceleration. Given the likely dynamics of the problem, however, it is anticipated that this assumption will not be violated in typical situations.

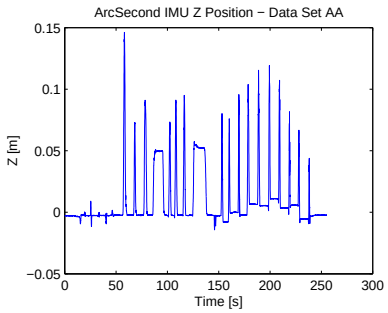
The estimated positions (in Earth frame), obtained by integrating the stabilized velocities, are presented in Figs. 99 through 102. The estimated positions for this data collected are not as accurate as for previous data collections. (There is a scale/magnitude difference that could not be reconciled). The estimates show some qualitative resemblance to the ground truth measured by the ArcSecond system, but performance falls significantly below the desired positioning accuracy.



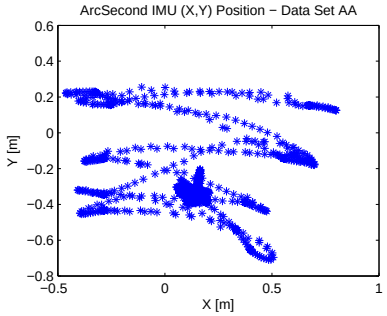
(a) X Position.



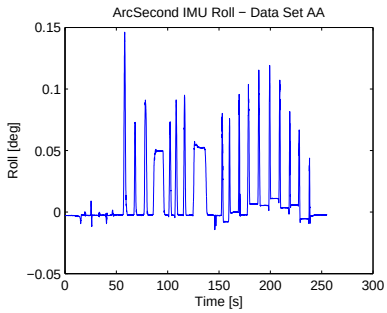
(b) Y Position.



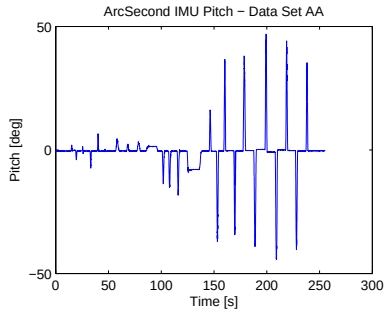
(c) Z Position.



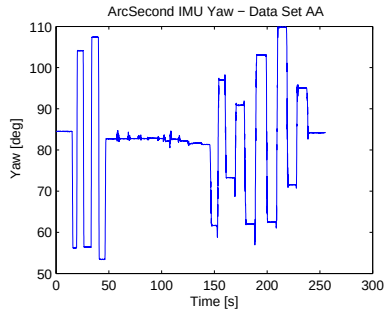
(d) (X,Y) Position.



(e) Roll.

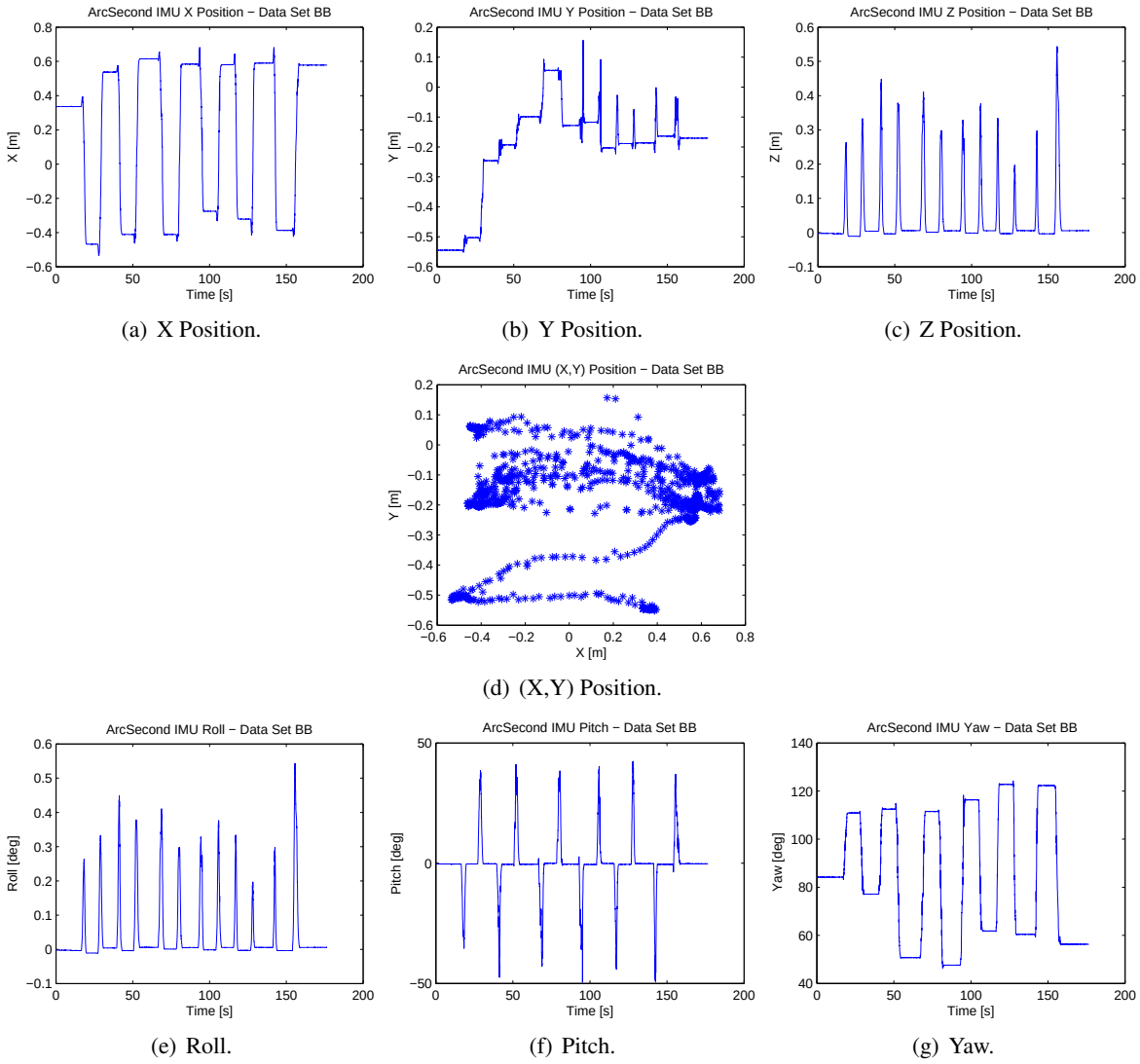


(f) Pitch.



(g) Yaw.

Figure 88: ArcSecond ground truth for data set 'AA'.

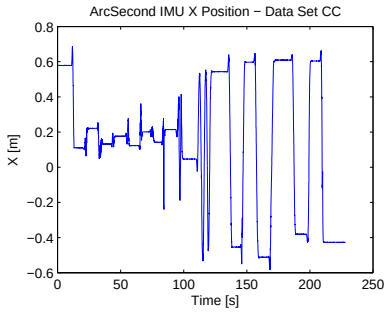


(a) X Position. (b) Y Position. (c) Z Position.

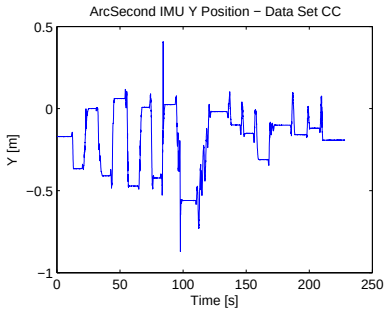
(d) (X,Y) Position.

(e) Roll. (f) Pitch. (g) Yaw.

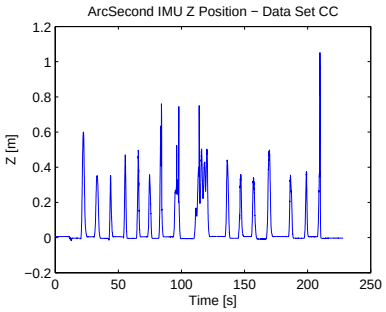
Figure 89: ArcSecond ground truth for data set 'BB'.



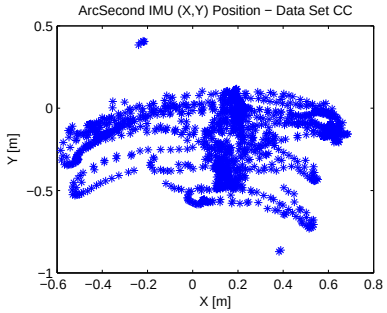
(a) X Position.



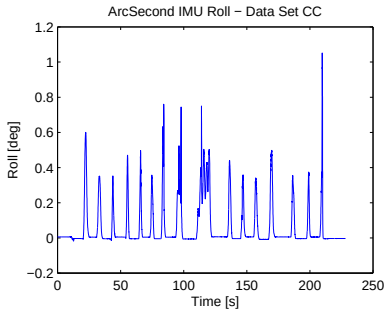
(b) Y Position.



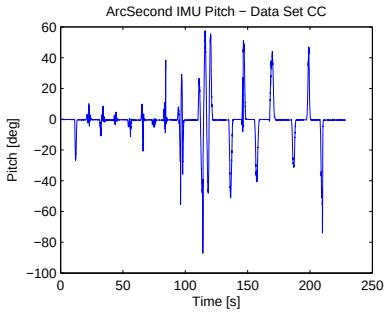
(c) Z Position.



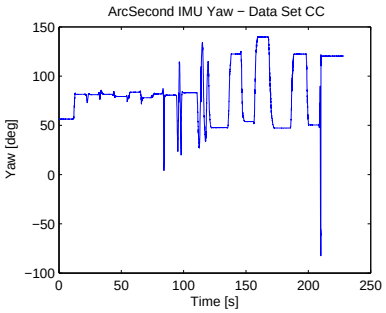
(d) (X,Y) Position.



(e) Roll.

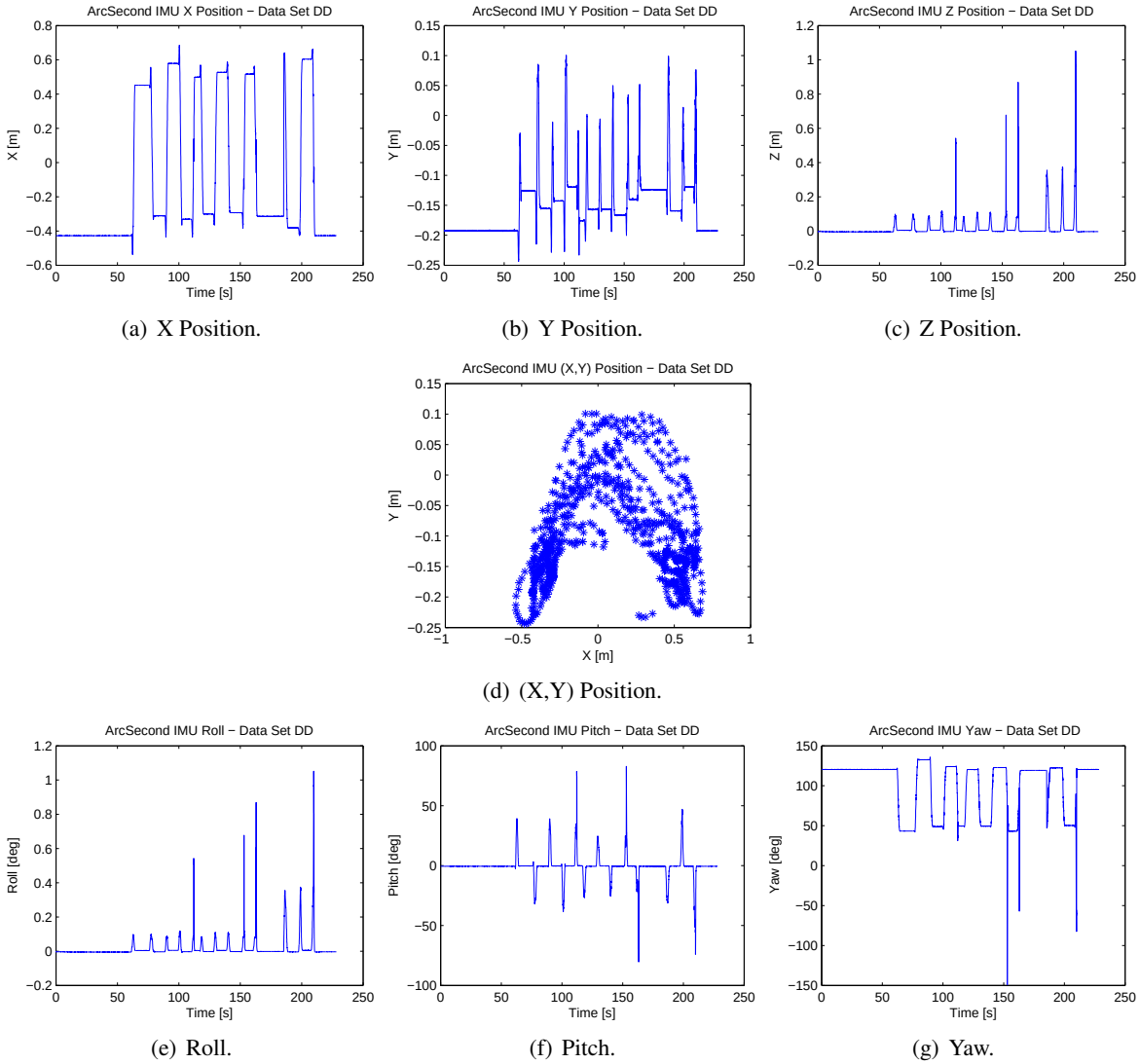


(f) Pitch.



(g) Yaw.

Figure 90: ArcSecond ground truth for data set 'CC'.

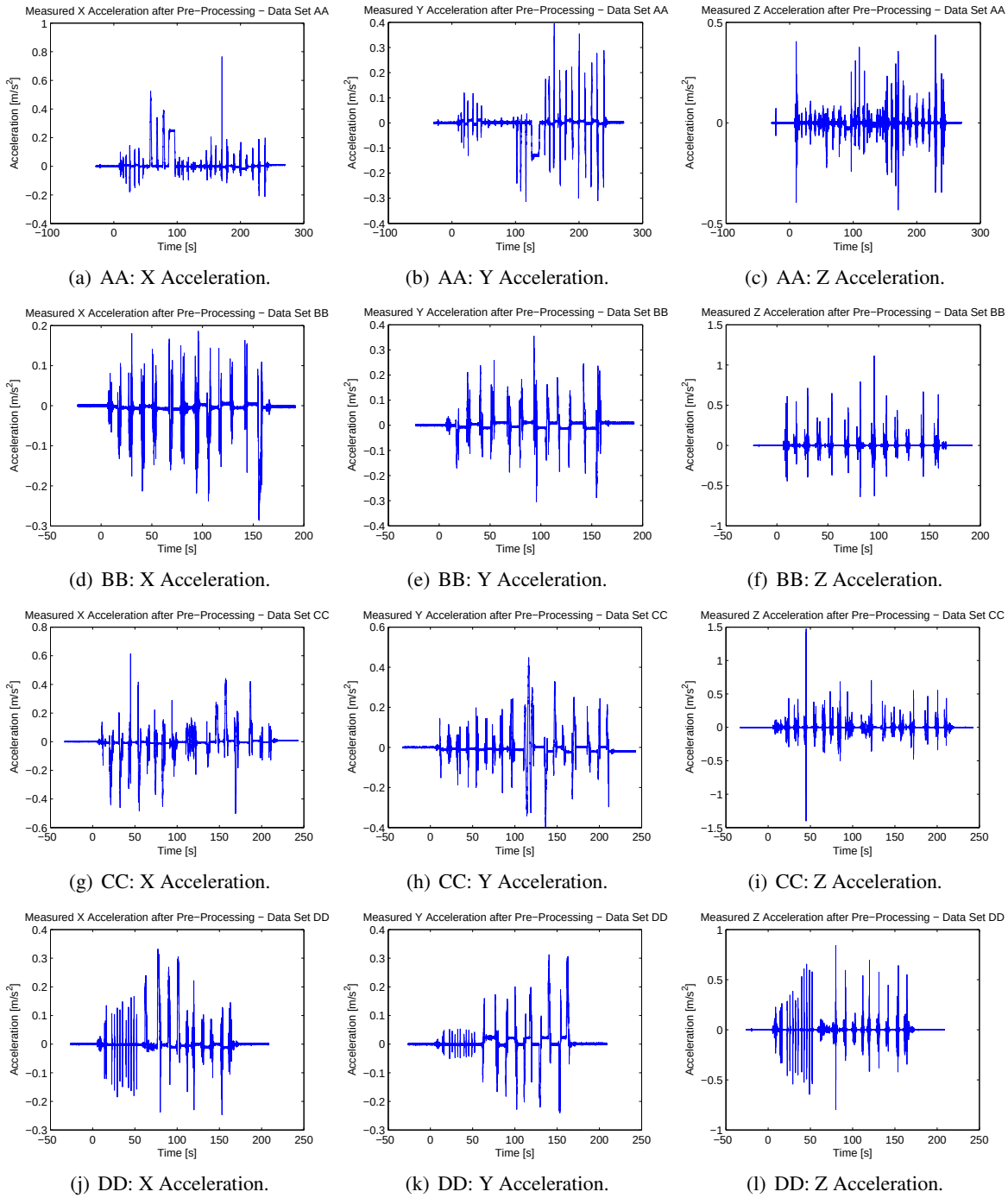


(a) X Position. (b) Y Position. (c) Z Position.

(d) (X,Y) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 91: ArcSecond ground truth for data set 'DD'.



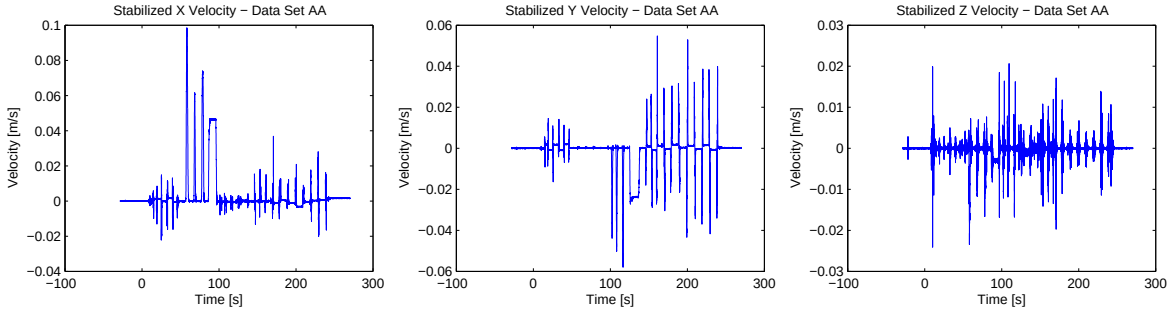
(a) AA: X Acceleration. (b) AA: Y Acceleration. (c) AA: Z Acceleration.

(d) BB: X Acceleration. (e) BB: Y Acceleration. (f) BB: Z Acceleration.

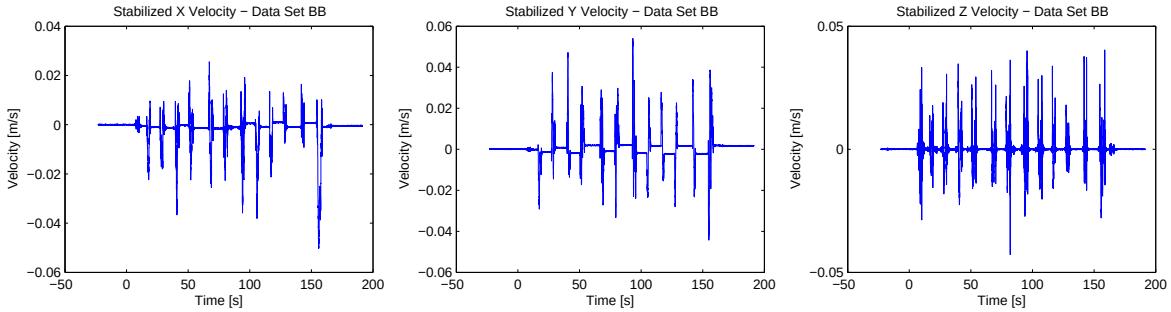
(g) CC: X Acceleration. (h) CC: Y Acceleration. (i) CC: Z Acceleration.

(j) DD: X Acceleration. (k) DD: Y Acceleration. (l) DD: Z Acceleration.

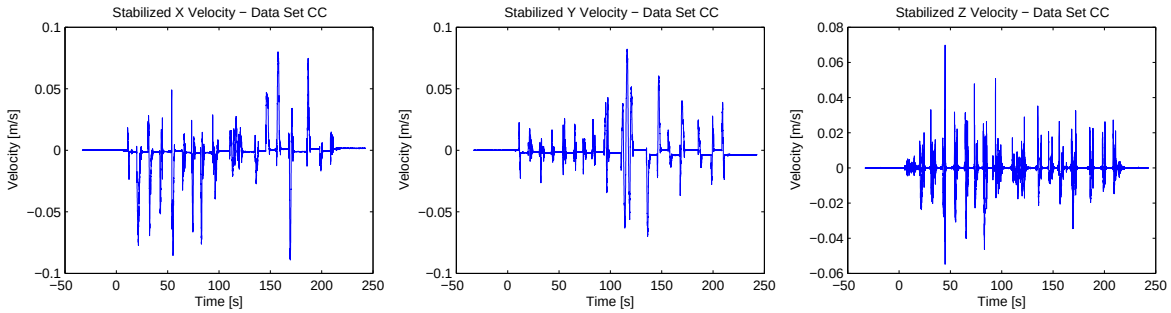
Figure 92: Measured accelerations after pre-processing.



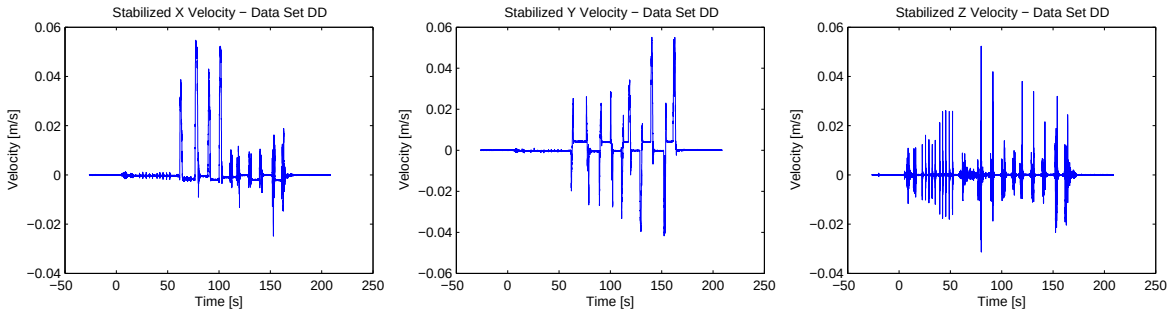
(a) AA: Stabilized X Velocity. (b) AA: Stabilized Y Velocity. (c) AA: Stabilized Z Velocity.



(d) BB: Stabilized X Velocity. (e) BB: Stabilized Y Velocity. (f) BB: Stabilized Z Velocity.

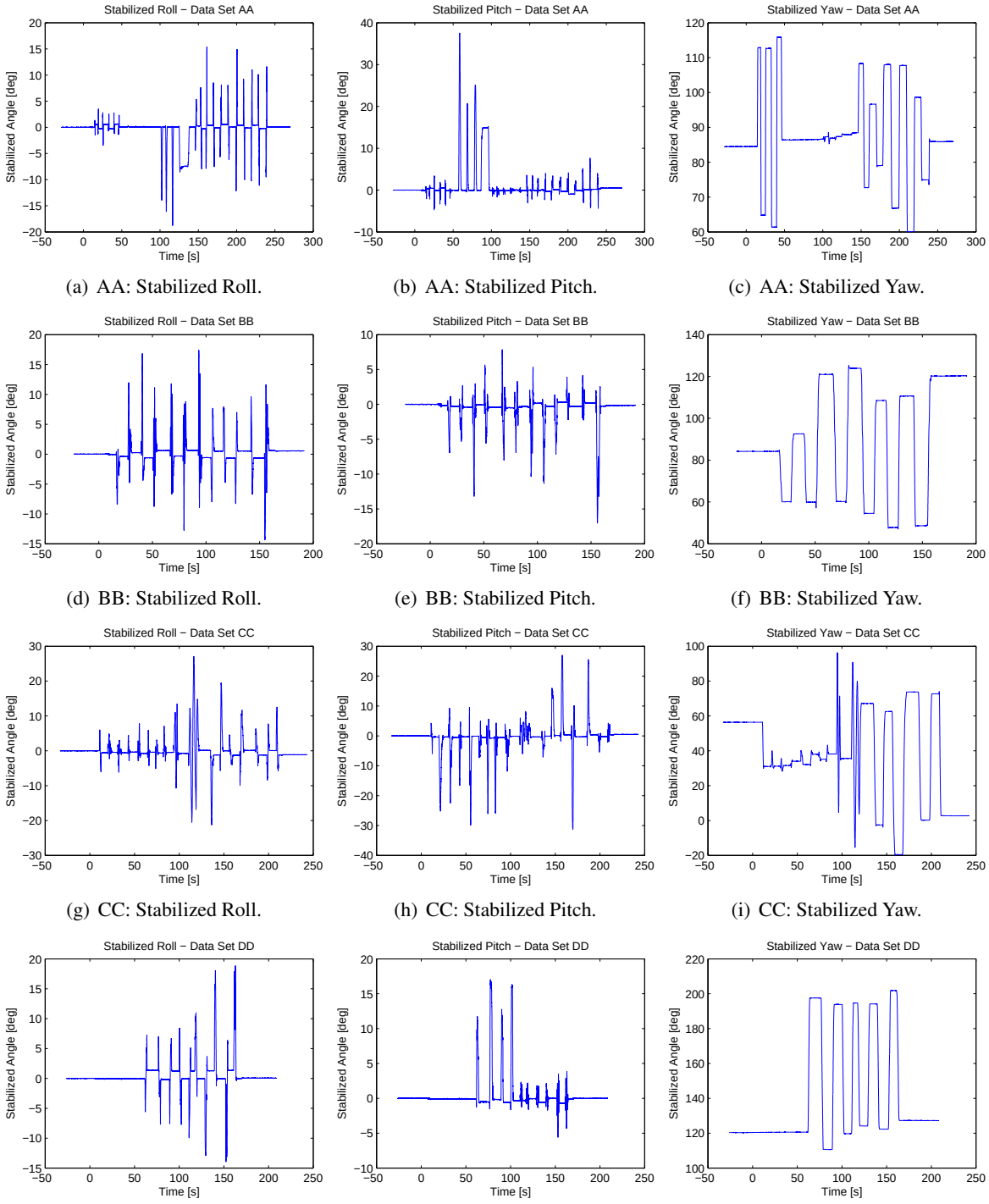


(g) CC: Stabilized X Velocity. (h) CC: Stabilized Y Velocity. (i) CC: Stabilized Z Velocity.



(j) DD: Stabilized X Velocity. (k) DD: Stabilized Y Velocity. (l) DD: Stabilized Z Velocity.

Figure 93: Stabilized velocities after bias and trend removal.



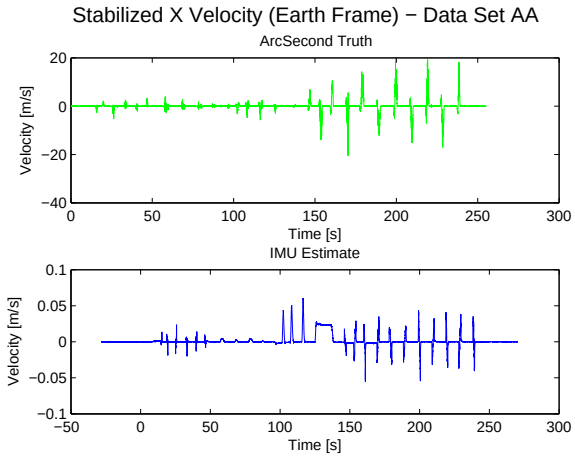
(a) AA: Stabilized Roll. (b) AA: Stabilized Pitch. (c) AA: Stabilized Yaw.

(d) BB: Stabilized Roll. (e) BB: Stabilized Pitch. (f) BB: Stabilized Yaw.

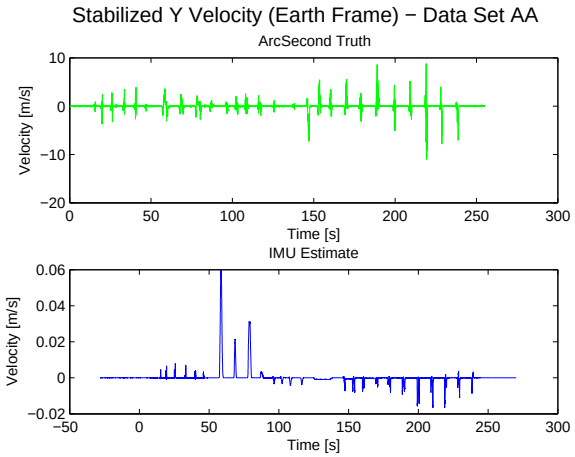
(g) CC: Stabilized Roll. (h) CC: Stabilized Pitch. (i) CC: Stabilized Yaw.

(j) DD: Stabilized Roll. (k) DD: Stabilized Pitch. (l) DD: Stabilized Yaw.

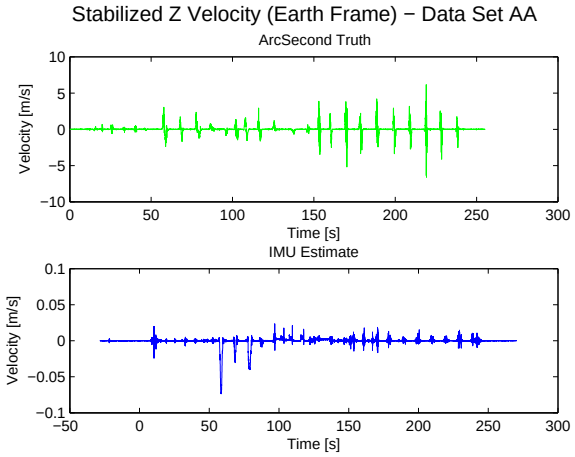
Figure 94: Stabilized attitude angles (roll, pitch, yaw) after bias and trend removal.



(a) Stabilized X velocity.

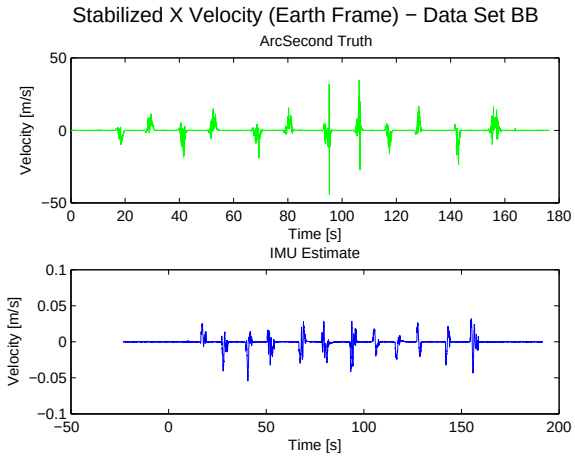


(b) Stabilized Y velocity.

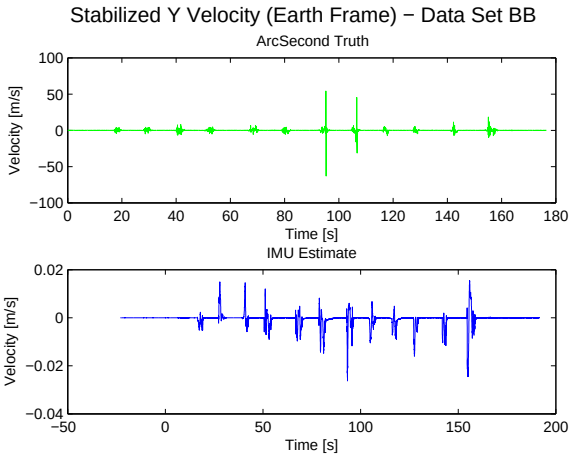


(c) Stabilized Z velocity.

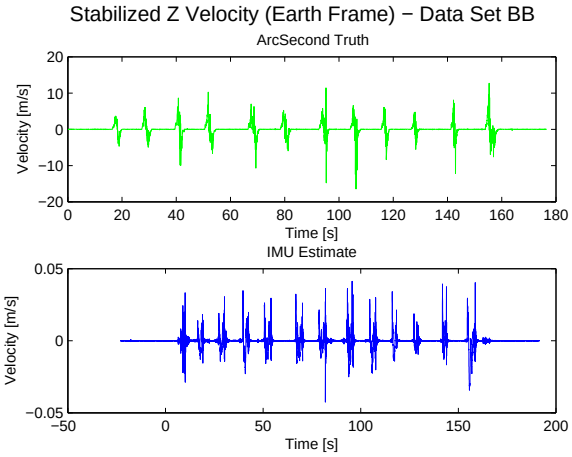
Figure 95: Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'AA'.



(a) Stabilized X velocity.

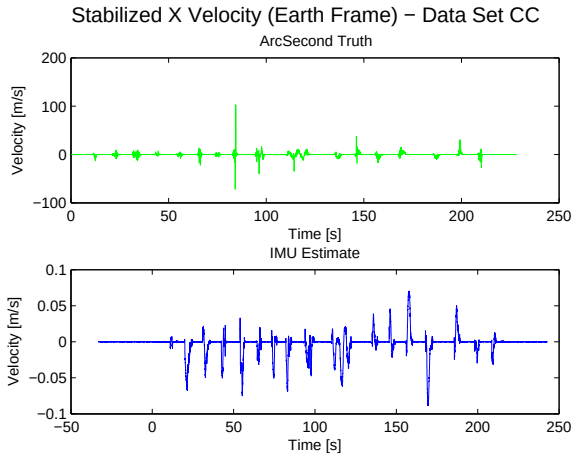


(b) Stabilized Y velocity.

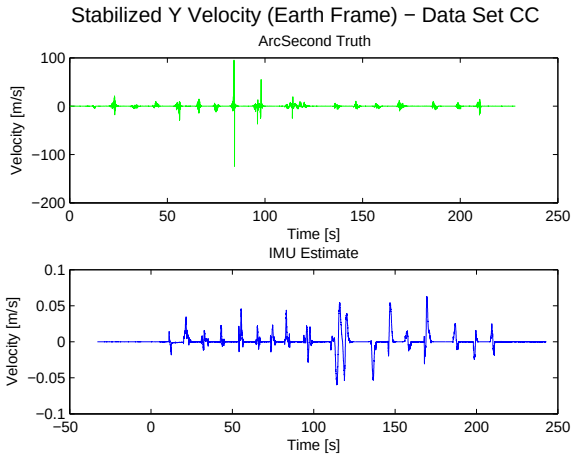


(c) Stabilized Z velocity.

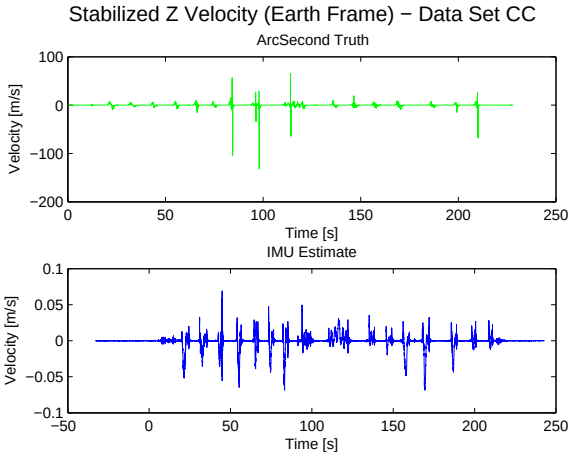
Figure 96: Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'BB'.



(a) Stabilized X velocity.

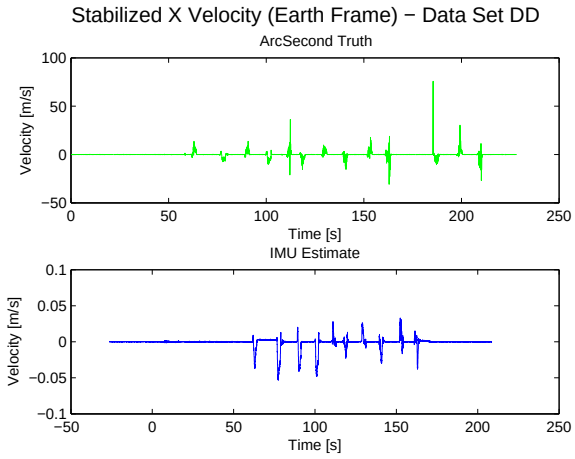


(b) Stabilized Y velocity.

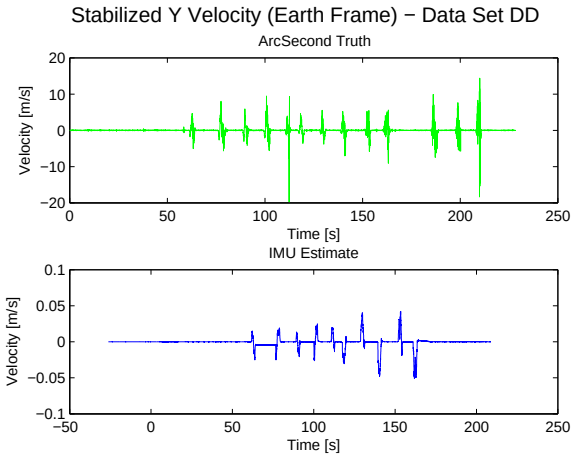


(c) Stabilized Z velocity.

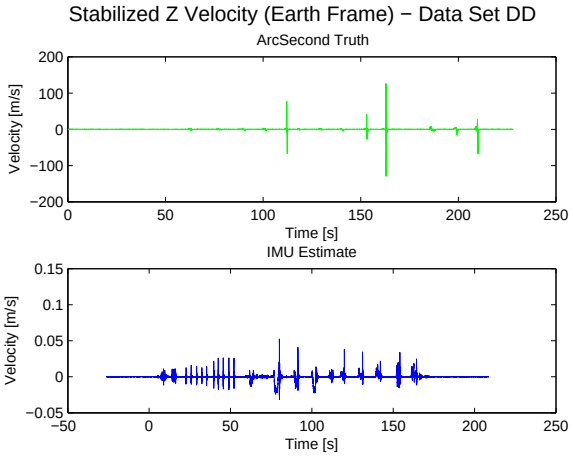
Figure 97: Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'CC'.



(a) Stabilized X velocity.

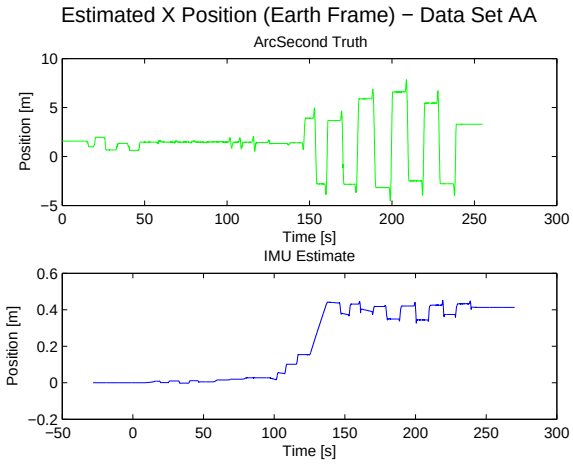


(b) Stabilized Y velocity.

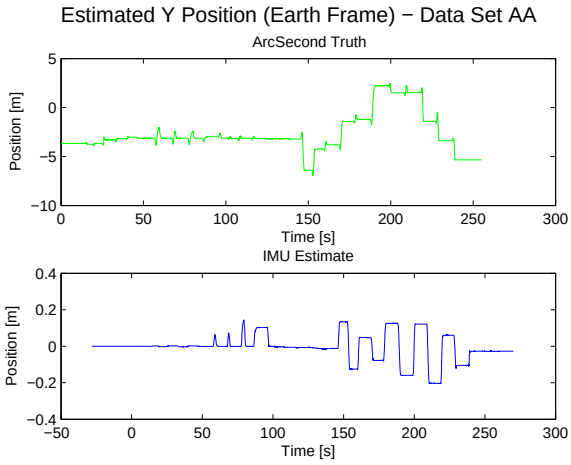


(c) Stabilized Z velocity.

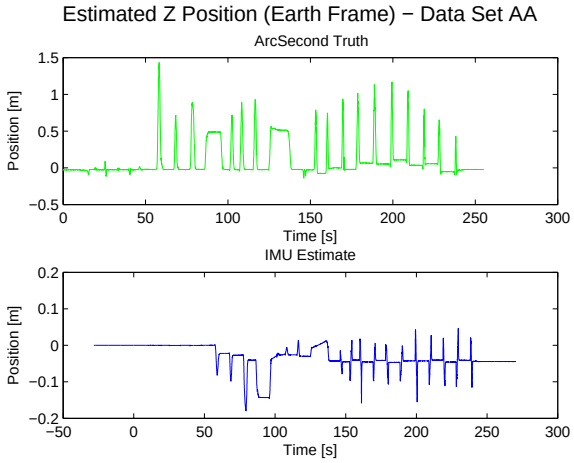
Figure 98: Stabilized velocities (in Earth frame) and comparison to ground truth for data set 'DD'.



(a) Estimated X position.

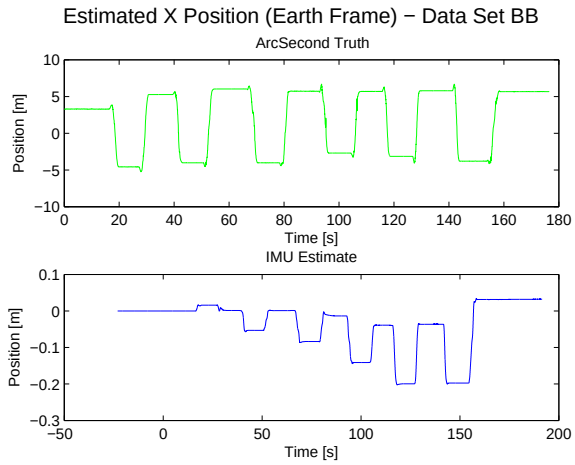


(b) Estimated Y position.

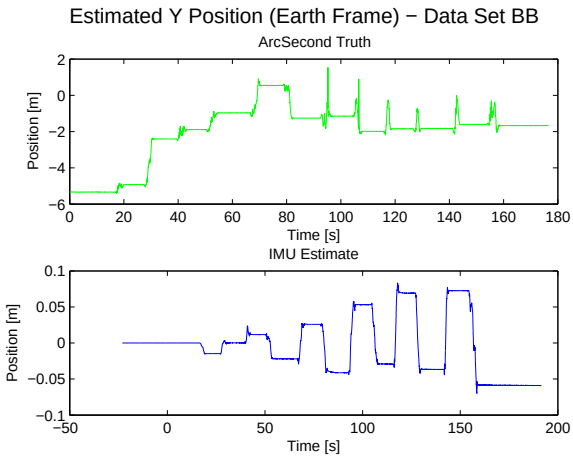


(c) Estimated Z position.

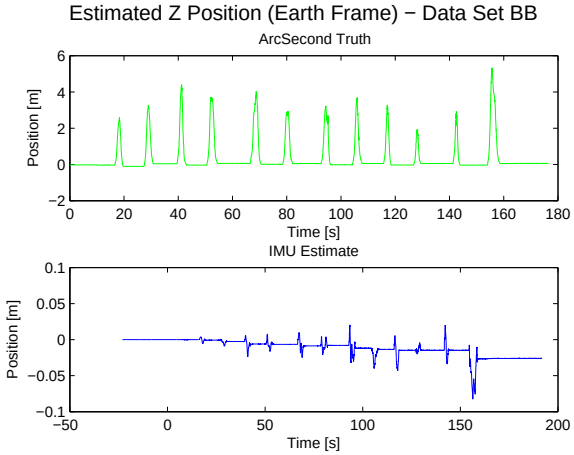
Figure 99: Estimated positions (in Earth frame) and comparison to ground truth for data set 'AA'.



(a) Estimated X position.

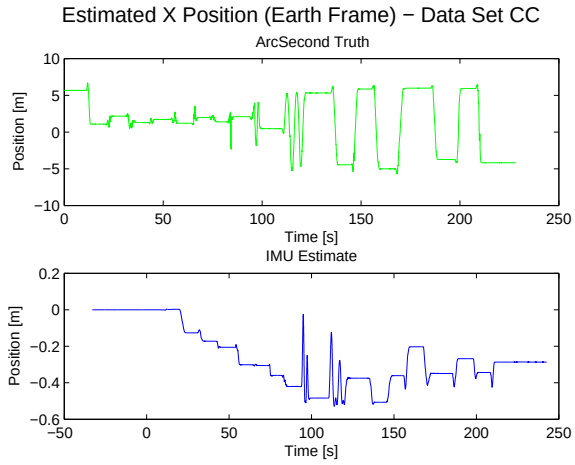


(b) Estimated Y position.

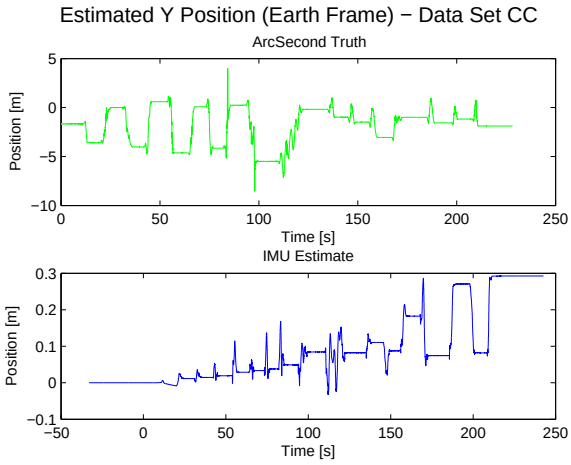


(c) Estimated Z position.

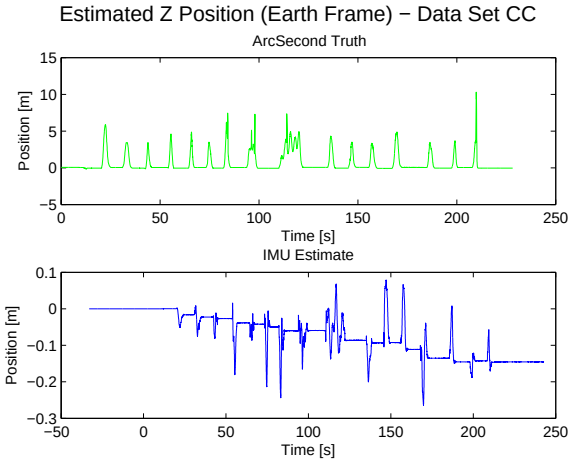
Figure 100: Estimated positions (in Earth frame) and comparison to ground truth for data set 'BB'.



(a) Estimated X position.

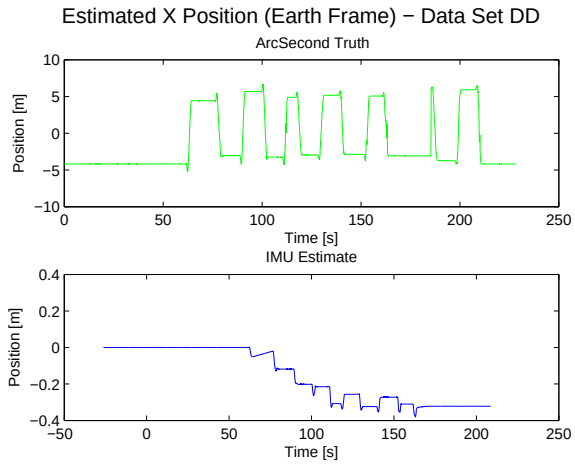


(b) Estimated Y position.

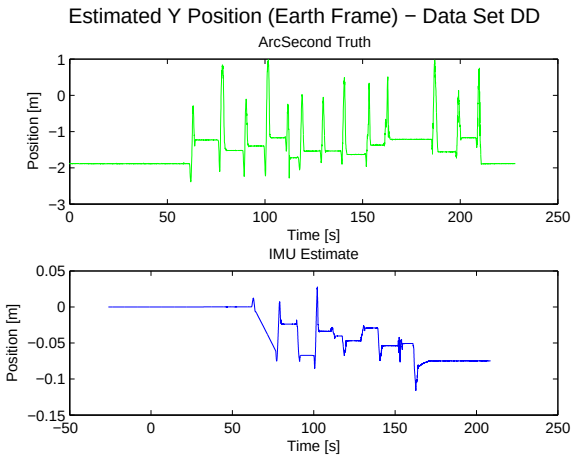


(c) Estimated Z position.

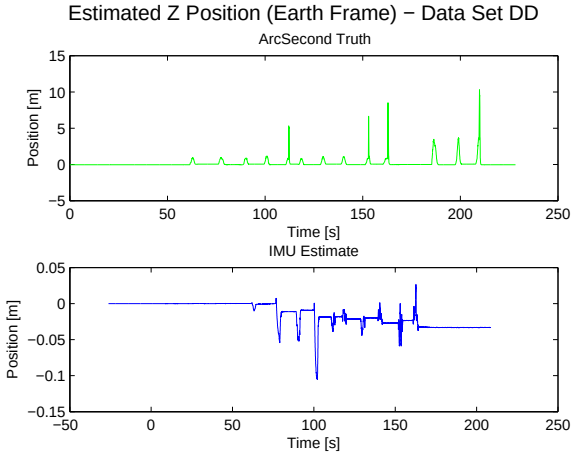
Figure 101: Estimated positions (in Earth frame) and comparison to ground truth for data set 'CC'.



(a) Estimated X position.



(b) Estimated Y position.



(c) Estimated Z position.

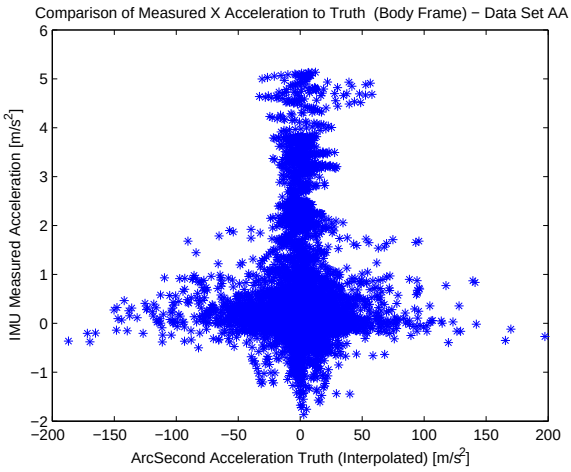
Figure 102: Estimated positions (in Earth frame) and comparison to ground truth for data set 'DD'.

6.2 System Identification Algorithms

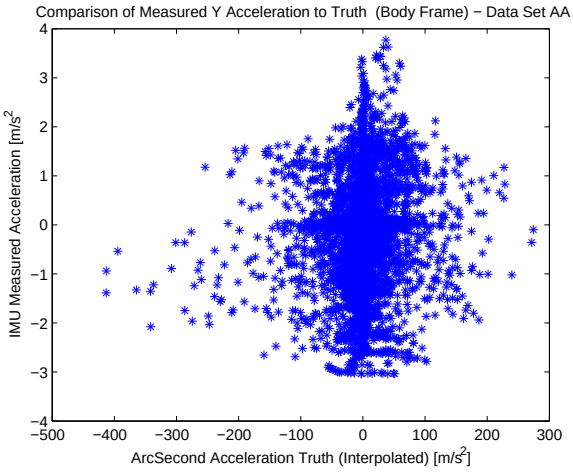
The system identification algorithms were applied to the measured IMU data. The parameter estimation routines, however, failed to converge and produce viable estimates of the underlying system that modifies the true accelerations to produce the accelerations reported by the IMU. The true accelerations versus the IMU measured accelerations are shown in Figs.103 through 106. These figures illustrate that there is no clear relationship between the true and measured accelerations.

6.3 Summary

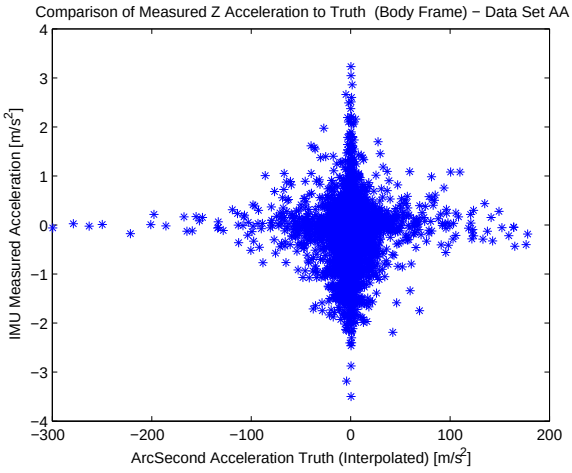
Adaptive error mitigation and system identification algorithms were applied to the board data collected in July 2006. Prior to applying the adaptive error mitigation algorithms, they were modified to eliminate the reliance on prior knowledge of the system's motion. Specifically, the assumptions of either linear motion or motion in a perfect arc of known radius about a fixed pivot point were removed. Processing this data produced results which do not meet the desired sensor positioning performance. It is believed that not having access to prior knowledge regarding the sensor's motion adversely impacted algorithm performance.



(a) AA: X Acceleration.

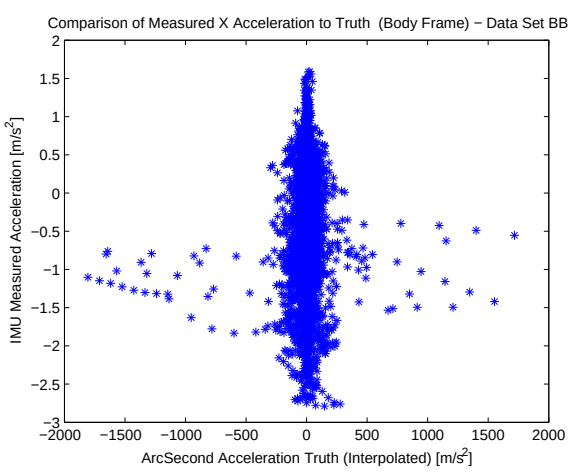


(b) AA: Y Acceleration.

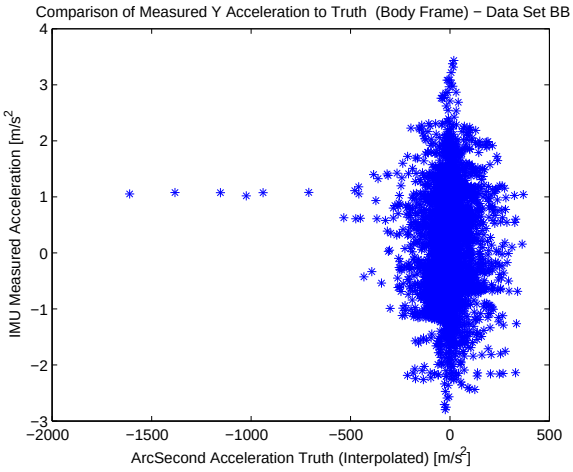


(c) AA: Z Acceleration.

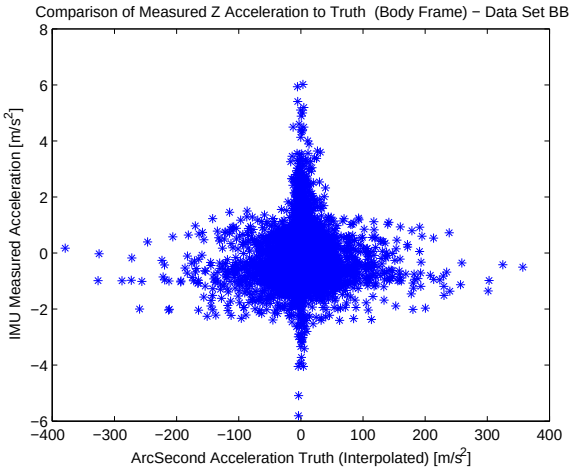
Figure 103: Comparisons of true and measured accelerations (in IMU body frame) for data set 'AA'.



(a) BB: X Acceleration.

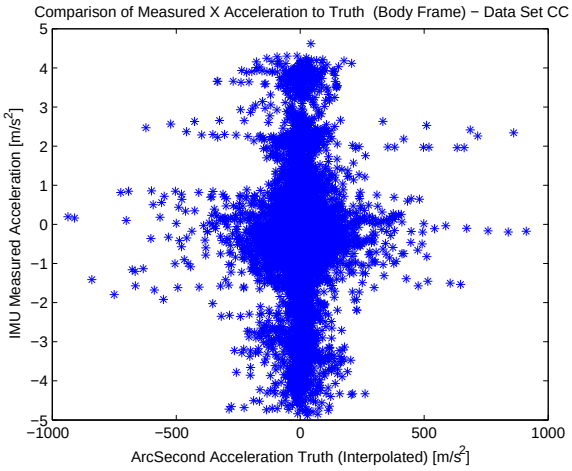


(b) BB: Y Acceleration.

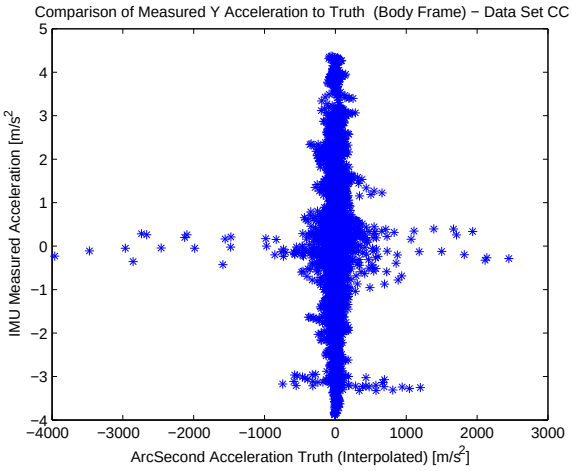


(c) BB: Z Acceleration.

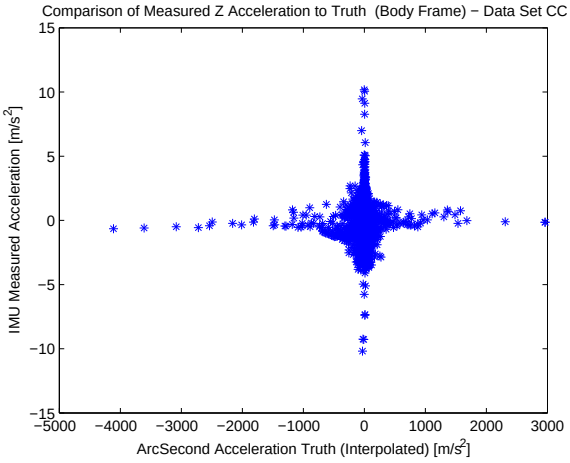
Figure 104: Comparisons of true and measured accelerations (in IMU body frame) for data set 'BB'.



(a) CC: X Acceleration.

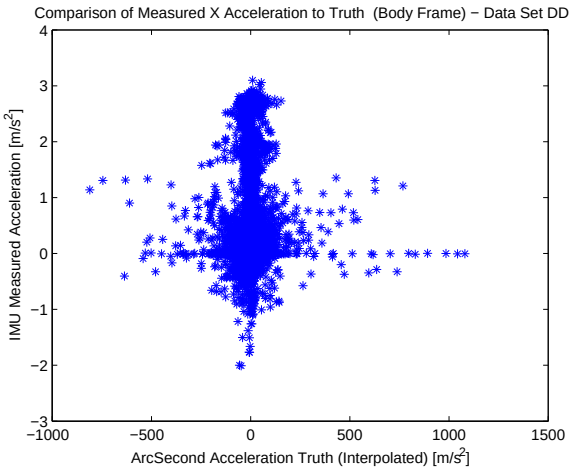


(b) CC: Y Acceleration.

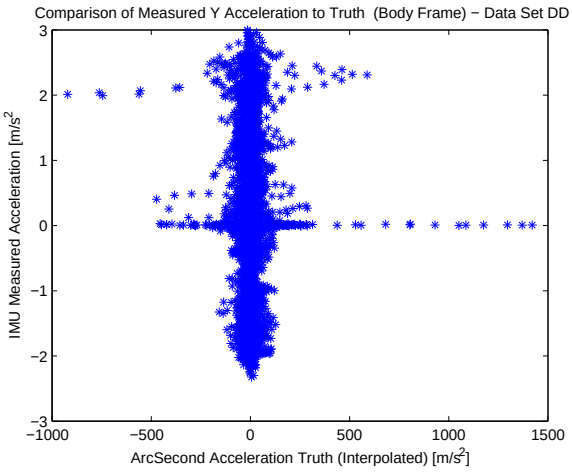


(c) CC: Z Acceleration.

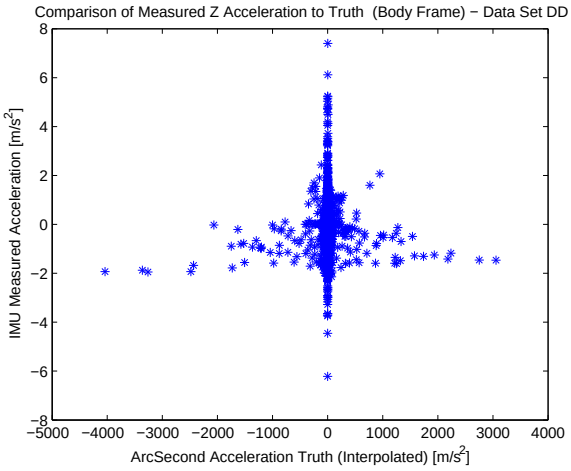
Figure 105: Comparisons of true and measured accelerations (in IMU body frame) for data set 'CC'.



(a) DD: X Acceleration.



(b) DD: Y Acceleration.



(c) DD: Z Acceleration.

Figure 106: Comparisons of true and measured accelerations (in IMU body frame) for data set 'DD'.

7 APG SAINT Data Processing (September 2006)

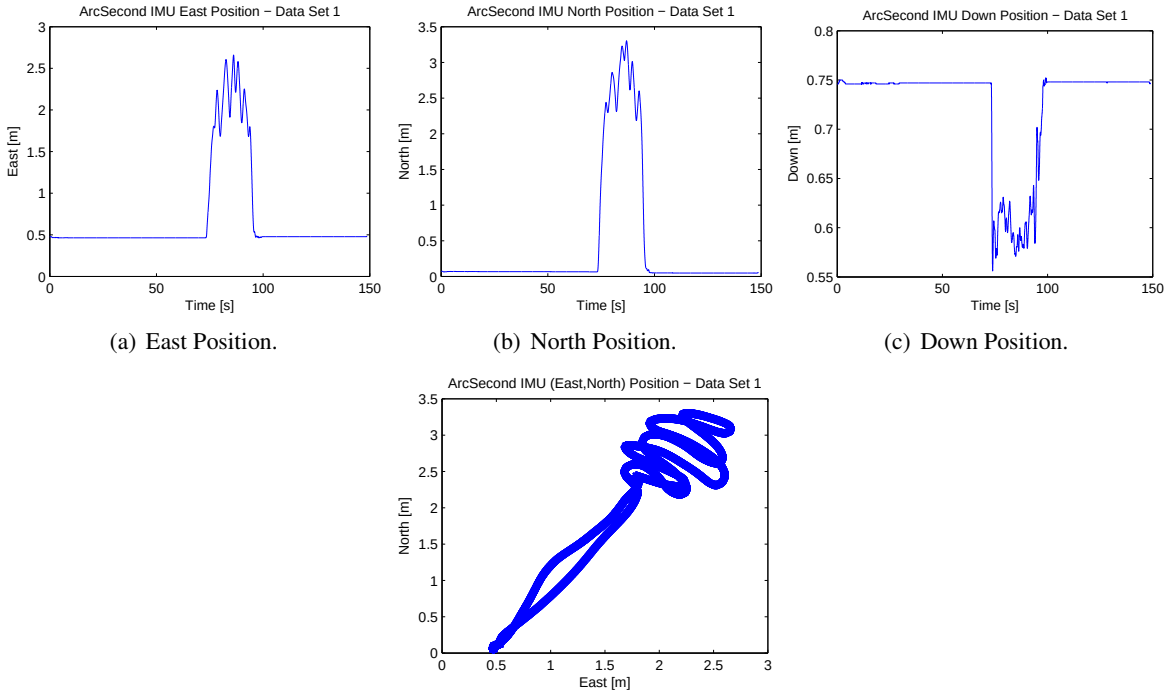
The adaptive error mitigation and system identification algorithms are applied to the "APG SAINT Data" measured in September 2006. This data set, provided by AETC/SAIC, was collected with a third vendor's (ENSCO's) IMU and with the ArcSecond system providing the ground truth. ENSCO's processed IMU data agrees very well with the ArcSecond ground truth. The goal of processing this data set was to test the algorithms developed thus far by comparing their results to the ENSCO's processed IMU data.

There were two data collections, morning and afternoon, and each data collection consists of 5 data sets. The ArcSecond ground truth for each data collection is shown in Figs. 107 through 116. Each data set consists of the sensor starting in a fixed location, being swept about as if measuring data above a target of interest, and then returning to the fixed starting location.

7.1 Adaptive Error Mitigation Algorithms

The adaptive error mitigation algorithms were applied to the measured IMU data. The raw accelerations after pre-processing (D/A conversion, bias removal, and trend removal) are shown in Figs. 117 and 118. This data collection produced measurements for which a universal pre-processing approach did not apply. In previous data collections, a set of pre-processing steps could be applied to all the measured data. For this data collection, that is not the case; the measurements did not consistently improve with the bias and trend removal. In some instances, pre-processing degraded the measurement quality, as is illustrated in some of the figures showing the pre-processed IMU measurements. While it is possible to choose pre-processing parameters and processes individually for each data set so that each set of measurements has neither a bias nor a trend, it was decided to show the results for a single set of pre-processing parameters so that the sensitivity to the pre-processing could be shown. The stabilized velocities and attitude angles (roll, pitch, yaw) follow in Figs. 119 through 122. Obviously, inadequacies in the acceleration pre-processing lead to difficulties in finding the stabilized velocities, and this is shown in the results. The stabilized velocities in the Earth frame (East, North) are shown in Figs. 123 though 132.

The difficulties encountered with this data set parallel those encountered with the previous (Board Data) data set. Specifically, the sensor motion was not constrained, and algorithm performance degraded without the benefit of prior knowledge. The estimated positions (in Earth frame), obtained by integrating the stabilized velocities, are presented in Figs. 133 through 142. The estimated positions for this data collected are not as accurate as for previous data collections. (There is a scale/magnitude difference that could not be reconciled). The estimates show little qualitative resemblance to the ground truth measured by the ArcSecond system, and performance falls significantly below the desired positioning accuracy.

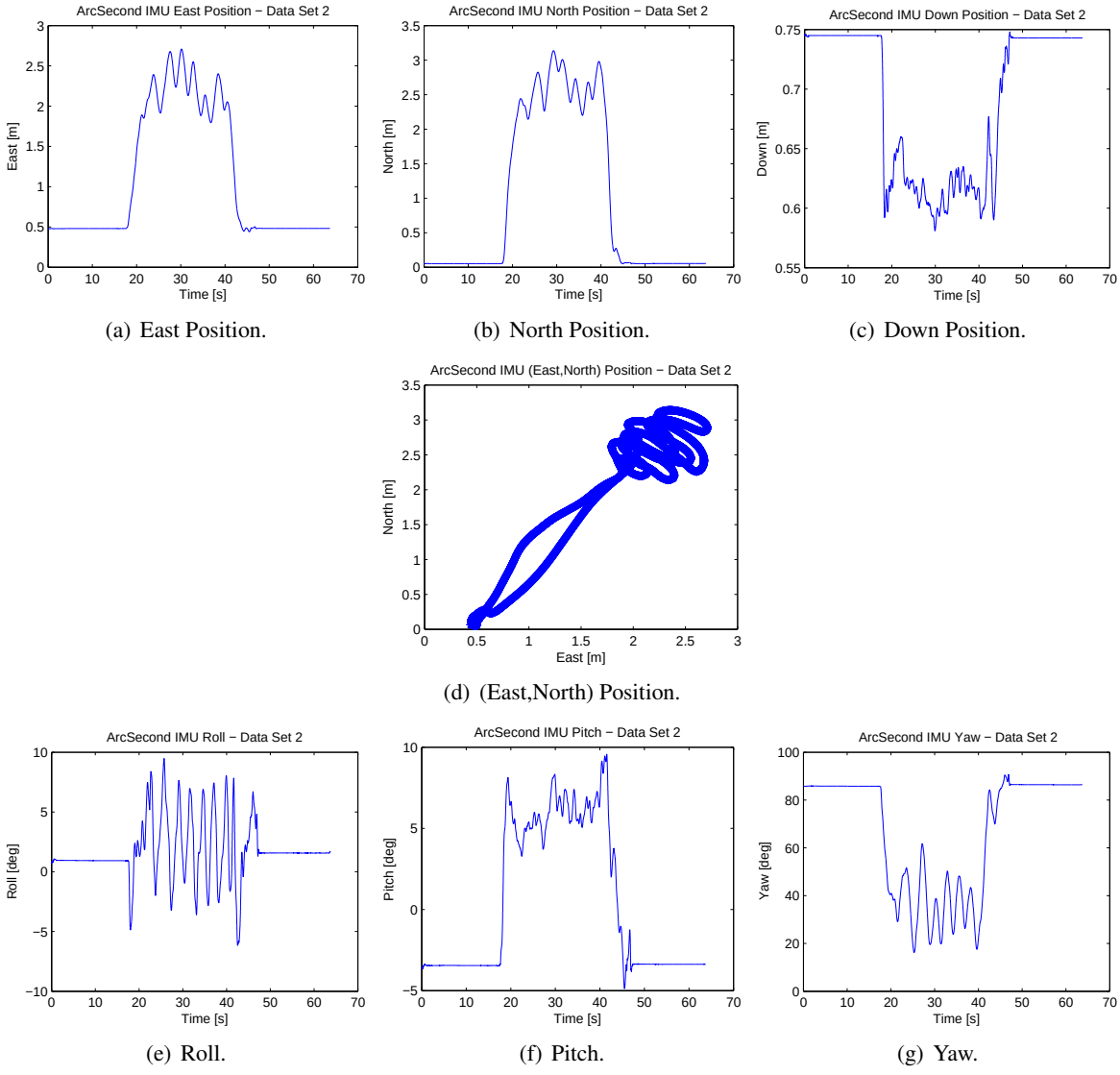


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 107: ArcSecond ground truth for data set 1 from the morning collection.

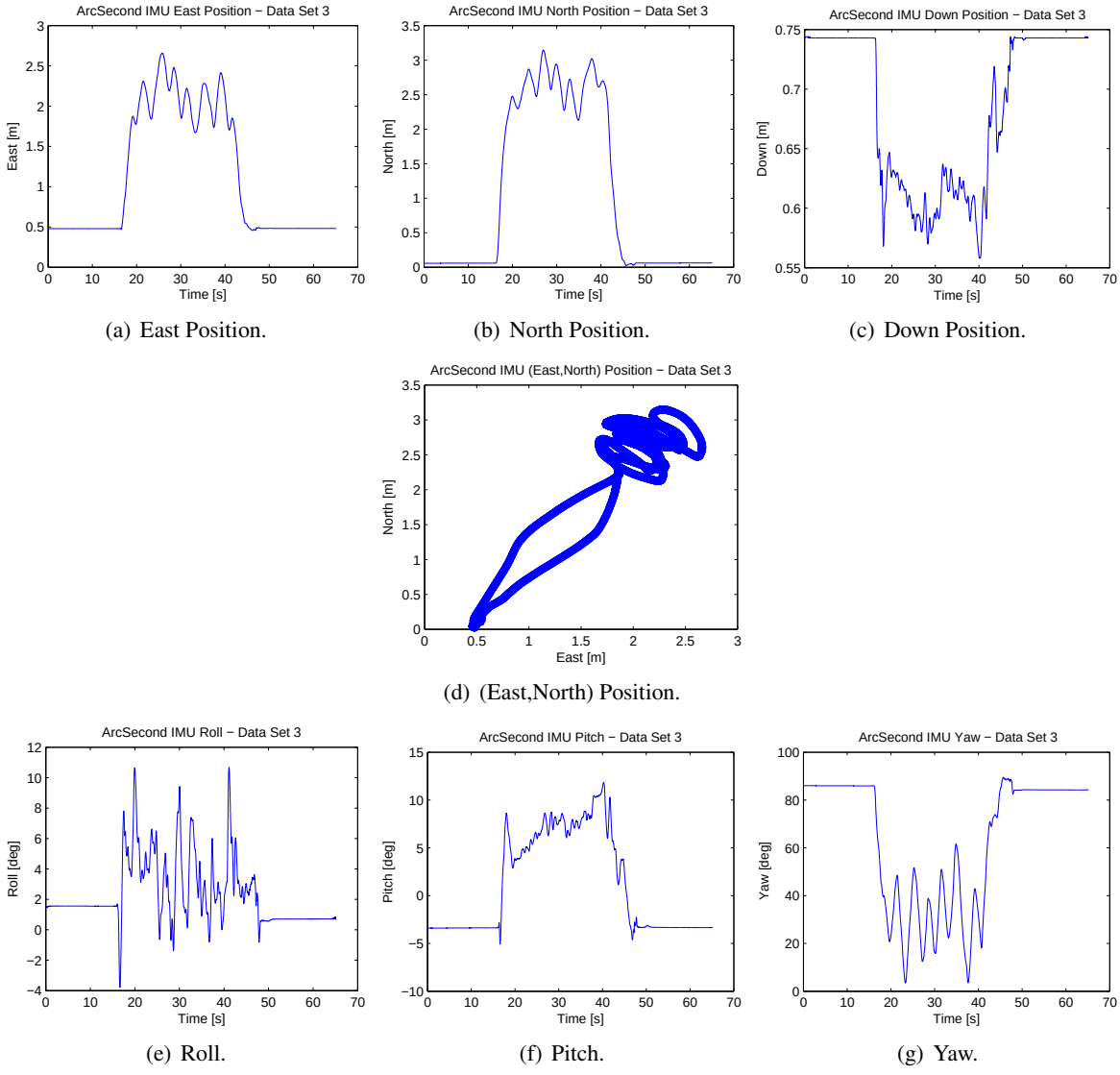


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 108: ArcSecond ground truth for data set 2 from the morning collection.

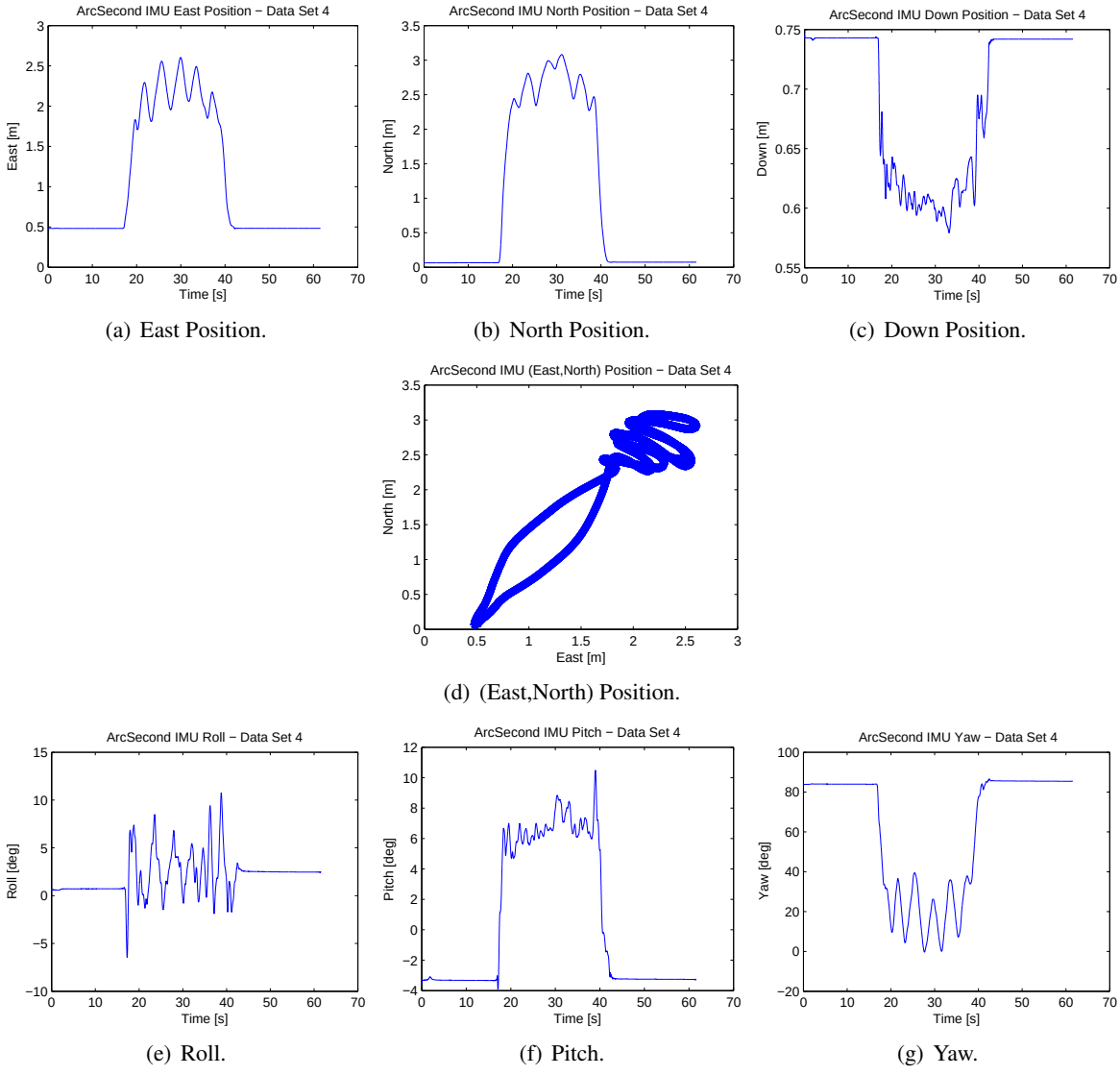


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 109: ArcSecond ground truth for data set 3 from the morning collection.

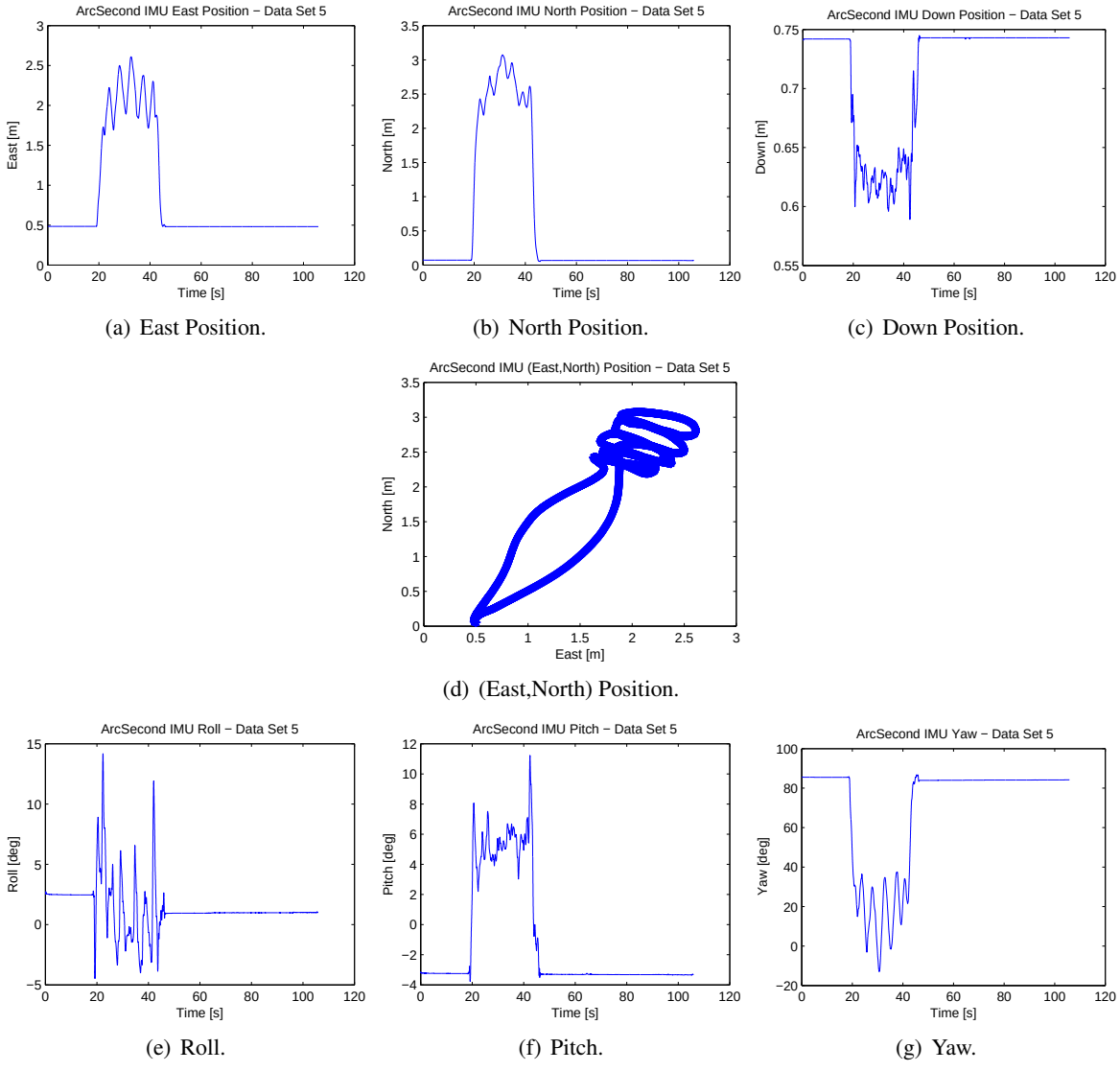


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 110: ArcSecond ground truth for data set 4 from the morning collection.

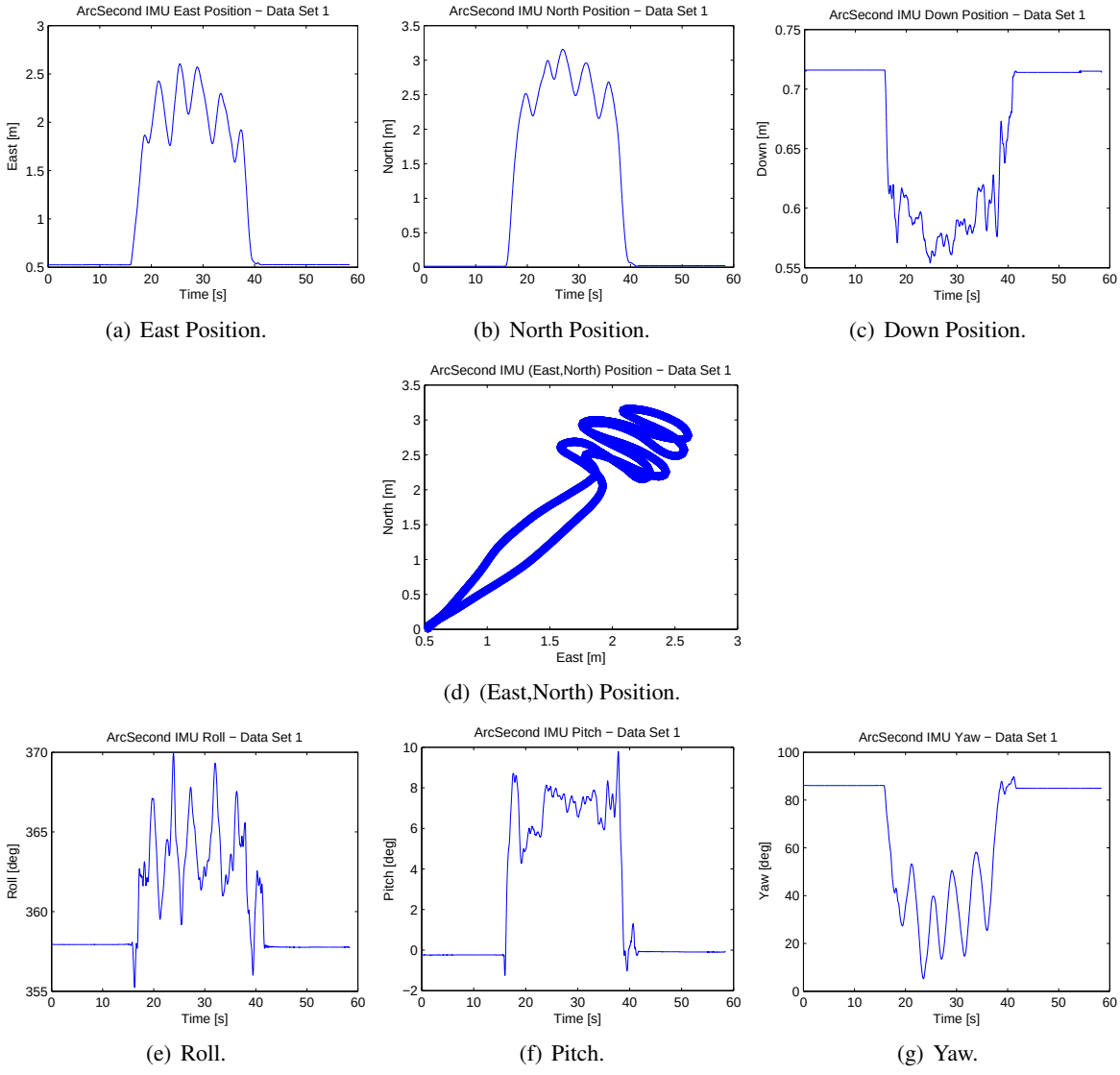


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 111: ArcSecond ground truth for data set 5 from the morning collection.

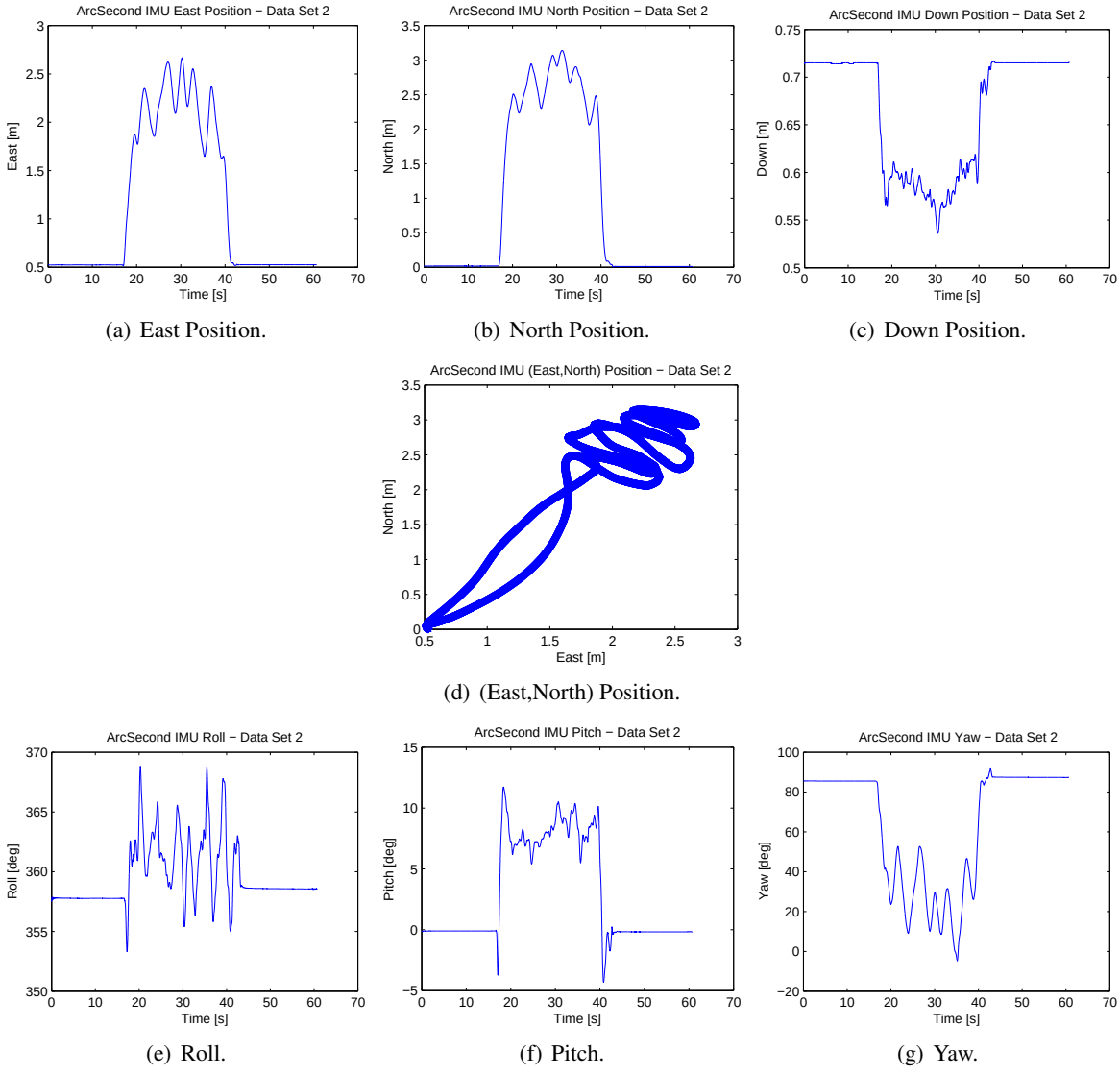


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 112: ArcSecond ground truth for data set 1 from the afternoon collection.

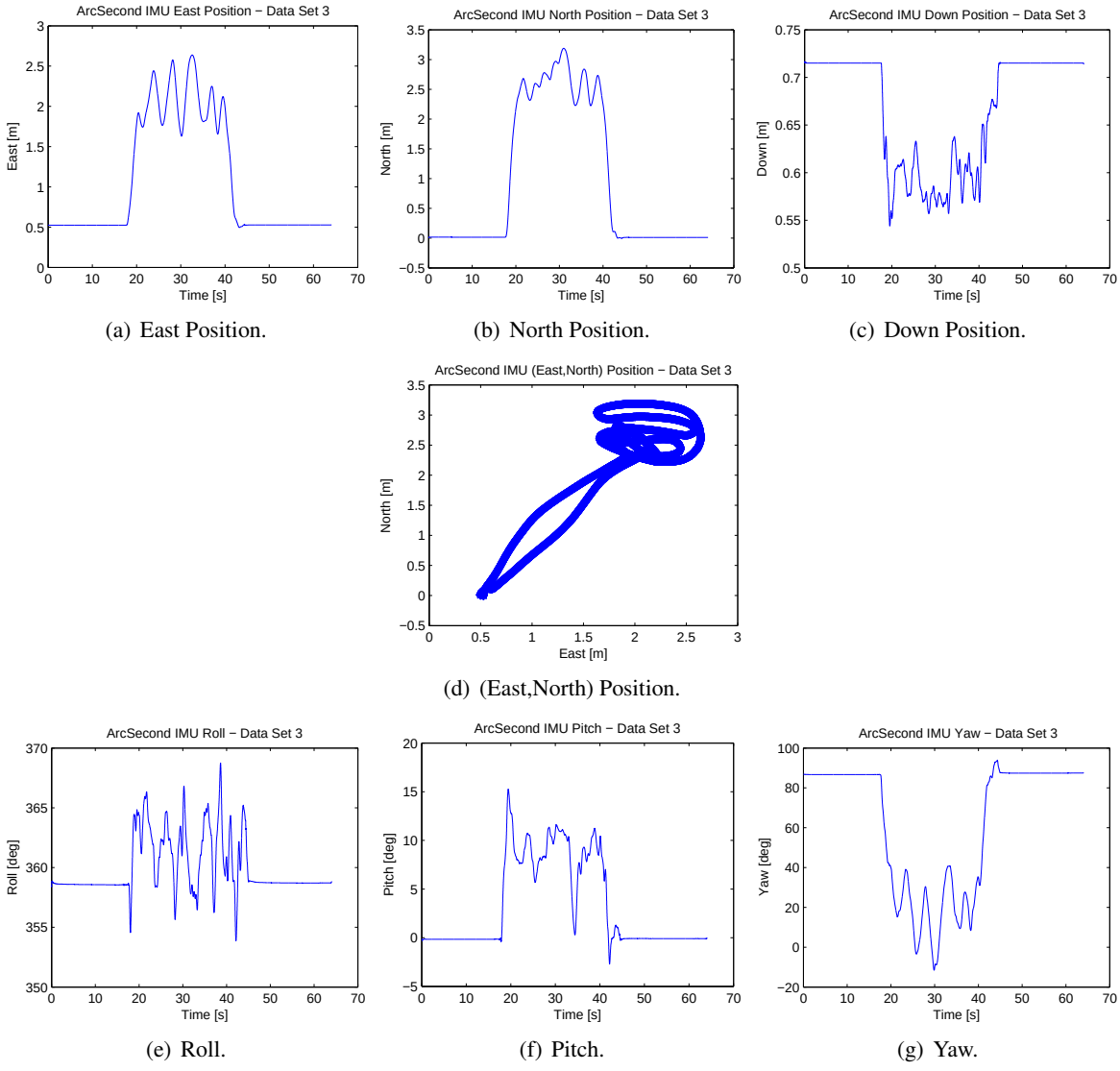


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 113: ArcSecond ground truth for data set 2 from the afternoon collection.

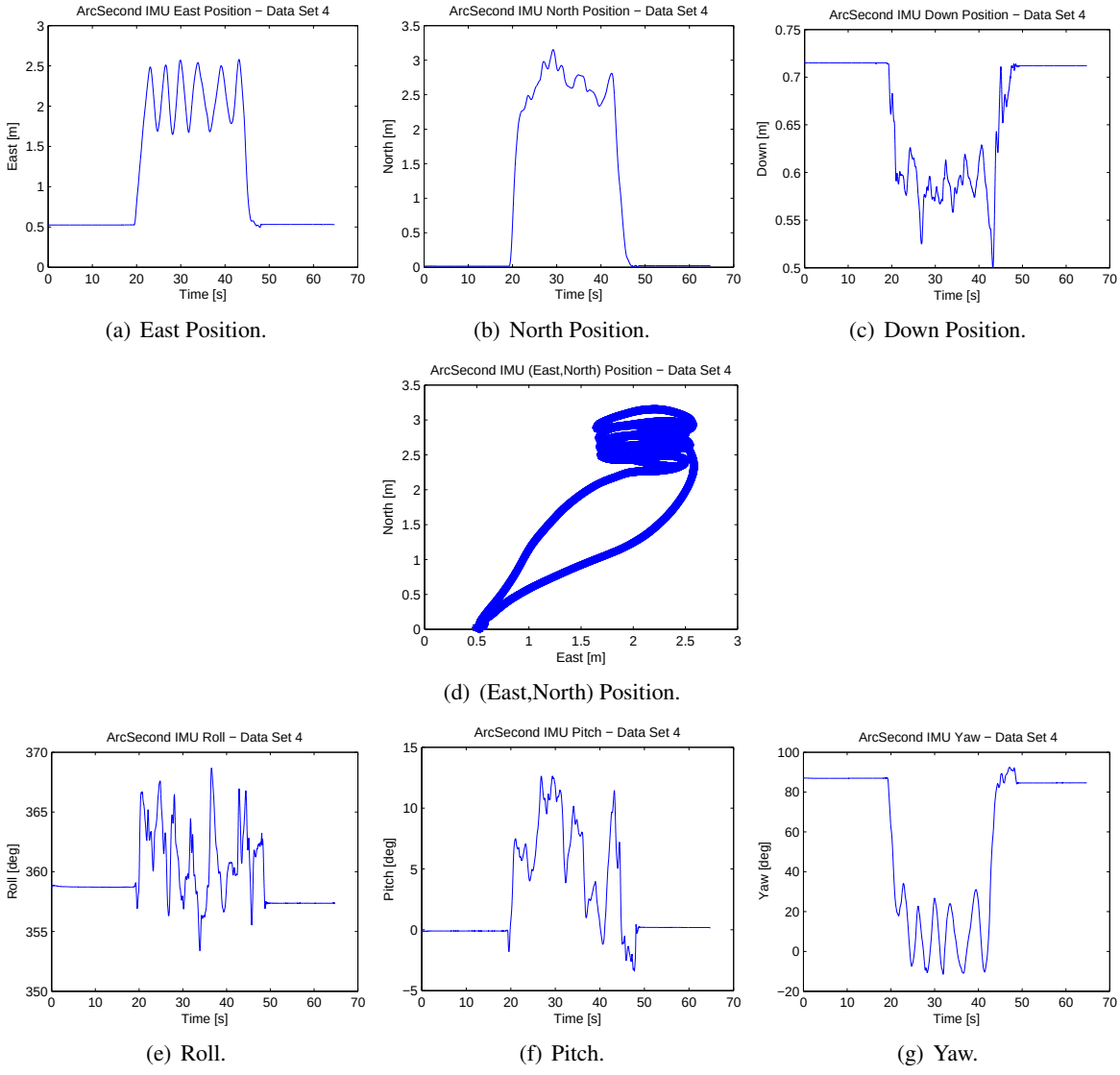


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 114: ArcSecond ground truth for data set 3 from the afternoon collection.

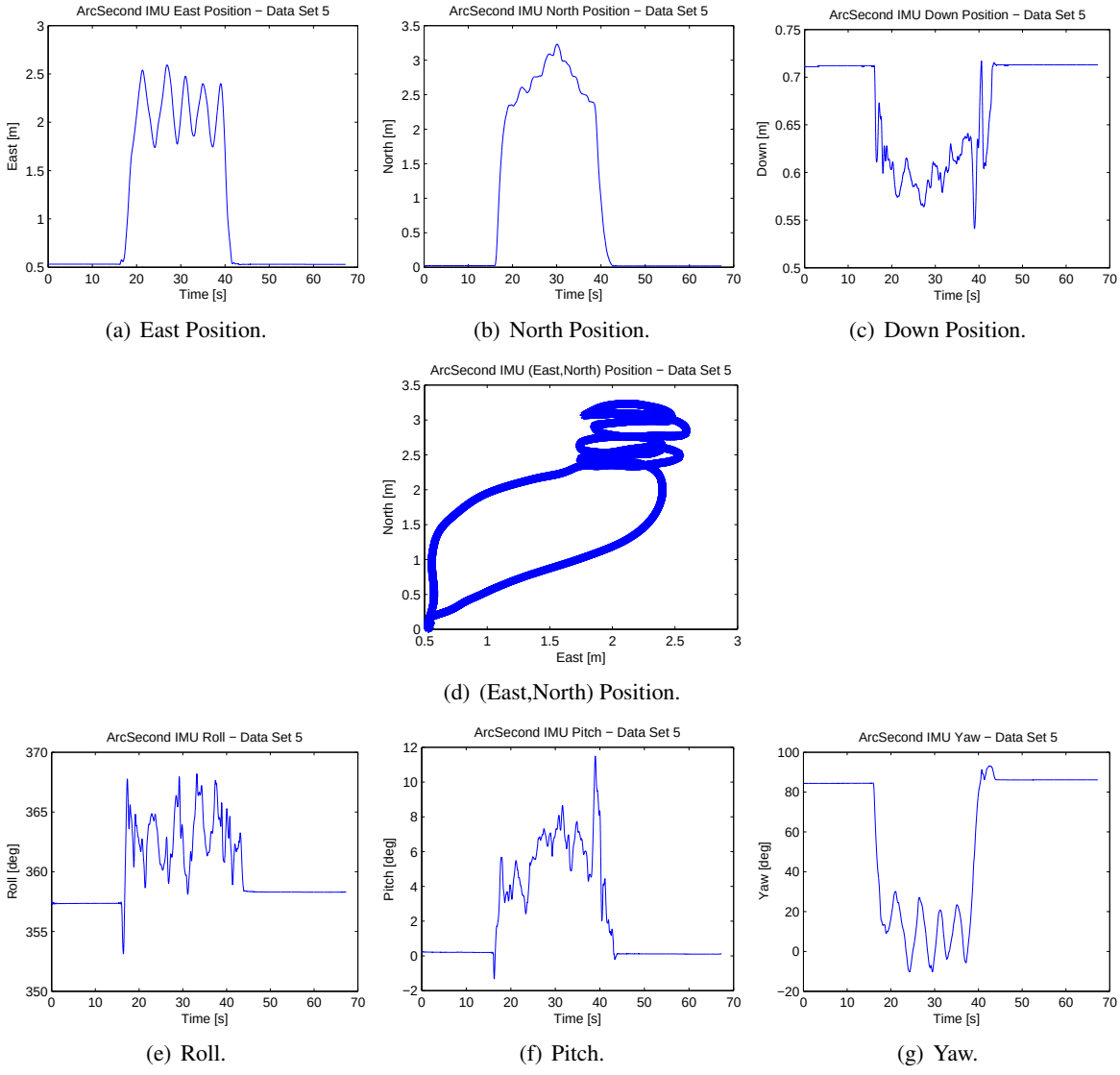


(a) East Position. (b) North Position. (c) Down Position.

(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

Figure 115: ArcSecond ground truth for data set 4 from the afternoon collection.

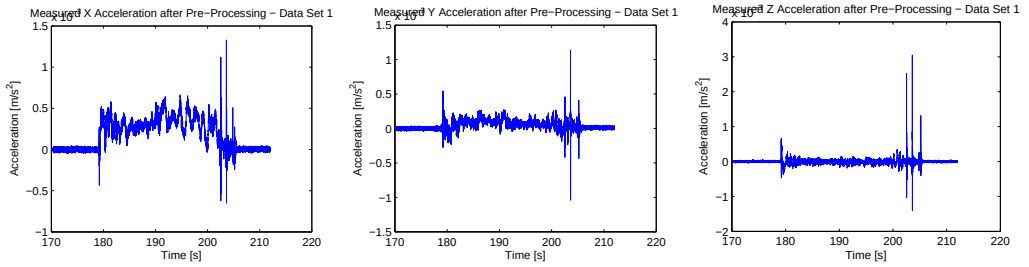


(a) East Position. (b) North Position. (c) Down Position.

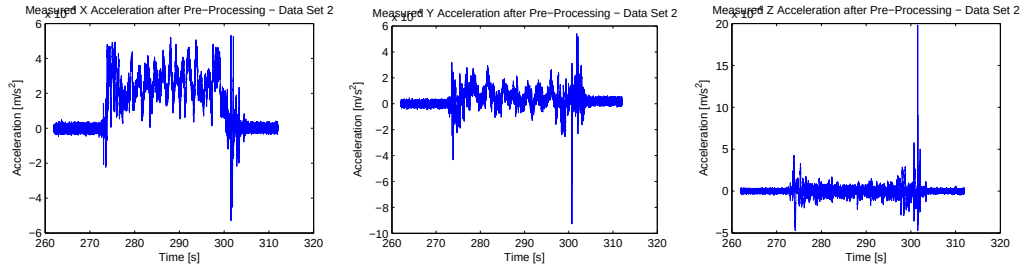
(d) (East,North) Position.

(e) Roll. (f) Pitch. (g) Yaw.

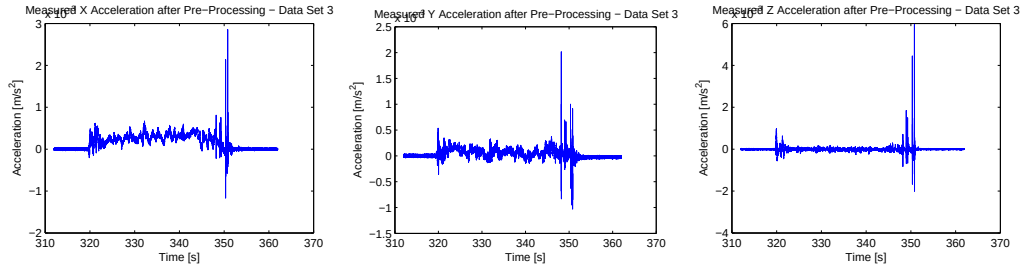
Figure 116: ArcSecond ground truth for data set 5 from the afternoon collection.



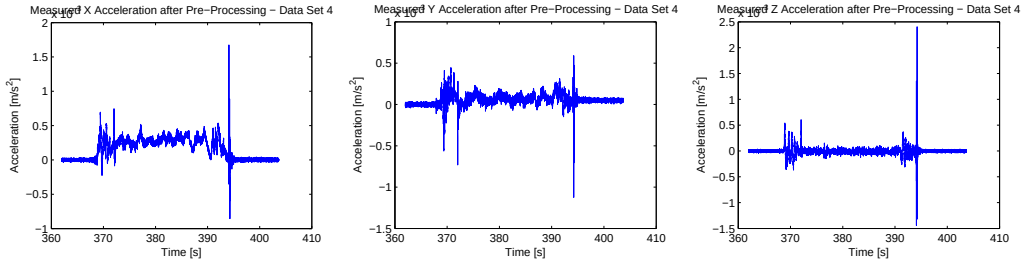
(a) Set 1: X Acceleration. (b) Set 1: Y Acceleration. (c) Set 1: Z Acceleration.



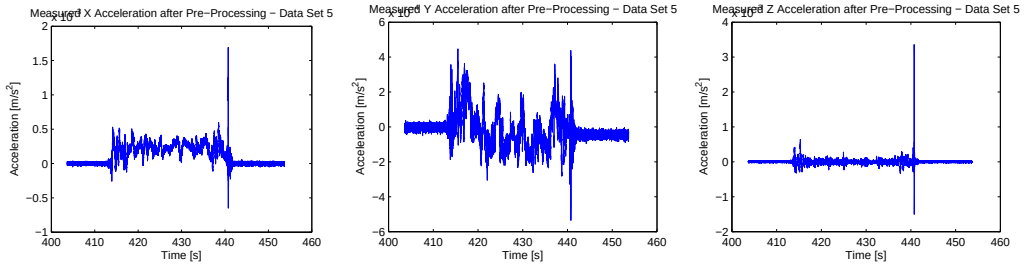
(d) Set 2: X Acceleration. (e) Set 2: Y Acceleration. (f) Set 2: Z Acceleration.



(g) Set 3: X Acceleration. (h) Set 3: Y Acceleration. (i) Set 3: Z Acceleration.

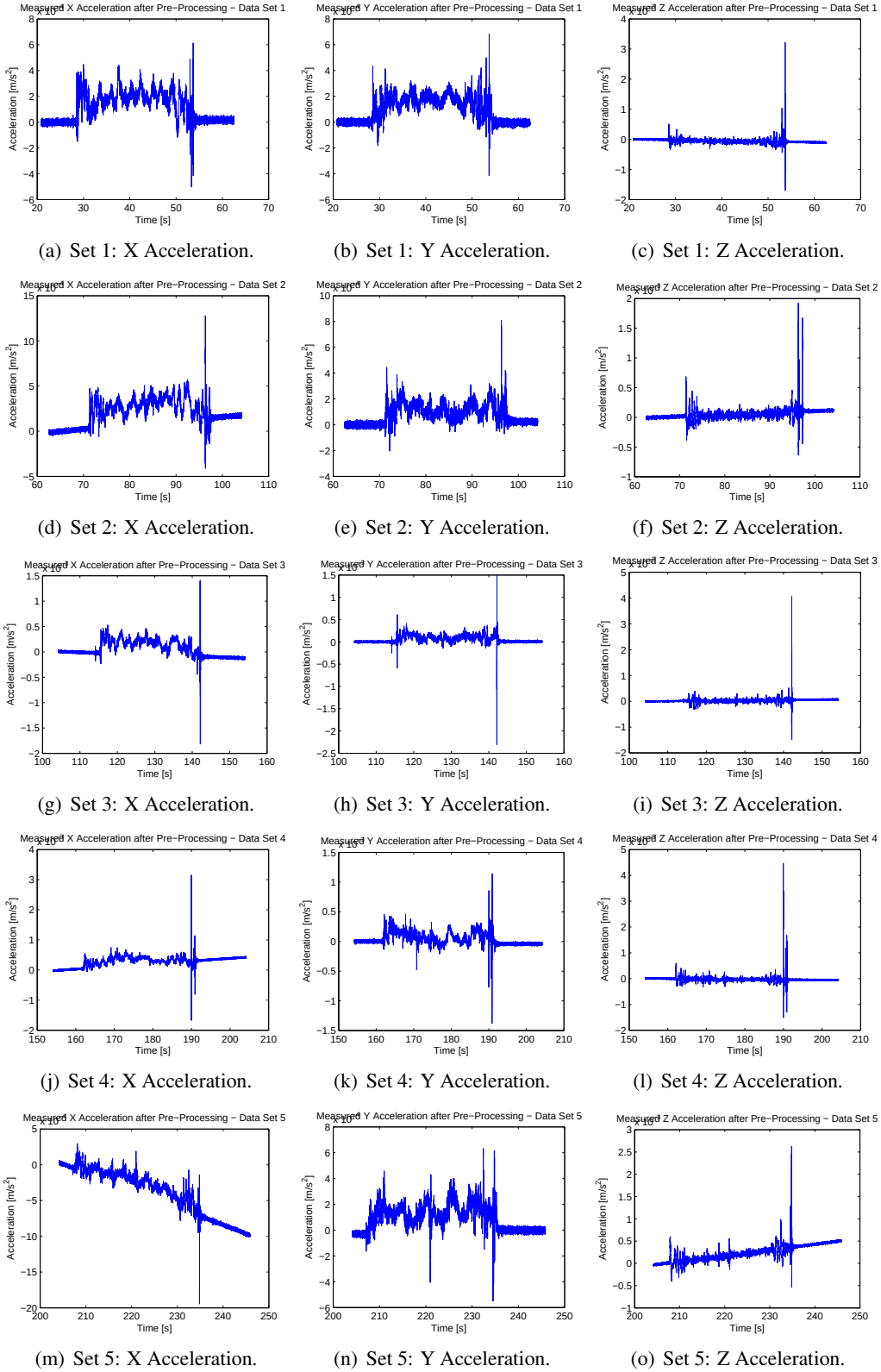


(j) Set 4: X Acceleration. (k) Set 4: Y Acceleration. (l) Set 4: Z Acceleration.



(m) Set 5: X Acceleration. (n) Set 5: Y Acceleration. (o) Set 5: Z Acceleration.

Figure 117: Measured accelerations after pre-processing.



(a) Set 1: X Acceleration. (b) Set 1: Y Acceleration. (c) Set 1: Z Acceleration.

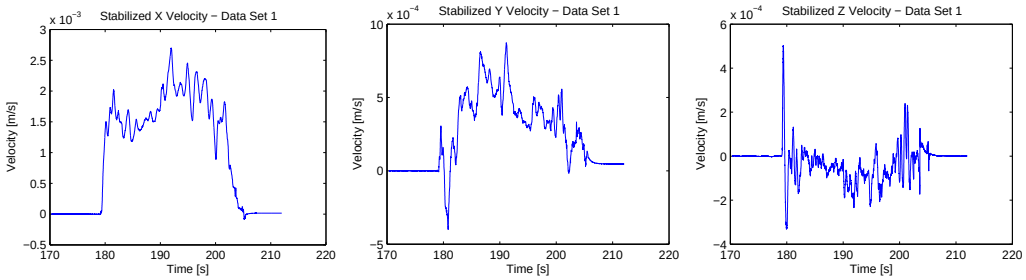
(d) Set 2: X Acceleration. (e) Set 2: Y Acceleration. (f) Set 2: Z Acceleration.

(g) Set 3: X Acceleration. (h) Set 3: Y Acceleration. (i) Set 3: Z Acceleration.

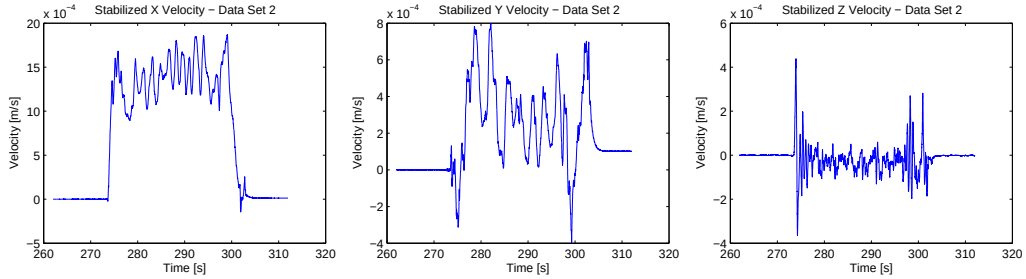
(j) Set 4: X Acceleration. (k) Set 4: Y Acceleration. (l) Set 4: Z Acceleration.

(m) Set 5: X Acceleration. (n) Set 5: Y Acceleration. (o) Set 5: Z Acceleration.

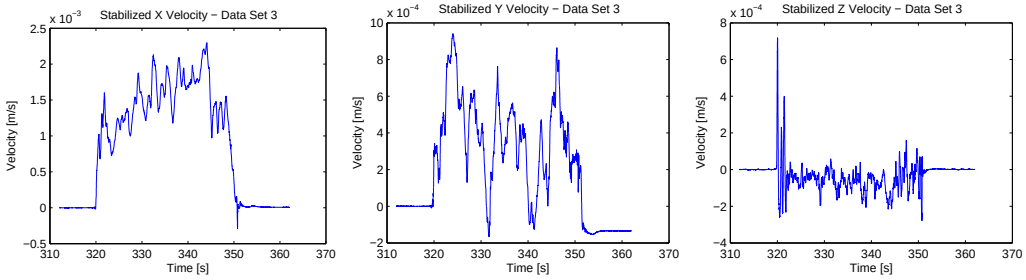
Figure 118: Measured accelerations after pre-processing.



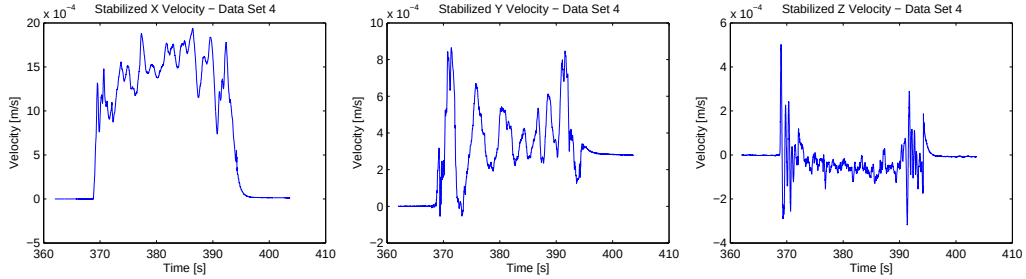
(a) Set 1: Stabilized X Velocity. (b) Set 1: Stabilized Y Velocity. (c) Set 1: Stabilized Z Velocity.



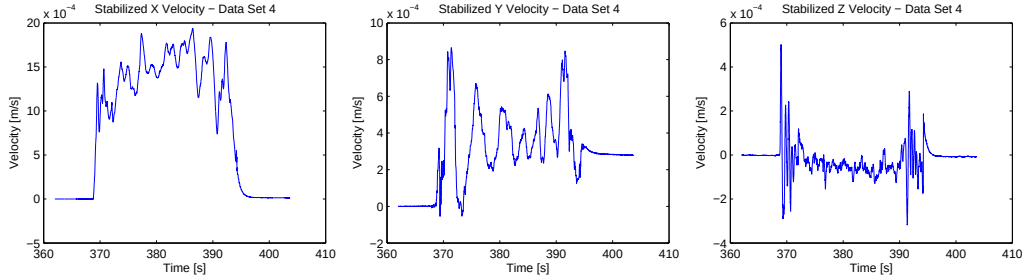
(d) Set 2: Stabilized X Velocity. (e) Set 2: Stabilized Y Velocity. (f) Set 2: Stabilized Z Velocity.



(g) Set 3: Stabilized X Velocity. (h) Set 3: Stabilized Y Velocity. (i) Set 3: Stabilized Z Velocity.

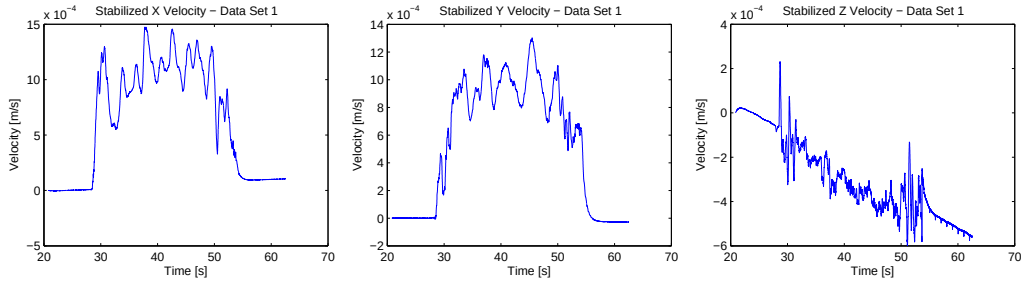


(j) Set 4: Stabilized X Velocity. (k) Set 4: Stabilized Y Velocity. (l) Set 4: Stabilized Z Velocity.

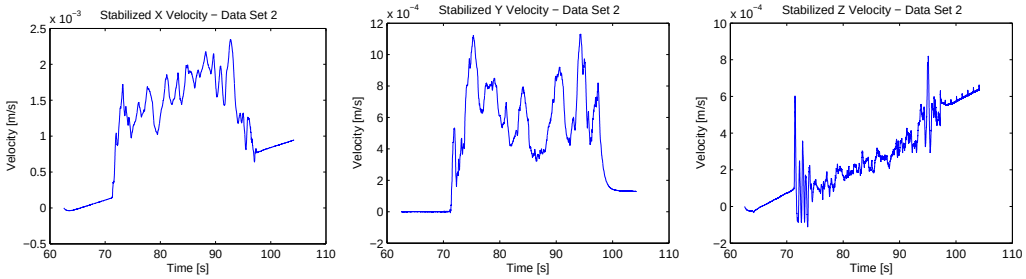


(m) Set 5: Stabilized X Velocity. (n) Set 5: Stabilized Y Velocity. (o) Set 5: Stabilized Z Velocity.

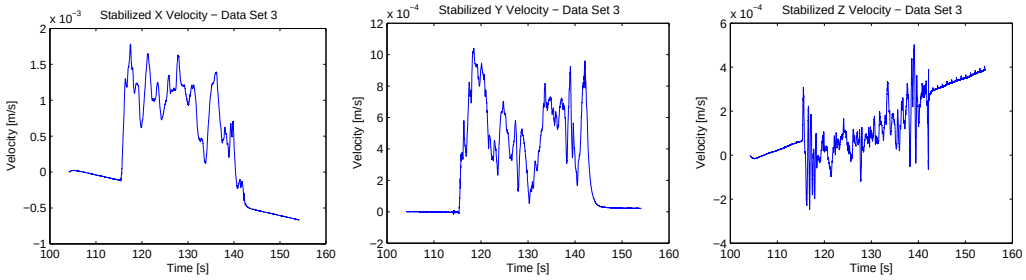
Figure 119: Stabilized velocities after bias and trend removal.



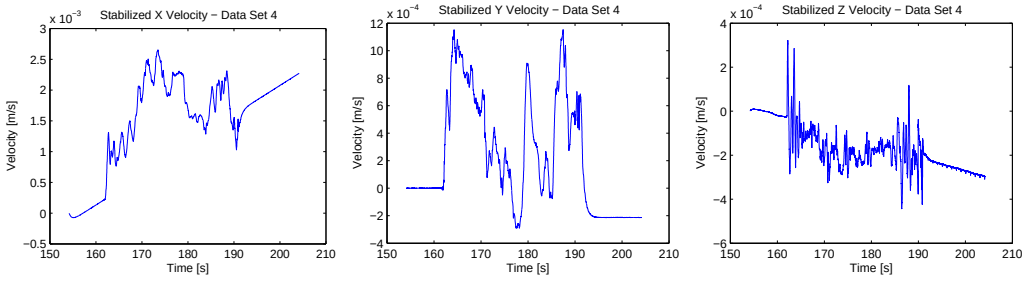
(a) Set 1: Stabilized X Velocity. (b) Set 1: Stabilized Y Velocity. (c) Set 1: Stabilized Z Velocity.



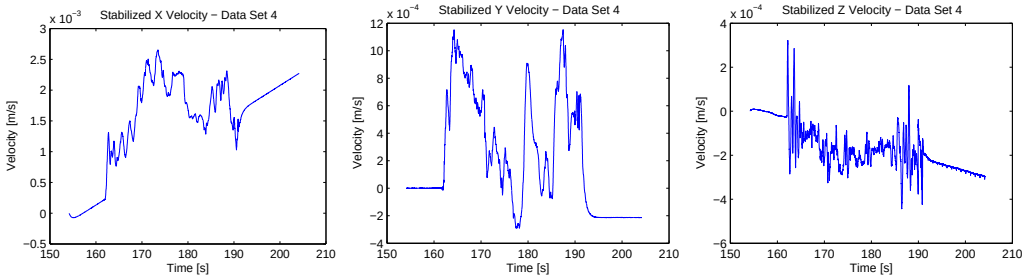
(d) Set 2: Stabilized X Velocity. (e) Set 2: Stabilized Y Velocity. (f) Set 2: Stabilized Z Velocity.



(g) Set 3: Stabilized X Velocity. (h) Set 3: Stabilized Y Velocity. (i) Set 3: Stabilized Z Velocity.

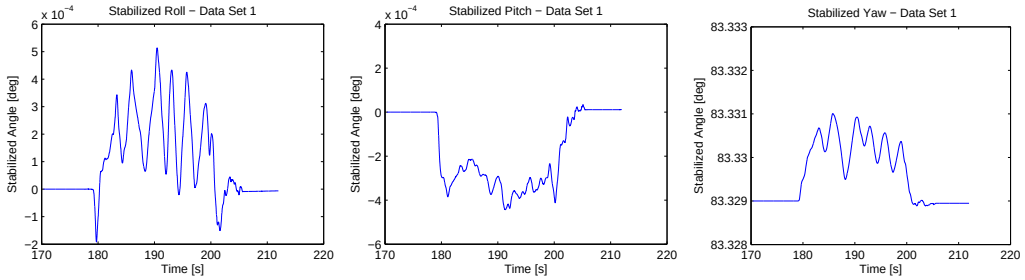


(j) Set 4: Stabilized X Velocity. (k) Set 4: Stabilized Y Velocity. (l) Set 4: Stabilized Z Velocity.

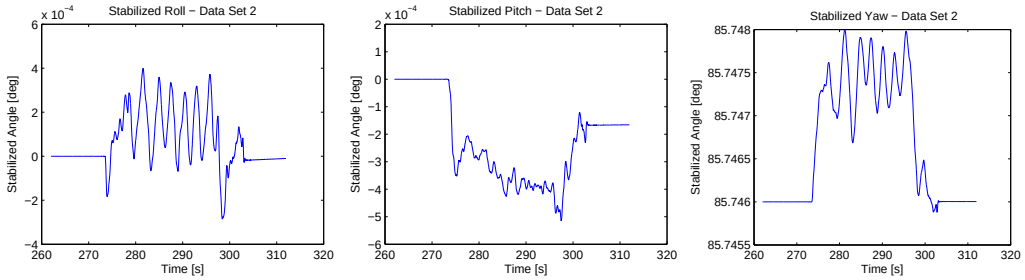


(m) Set 5: Stabilized X Velocity. (n) Set 5: Stabilized Y Velocity. (o) Set 5: Stabilized Z Velocity.

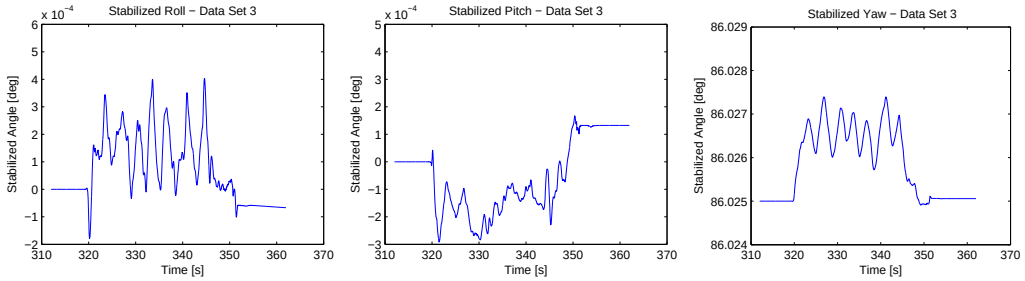
Figure 120: Stabilized velocities after bias and trend removal.



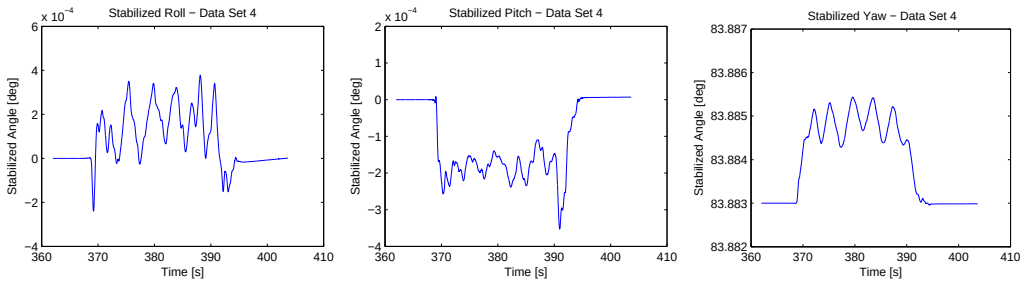
(a) Set 1: Stabilized Roll. (b) Set 1: Stabilized Pitch. (c) Set 1: Stabilized Yaw.



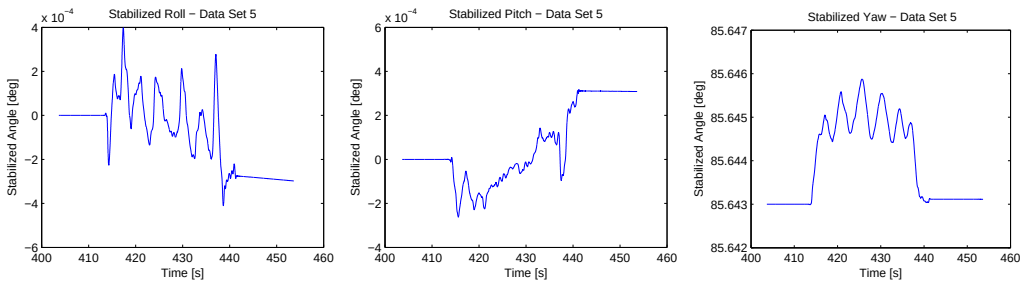
(d) Set 2: Stabilized Roll. (e) Set 2: Stabilized Pitch. (f) Set 2: Stabilized Yaw.



(g) Set 3: Stabilized Roll. (h) Set 3: Stabilized Pitch. (i) Set 3: Stabilized Yaw.

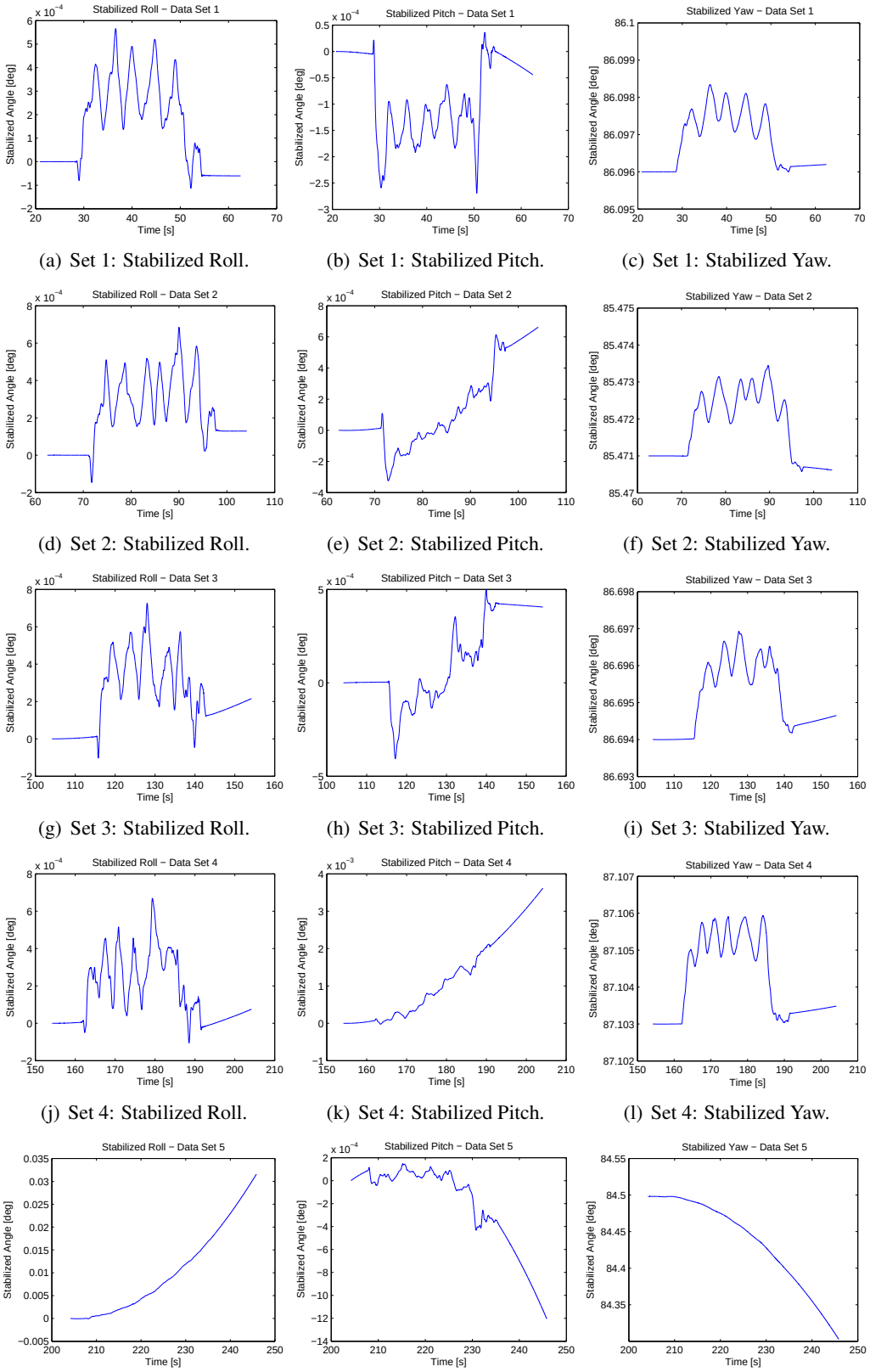


(j) Set 4: Stabilized Roll. (k) Set 4: Stabilized Pitch. (l) Set 4: Stabilized Yaw.



(m) Set 5: Stabilized Roll. (n) Set 5: Stabilized Pitch. (o) Set 5: Stabilized Yaw.

Figure 121: Stabilized attitude angles (roll, pitch, yaw) after bias and trend removal.



(a) Set 1: Stabilized Roll. (b) Set 1: Stabilized Pitch. (c) Set 1: Stabilized Yaw.

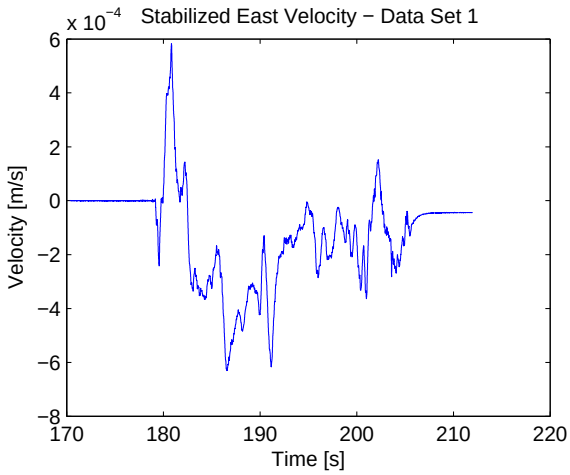
(d) Set 2: Stabilized Roll. (e) Set 2: Stabilized Pitch. (f) Set 2: Stabilized Yaw.

(g) Set 3: Stabilized Roll. (h) Set 3: Stabilized Pitch. (i) Set 3: Stabilized Yaw.

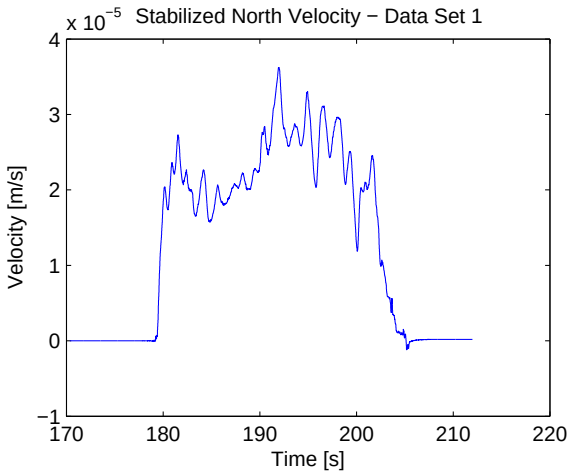
(j) Set 4: Stabilized Roll. (k) Set 4: Stabilized Pitch. (l) Set 4: Stabilized Yaw.

(m) Set 5: Stabilized Roll. (n) Set 5: Stabilized Pitch. (o) Set 5: Stabilized Yaw.

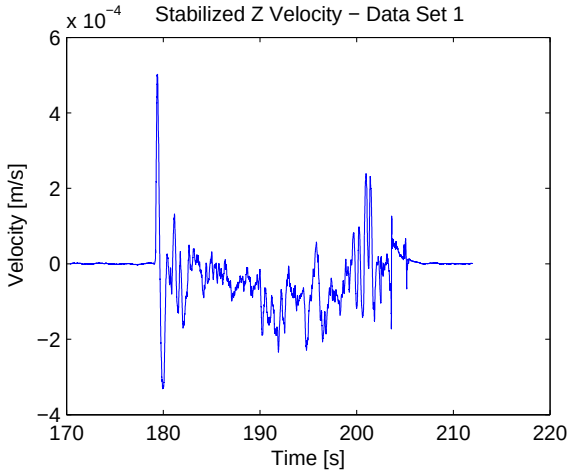
Figure 122: Stabilized attitude angles (roll, pitch, yaw) after bias and trend removal.



(a) Stabilized East velocity.

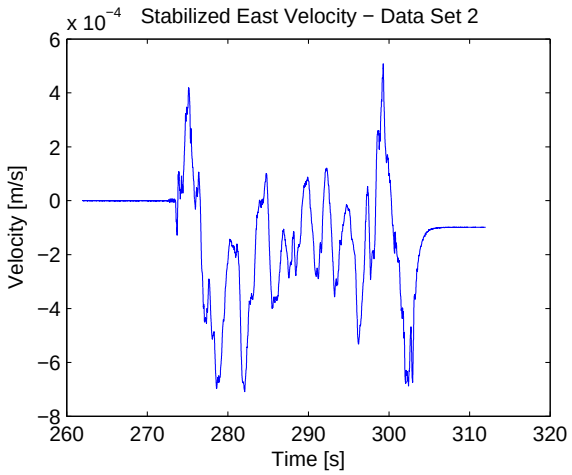


(b) Stabilized North velocity.

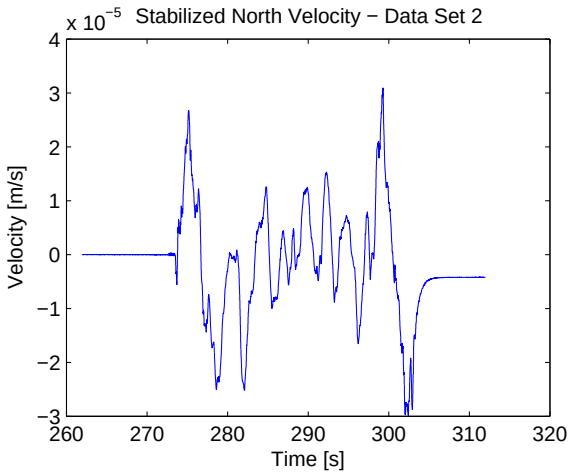


(c) Stabilized Down velocity.

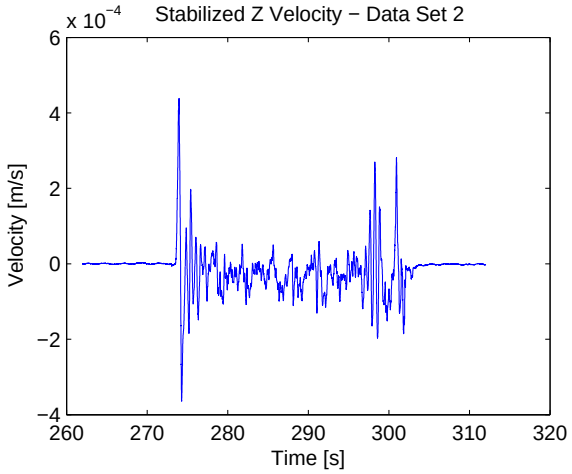
Figure 123: Stabilized velocities (in Earth frame) for data set 1, morning collection.



(a) Stabilized East velocity.

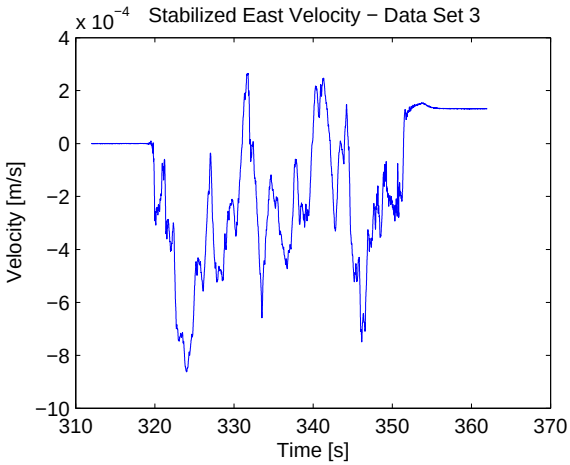


(b) Stabilized North velocity.

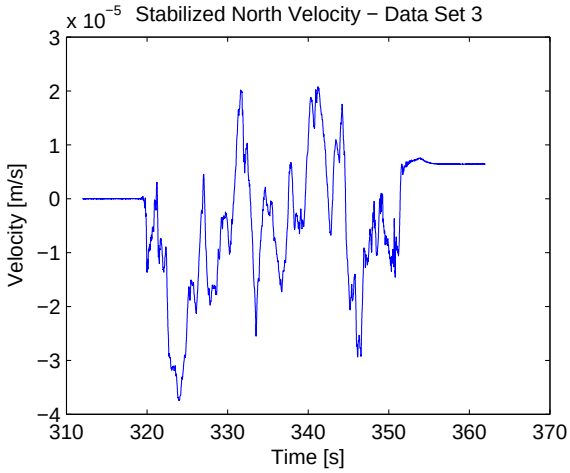


(c) Stabilized Down velocity.

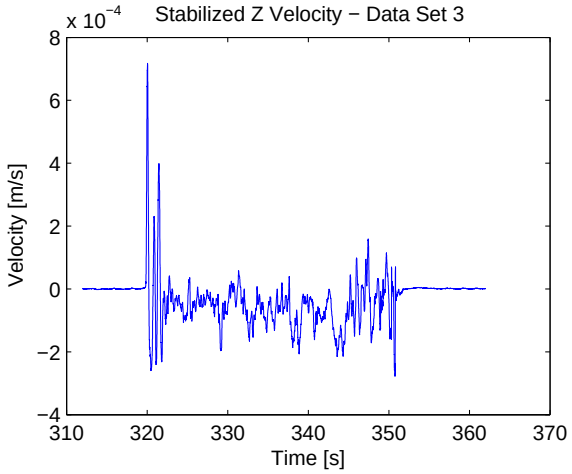
Figure 124: Stabilized velocities (in Earth frame) for data set 2, morning collection.



(a) Stabilized East velocity.

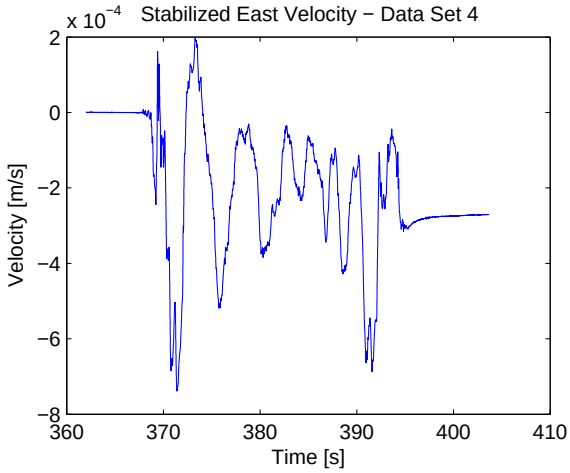


(b) Stabilized North velocity.

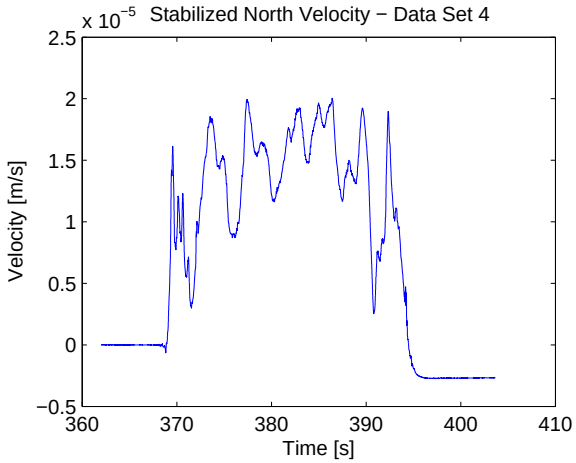


(c) Stabilized Down velocity.

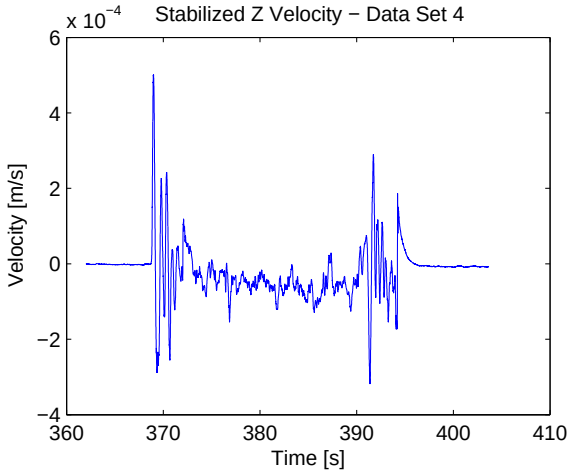
Figure 125: Stabilized velocities (in Earth frame) for data set 3, morning collection.



(a) Stabilized East velocity.

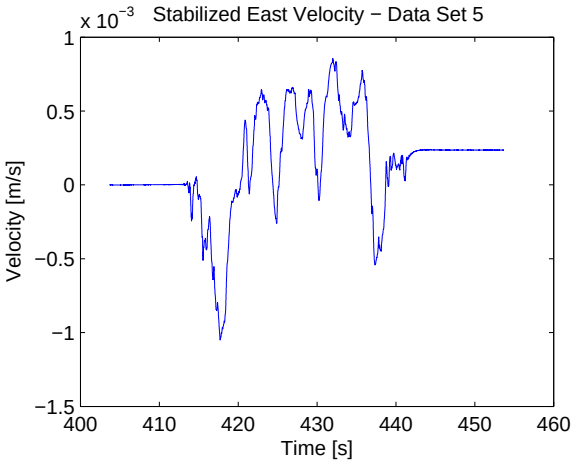


(b) Stabilized North velocity.

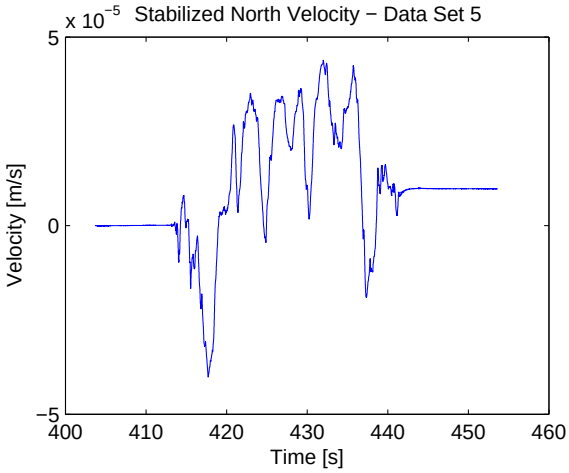


(c) Stabilized Down velocity.

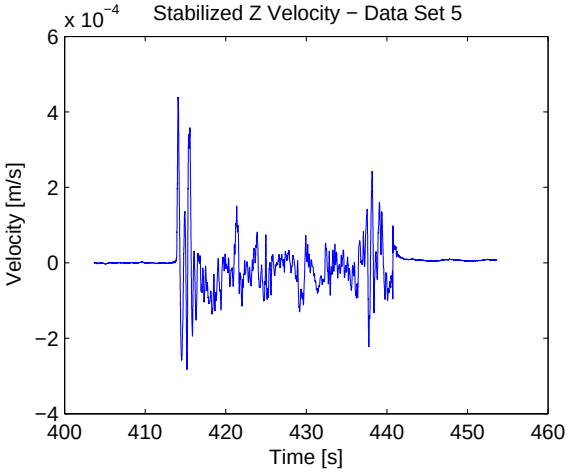
Figure 126: Stabilized velocities (in Earth frame) for data set 4, morning collection.



(a) Stabilized East velocity.

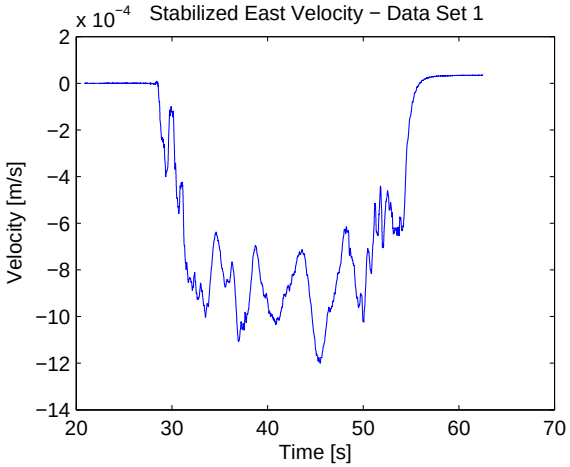


(b) Stabilized North velocity.

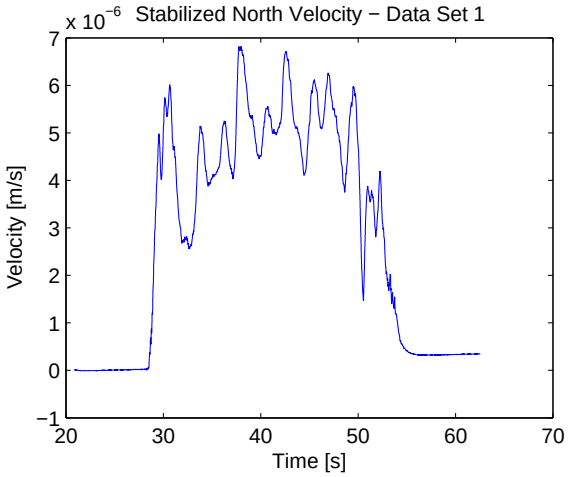


(c) Stabilized Down velocity.

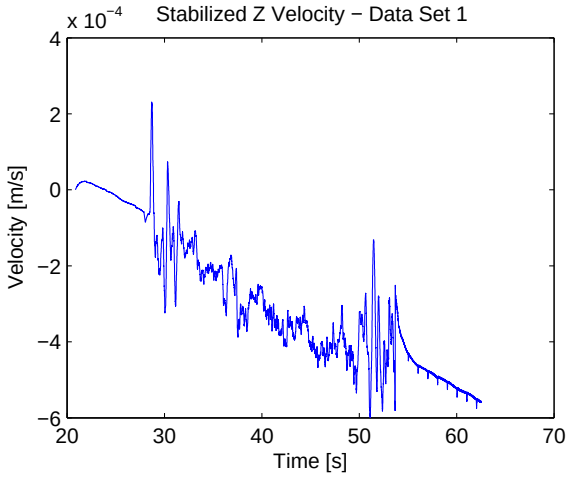
Figure 127: Stabilized velocities (in Earth frame) for data set 5, morning collection.



(a) Stabilized East velocity.

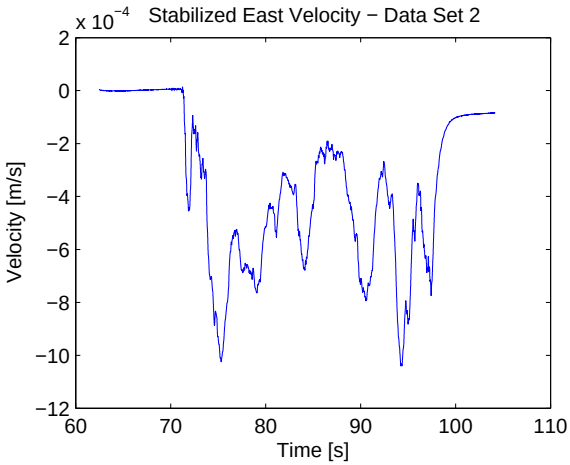


(b) Stabilized North velocity.

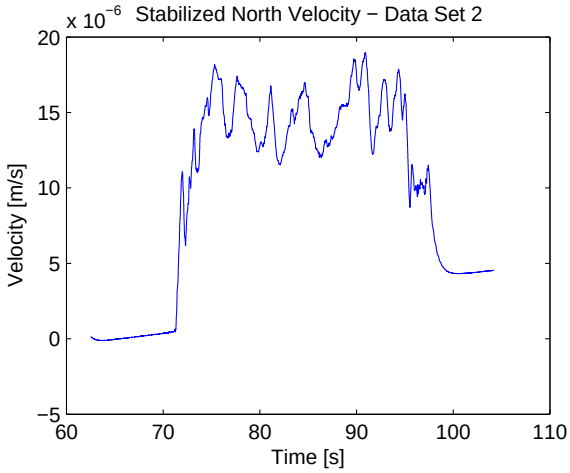


(c) Stabilized Down velocity.

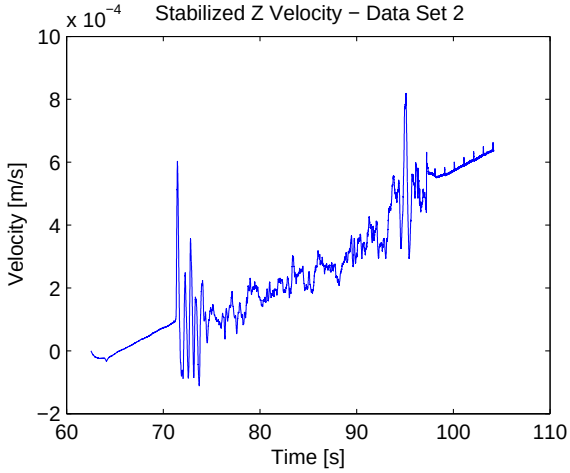
Figure 128: Stabilized velocities (in Earth frame) for data set 1, afternoon collection.



(a) Stabilized East velocity.

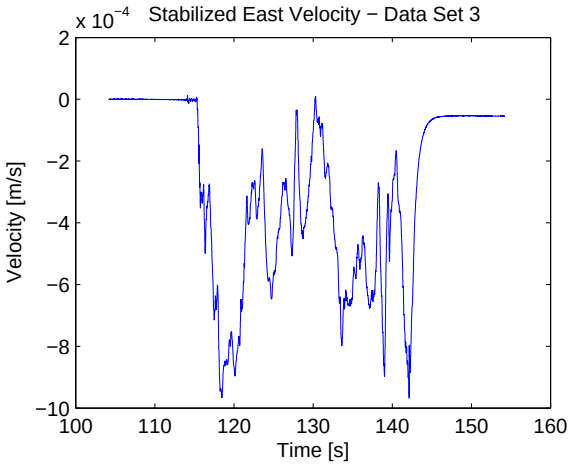


(b) Stabilized North velocity.

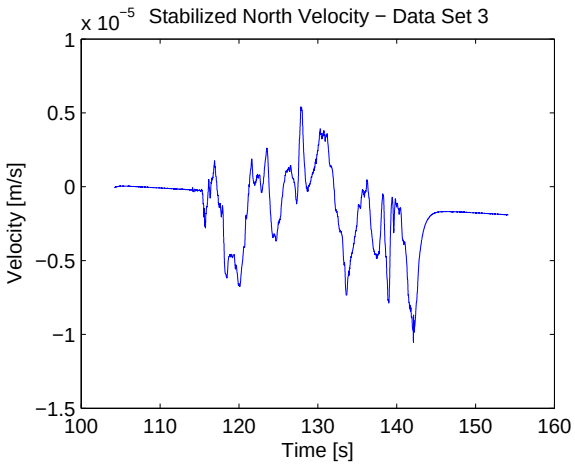


(c) Stabilized Down velocity.

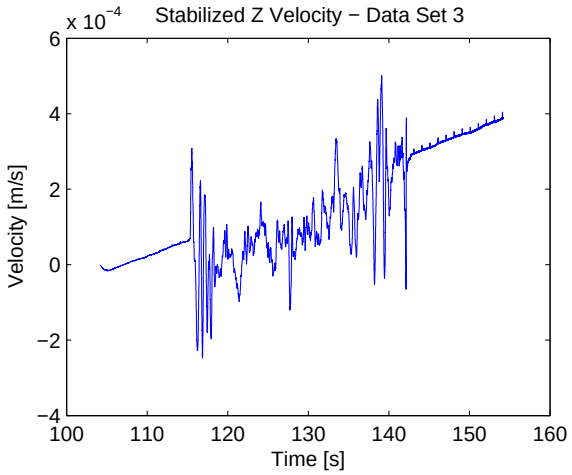
Figure 129: Stabilized velocities (in Earth frame) for data set 2, afternoon collection.



(a) Stabilized East velocity.

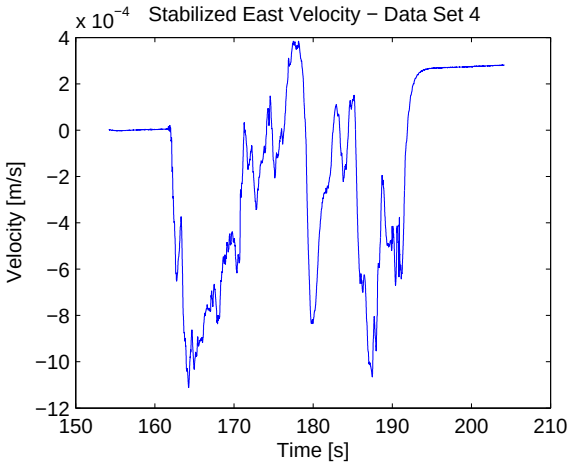


(b) Stabilized North velocity.

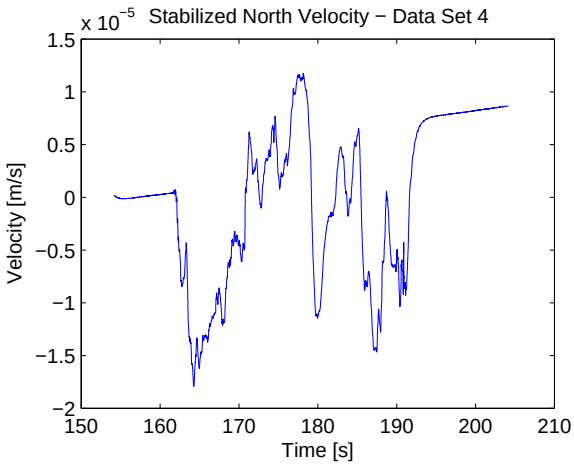


(c) Stabilized Down velocity.

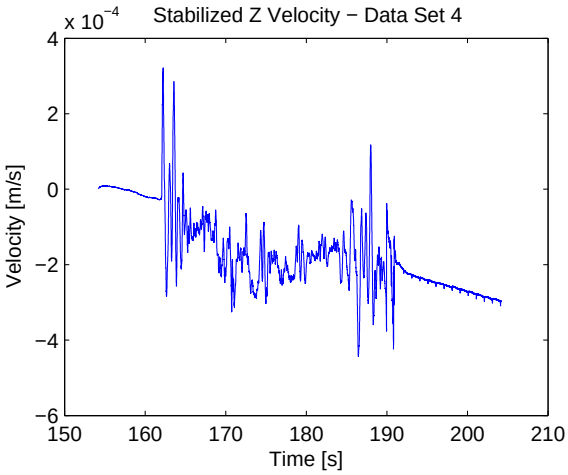
Figure 130: Stabilized velocities (in Earth frame) for data set 3, afternoon collection.



(a) Stabilized East velocity.

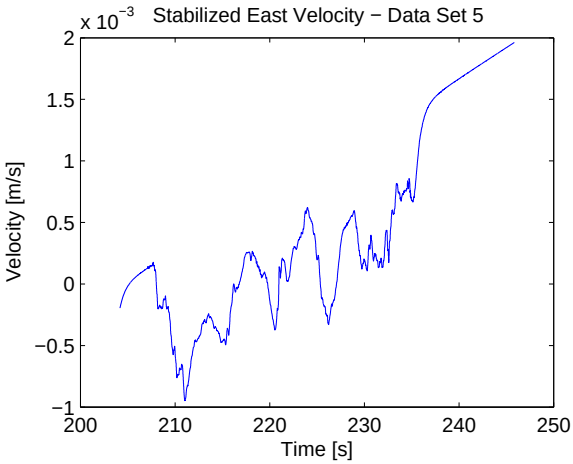


(b) Stabilized North velocity.

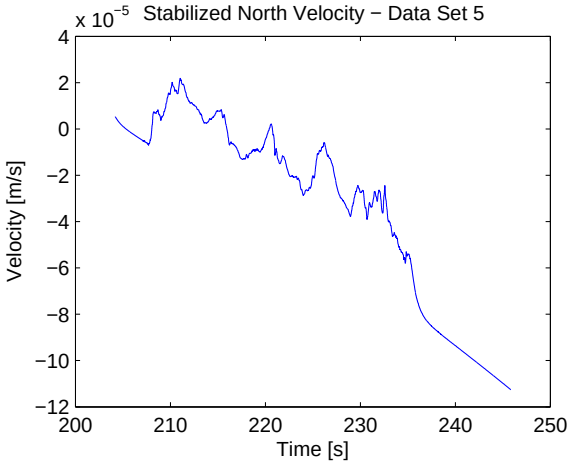


(c) Stabilized Down velocity.

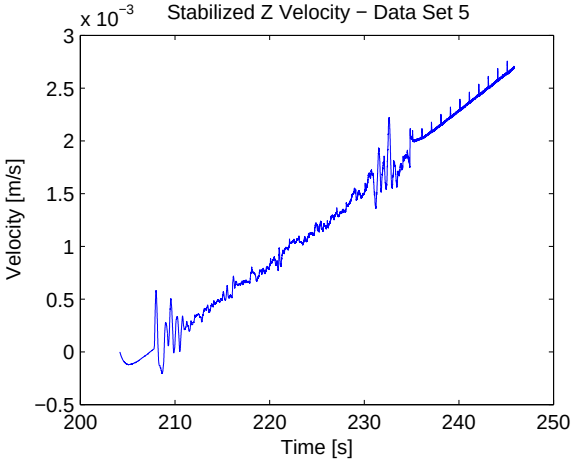
Figure 131: Stabilized velocities (in Earth frame) for data set 4, afternoon collection.



(a) Stabilized East velocity.

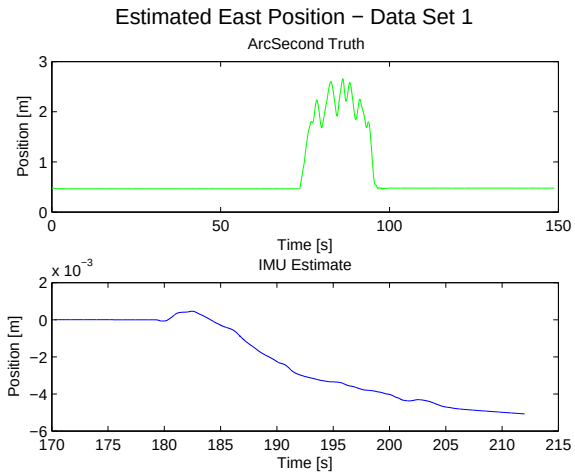


(b) Stabilized North velocity.

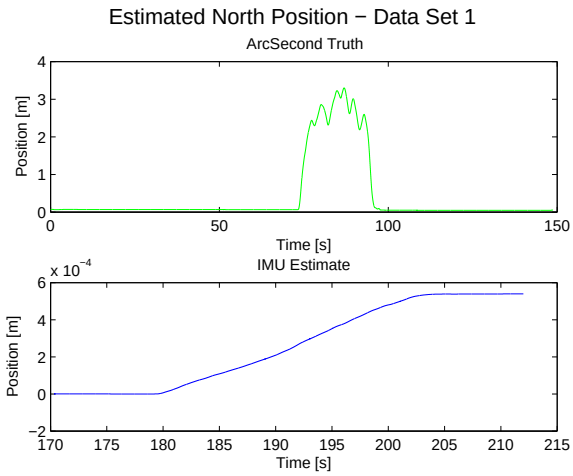


(c) Stabilized Down velocity.

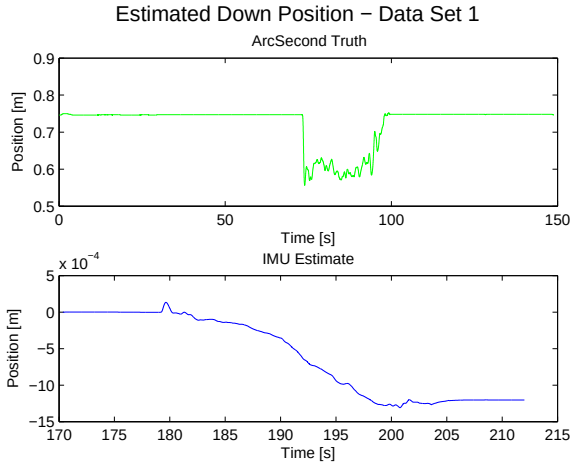
Figure 132: Stabilized velocities (in Earth frame) for data set 5, afternoon collection.



(a) Estimated East position.

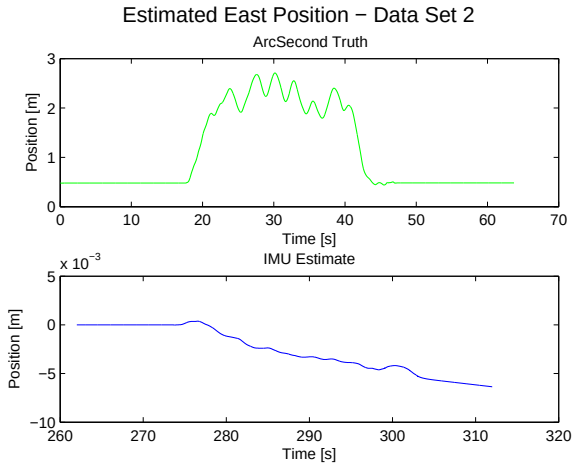


(b) Estimated North position.

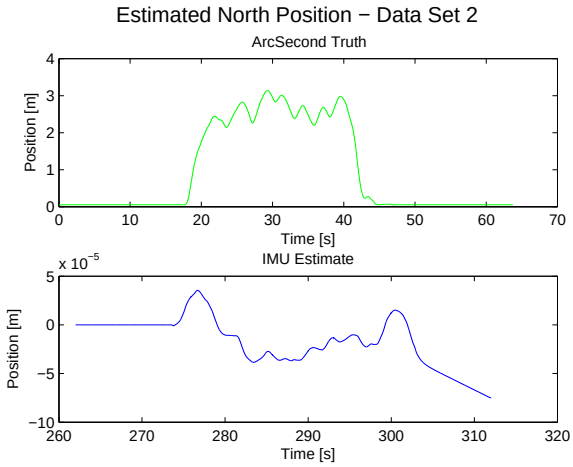


(c) Estimated Down position.

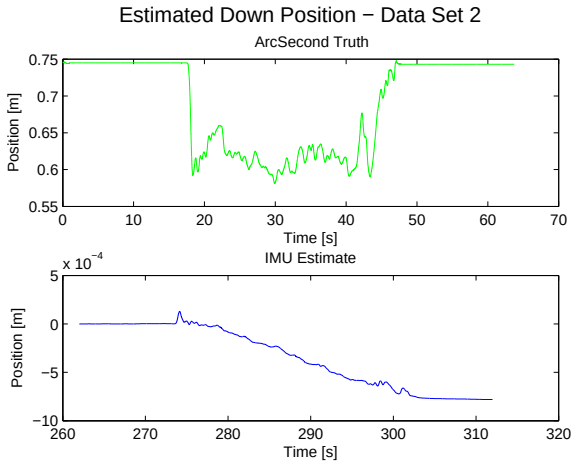
Figure 133: Estimated positions (in Earth frame) and comparison to ground truth for data set 1, morning collection.



(a) Estimated East position.

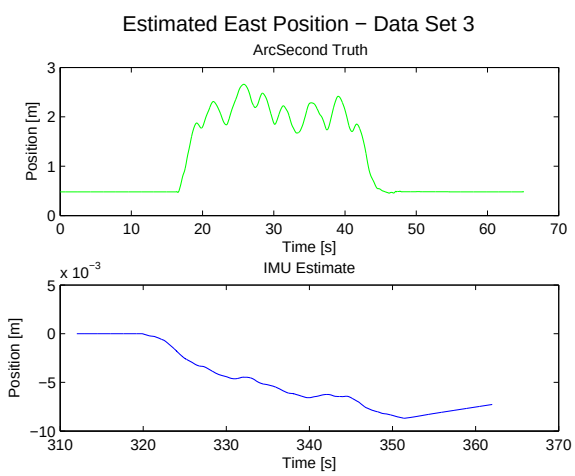


(b) Estimated North position.

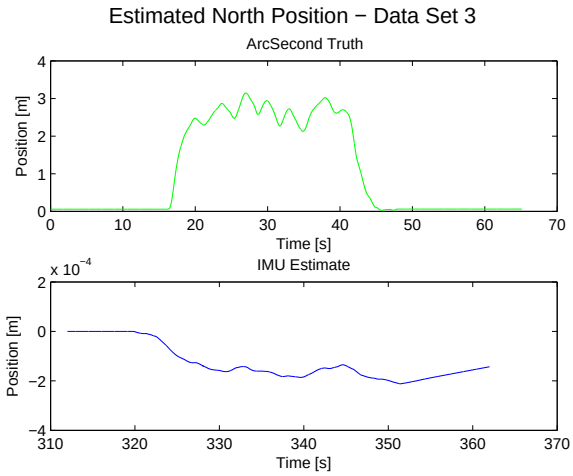


(c) Estimated Down position.

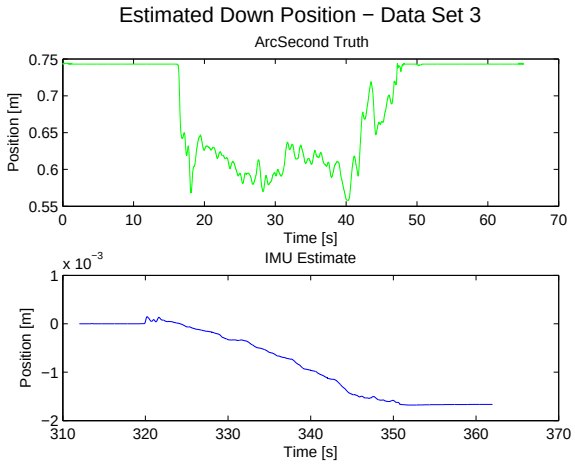
Figure 134: Estimated positions (in Earth frame) and comparison to ground truth for data set 2, morning collection.



(a) Estimated East position.

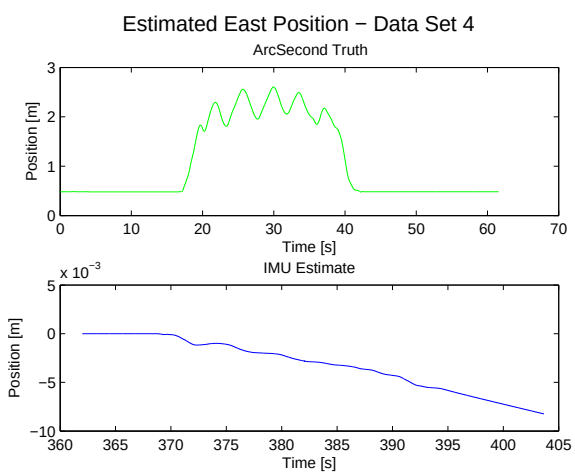


(b) Estimated North position.

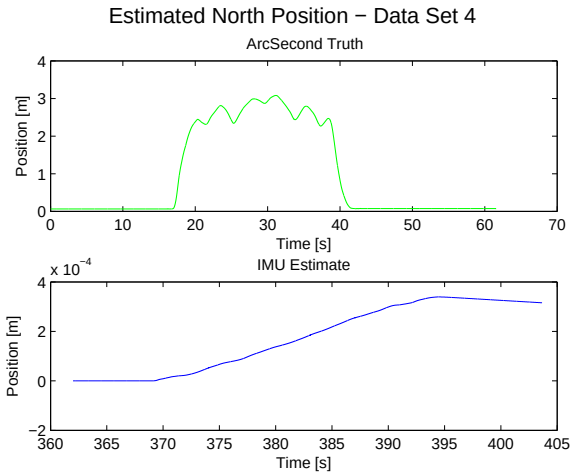


(c) Estimated Down position.

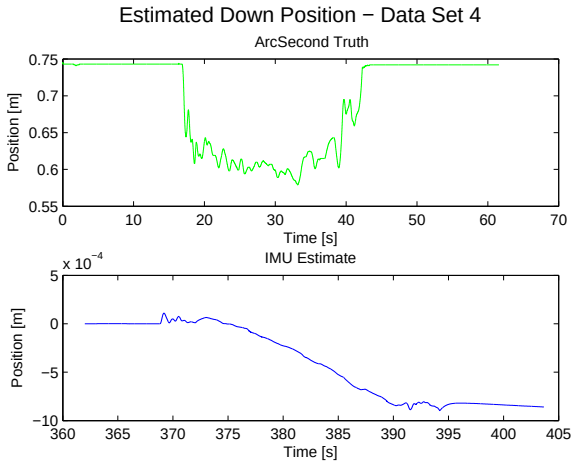
Figure 135: Estimated positions (in Earth frame) and comparison to ground truth for data set 3, morning collection.



(a) Estimated East position.

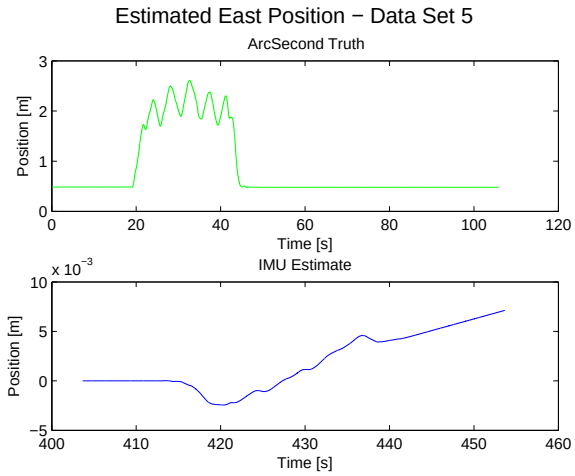


(b) Estimated North position.

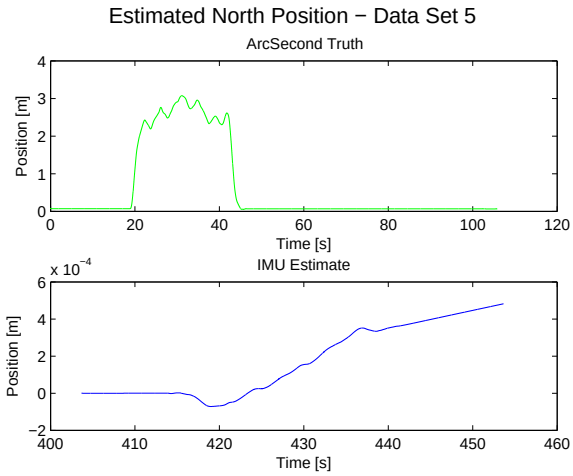


(c) Estimated Down position.

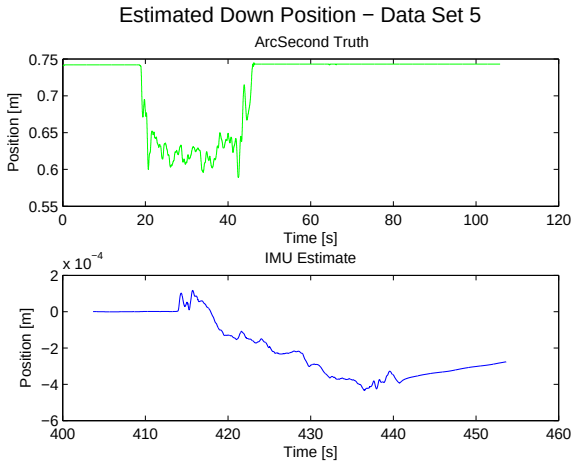
Figure 136: Estimated positions (in Earth frame) and comparison to ground truth for data set 4, morning collection.



(a) Estimated East position.

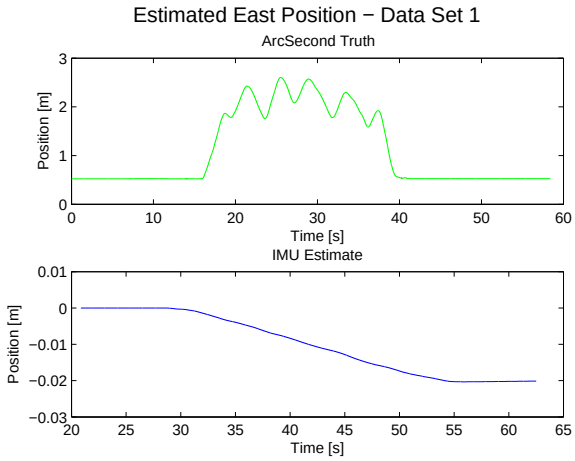


(b) Estimated North position.

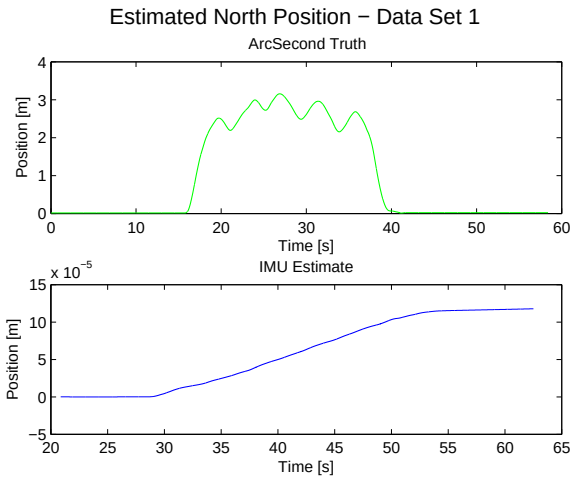


(c) Estimated Down position.

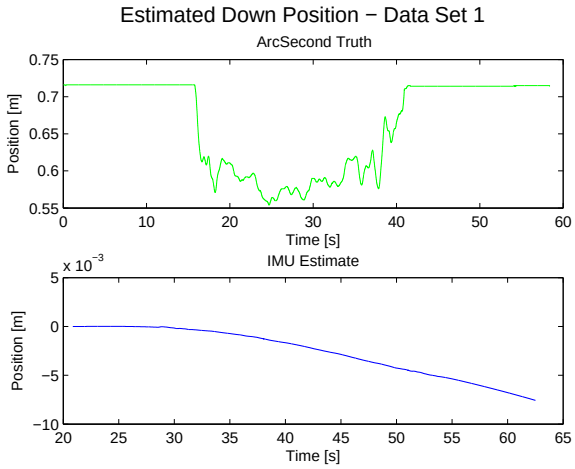
Figure 137: Estimated positions (in Earth frame) and comparison to ground truth for data set 5, morning collection.



(a) Estimated East position.

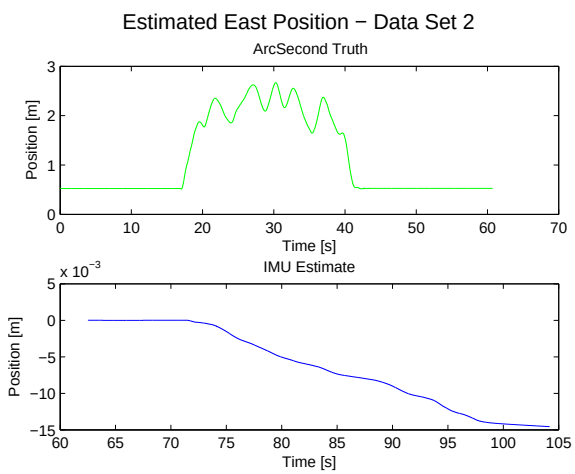


(b) Estimated North position.

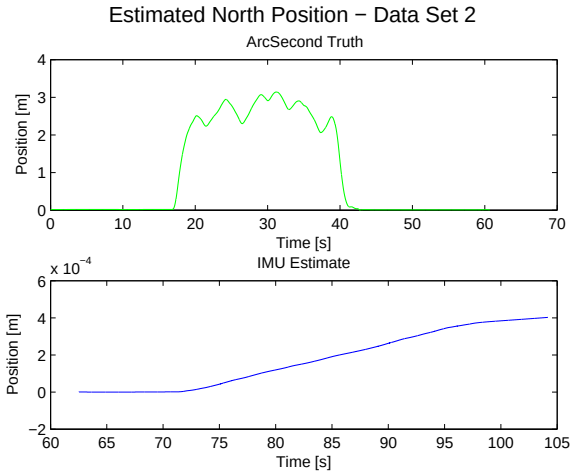


(c) Estimated Down position.

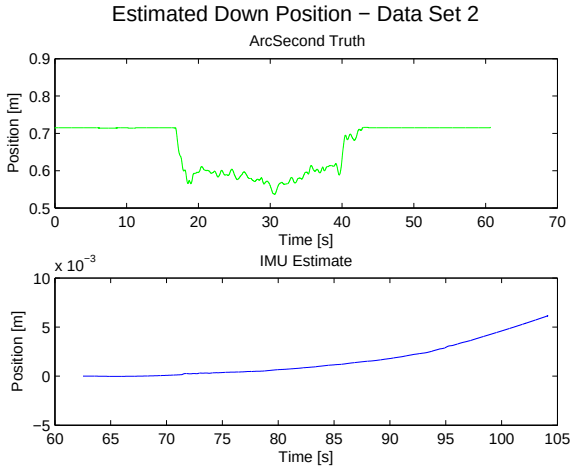
Figure 138: Estimated positions (in Earth frame) and comparison to ground truth for data set 1, afternoon collection.



(a) Estimated East position.

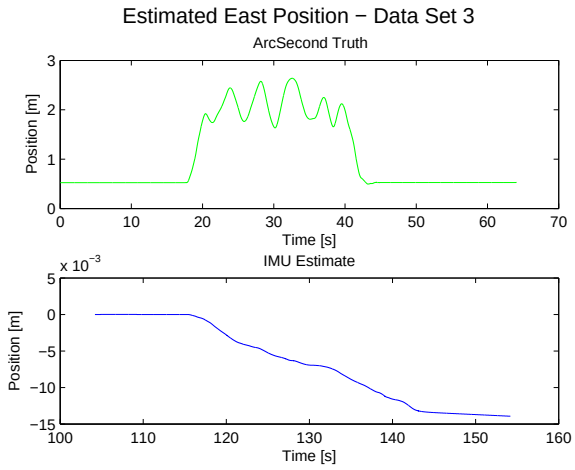


(b) Estimated North position.

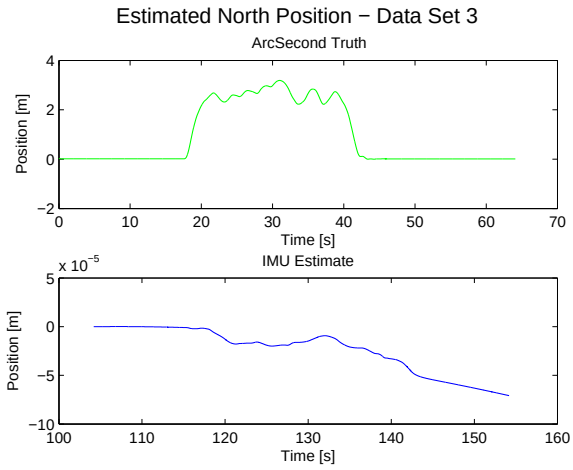


(c) Estimated down position.

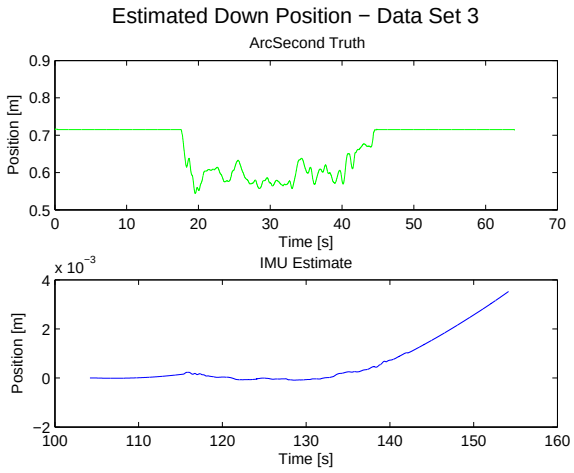
Figure 139: Estimated positions (in Earth frame) and comparison to ground truth for data set 2, afternoon collection.



(a) Estimated East position.

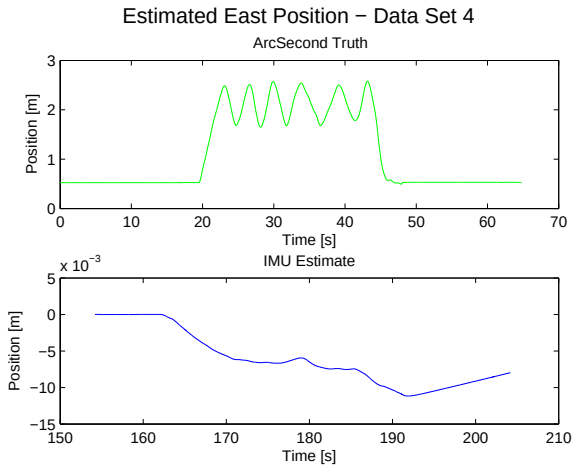


(b) Estimated North position.

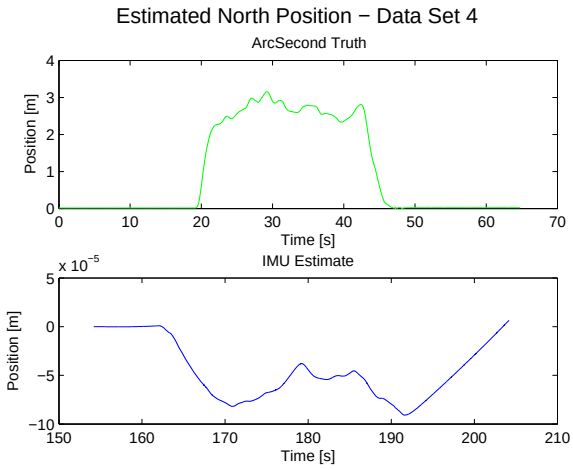


(c) Estimated Down position.

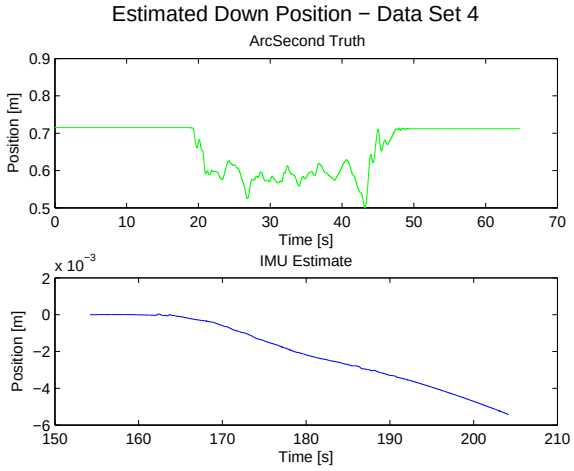
Figure 140: Estimated positions (in Earth frame) and comparison to ground truth for data set 3, afternoon collection.



(a) Estimated East position.

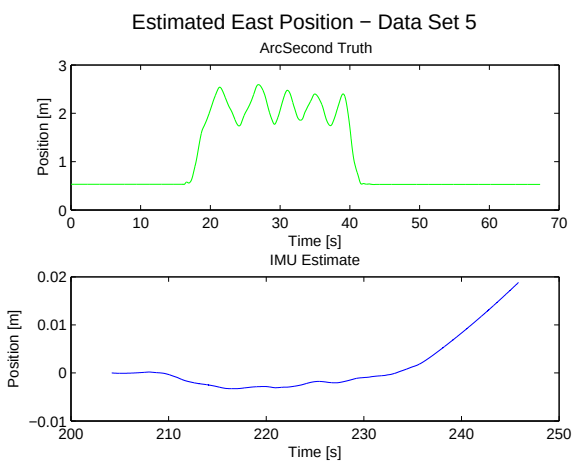


(b) Estimated North position.

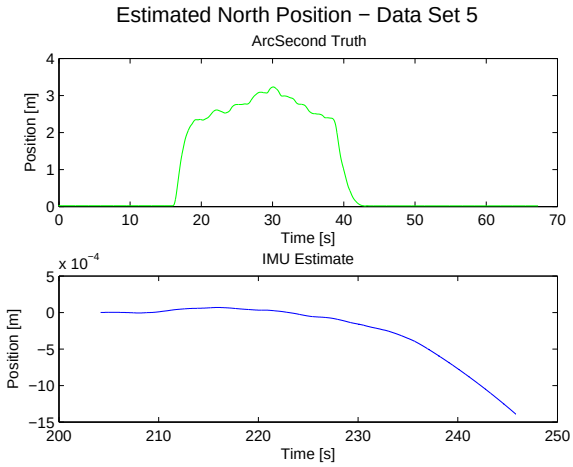


(c) Estimated Down position.

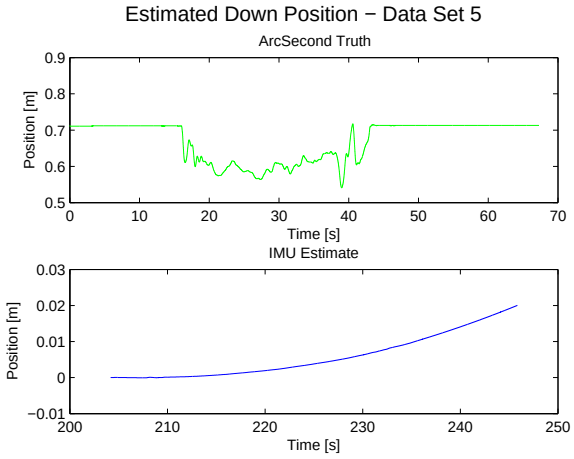
Figure 141: Estimated positions (in Earth frame) and comparison to ground truth for data set 4, afternoon collection.



(a) Estimated East position.



(b) Estimated North position.



(c) Estimated Down position.

Figure 142: Estimated positions (in Earth frame) and comparison to ground truth for data set 5, afternoon collection.

7.2 System Identification Algorithms

The system identification algorithms were applied to the data collected at APG. Again, the estimates of the system parameters failed to converge. Plots of the measured accelerations versus the true accelerations show characteristics similar to the plots for the Board Data; there is no clear relationship between the measured and true accelerations.

7.3 Summary

Adaptive error mitigation and system identification algorithms were applied to the IMU data collected at APG in September 2006. Prior to applying the adaptive error mitigation algorithms, they were modified to eliminate the reliance on prior knowledge of the system's motion. Specifically, the assumptions of either linear motion or motion in a perfect arc of known radius about a fixed pivot point were removed. Processing this data produced results which do not meet the desired sensor positioning performance. It is believed that not having access to prior knowledge regarding the sensor's motion adversely impacted algorithm performance.