

**A MAXIMUM EFFORT CONTROL SYSTEM FOR THE
TRACKING AND CONTROL OF A GUIDED CANINE**

by

Jeffrey David Miller

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 13, 2010

Keywords: cyborg, bang-bang control, canine, unmanned system,
fuzzy logic, autonomous control, EKF, GPS/INS integration

Copyright 2010 by Jeffrey David Miller

Approved by

David M. Bevly, Chair, Associate Professor of Mechanical Engineering
George T. Flowers, Professor of Mechanical Engineering
Song-yul Choe, Associate Professor of Mechanical Engineering

Abstract

This dissertation presents an approach for autonomously controlling a canine using an embedded command module with vibration and tone generation capabilities and an embedded control suite comprised of a microprocessor, wireless radio, GPS receiver, three accelerometers, three gyroscopes, and three magnetometers. In order to track the canine's motions, which inherently contain non-conventional noise characteristics, GPS, inertial sensor, and magnetometer measurements were integrated using a specialized Extended Kalman Filter (EKF), equipped with a Fuzzy Logic Controller for adaptive tuning of the Process Noise Covariance Matrix (Q). This allowed for rejection of unmodeled canine motion characteristics which tend to corrupt accelerometer bias tracking in a standard EKF. The EKF solution provided an optimized estimate of the canine position and velocity and also proved to be effective in tracking the canine's position and velocity during brief GPS outages. On average, the filter proved to track the canine's position with a 7.5 meter error and the canine's velocity with a 1.2 meter per second error after 10 seconds of simulated GPS outage.

Using the tracking solution, a Canine Maximum Effort Controller (CMEC) was implemented for autonomous control of the canine. The CMEC proved to be effective at guiding the canine to multiple waypoints. Results from structured and non-structured environment two waypoint trials indicated a 97.7% success rate. Three waypoint trials

resulted in a success rate of 70.1%, and the overall success rate of the control system was found to be 86.6%.

In order to determine the best orientation deviation threshold choice to be used in the CMEC in future work without resorting to trial and error, a Canine Trial Simulator (CTS) was developed based on a Canine Behavior Statistical Model (CBSM) and the CMEC. The CBSM was comprised of actual statistical information that describes a canine's behavior over time. After simulations of two and three waypoint trials and verification with previously conducted field trials, it was determined that for the canine used in this dissertation, an orientation deviation threshold between 40 and 50 degrees would be ideal for use in the CMEC.

Acknowledgments

Without the gracious assistance of Almighty God, many friends, family, and colleagues, this doctoral research simply could not have been conducted. Through His providential agencies, God directed me to this work and blessed me with breakthroughs in my research when they were needed. My wife, Julie, was very patient and encouraging throughout my graduate school experience. Without her backing, it simply would not have been worth the experience, and I thank her. Thanks to my sweet children as well, who had to sacrifice time with Daddy many times due to my work—my young daughters, Evelyn and Celeste, as well as my son, Davy, who thanks to the grace of God, was born healthy in the midst of my writing this dissertation. I also thank my parents who encouraged me to follow in my father's footsteps and pursue doctoral work and my church family, who prayed for my success.

Special thanks to Dr. David Bevely for seeing fit to select me for this research, provide funding for it, and guide me along the way. Thanks to my colleagues in the GPS and Vehicle Dynamics Laboratory for their knowledge and humor, especially Ben Clark, Dr. Matt Lashley, Dr. Will Travis, Dr. Rob Daily, and David Hodo. More specifically, I want to thank the Auburn engineering canine team that have been involved in this project, including my colleague Dr. Winard Britt for his computer science expertise and assistance, as well as undergraduate students Stephan Henning, Conrad Bass, and William Lyles for their assistance with testing and electrical engineering expertise.

Special thanks also to Dr. Paul Waggoner, Daniel Johnson, the canines, and the rest of the Auburn University Canine and Detection Research Institute for their excellent work in training canines to respond to remote control, as well as their willingness to be available to me when I needed to run tests. Thanks also to the Office of Naval Research for their financial support of this project.

Table of Contents

Abstract.....	ii	
Acknowledgments.....	iv	
List of Tables	xii	
List of Figures	xiii	
Nomenclature.....	xviii	
Chapter 1	Introduction.....	1
1.1	Prior Work	1
1.2	Overview.....	3
1.3	Contributions.....	7
Chapter 2	Materials and Methods	11
2.1	Introduction.....	11
2.2	The Canines	11
2.2.1	Introduction.....	11
2.2.2	Canine Training	12
2.2.2.1	The Trained Canine.....	13
2.2.2.2	Ethical Approval	13
2.2.2.3	Training Procedures	14
2.3	Hardware and Packaging	14
2.4	Software	18

Chapter 3	Fuzzy Logic-Tuned Navigation Algorithms for Canine Tracking	20
3.1	Introduction.....	20
3.2	The Extended Kalman Filter.....	22
3.3	Adaptive Tuning of the Process Noise Covariance Matrix using Fuzzy Logic	26
3.3.1	Prior Work	27
3.3.2	Benefits of Fuzzy Logic Control.....	29
3.3.3	Fuzzy Controller Components	30
3.3.3.1	Fuzzy Rule Base	31
3.3.3.2	Condition Interface/Fuzzifier.....	32
3.3.3.3	Inference Engine	33
3.3.3.4	Action Interface/Defuzzifier	33
3.3.4	Guidance System	34
3.3.5	Tracking Algorithms.....	34
3.3.5.1	Standard EKF Components.....	34
3.3.5.2	Fuzzy Logic Controller Components.....	39
3.3.5.2.1	Fuzzy Rule Base	40
3.3.5.2.2	Condition Interface	40
3.3.5.2.3	Inference Engine	44
3.3.5.2.4	Action Interface	44
3.3.6	Experimental Results	45
3.4	Conclusions.....	51
Chapter 4	Control Algorithm	53
4.1	The Maximum Effort Controller.....	53

4.1.1	Introduction.....	53
4.1.2	Description.....	54
4.1.3	Benefits.....	55
4.1.4	Maximum Effort Controller for the Canine.....	55
4.1.4.1	Canine MEC Description.....	55
4.1.4.2	CMEC Thresholds.....	60
4.1.4.3	Other Features of the CMEC.....	61
4.1.5	Preliminary Pedestrian-Conducted Trial Results.....	64
4.1.5.1	Trial Setup and Description.....	64
4.1.5.2	Preliminary Sample Results and Discussion.....	64
4.2	Autonomous Canine Control Results and Discussion.....	68
4.2.1	Field Trial Descriptions.....	68
4.2.2	Trial Examples.....	72
4.2.2.1	Example Two Waypoint Autonomous Canine Guidance Trial.....	72
4.2.2.2	Example Three Waypoint Autonomous Canine Guidance Trial.....	75
4.2.3	Results and Conclusion.....	79
Chapter 5	Canine Trial Simulator for CMEC Threshold Selection.....	83
5.1	Introduction.....	83
5.1.1	Canine Trial Simulator Broad Description.....	83
5.1.2	Discussion of CTS Purpose.....	83
5.1.3	Canine Behavioral Inconsistency.....	84
5.2	Canine Behavior Statistical Model.....	86

5.3	The Canine Trial Simulator	88
5.3.1	CTS Features.....	88
5.3.2	CTS Description.....	89
5.3.3	Simulated Trial Setup	92
5.3.4	Example of a CTS Three Waypoint Trial	92
5.3.5	Two Waypoint CTS Results	95
5.3.6	Three Waypoint CTS Results	98
5.3.7	CTS Results Verification	99
5.3.7.1	Two Waypoint CTS Verification.....	99
5.3.7.2	Three Waypoint CTS Verification.....	102
5.3.8	Conclusion	103
Chapter 6	Conclusions	105
6.1	Conclusion	105
6.2	Future Work	106
References	111
Appendix A	Canine Training Procedures.....	124
A.1	Equipment.....	124
A.2	Training.....	125
A.2.1	Odor Detection Training.....	125
A.2.2	Remote Tone and Vibration Control.....	127
A.2.3	Final Training Status.....	131
Appendix B	Preliminary Canine Position and Orientation Determination	133
B.1	Introduction.....	133

B.2	Canine Testing	133
B.3	Tracking Algorithm	134
B.4	Challenges.....	140
B.5	Experimental Results	148
B.6	Conclusions.....	155
Appendix C	Canine Heading Estimation Improvements Using Magnetometers.....	156
C.1	Introduction.....	156
C.2	Guidance System	156
C.3	Canine Testing	157
C.4	Improvements	157
C.5	Tracking Algorithms.....	159
C.6	Experimental Results	164
C.7	Conclusions.....	168
Appendix D	Canine Position and Velocity Estimate Improvements by Accounting for Pitch in a Canine's Motion	170
D.1	Introduction.....	170
D.2	Guidance System	171
D.3	Canine Testing	171
D.4	Tracking Algorithms.....	171
D.5	Pitch Effects	175
D.6	Canine Motion Simulator.....	179
	D.6.1 Trial Profile.....	179
	D.6.2 Noise Characteristics	180
	D.6.3 Measurement Equations.....	182

D.6.4	Results and Discussion	183
D.6.5	Conclusions.....	186
D.7	Experimental Results	186
D.8	Conclusions.....	196

List of Tables

Table 3.1	Average canine motion tracking error comparisons using standard and adaptive EKF approaches after a 10 second GPS outage, as well as two-tailed, unpaired t-test results. The canine is in odor detection mode.....	49
Table 4.1	Possible commands that can be issued to the canine	58
Table 4.2	Thresholds and descriptions for the CMEC.....	61
Table 4.3	Summary of canine MEC anomaly detection types.....	62
Table 4.4	Success rate results from the autonomously guided canine trials.....	80
Table 5.1	Canine behavioral statistics for simulator use	87
Table C.1	Various Process Noise Covariance Matrix (Q) values for pedestrian and canine applications	163
Table D.1	Average canine motion tracking error results for various sensor configurations after a 10 second GPS outage. The canine is in odor detection mode	196
Table D.2	Two-tailed, paired samples T-tests for canine position and velocity.....	196

List of Figures

Figure 1.1	High-level operational overview of the project objective.....	4
Figure 1.2	Block diagram of the autonomous canine control system	5
Figure 1.3	Mid-level overview of the hardware and information operational flow during an autonomous canine control trial.....	7
Figure 2.1	Labrador Retrievers Major (left) and Badealya (right).....	12
Figure 2.2	Rabbit RCM 4100 microprocessor	15
Figure 2.3	XBee-Pro wireless module	15
Figure 2.4	The battery for the guidance harness (left), the box containing the microprocessor, wireless radio module, and GPS receiver (right), and the inertial sensors with magnetometers (orange unit)	16
Figure 2.5	The command module with hand-held radio for remote control capability	16
Figure 2.6	U-Blox GPS receiver	17
Figure 2.7	Canine Major wearing guidance harness	17
Figure 2.8	High-level overview of the hardware to software interfacing architecture.....	18
Figure 3.1	Overview of the Extended Kalman Filter	25
Figure 3.2	Fuzzy Logic Controller diagram.....	31
Figure 3.3	Visual illustration of northern and eastern velocity determination.....	36
Figure 3.4	Input (Velocity) membership functions.....	42
Figure 3.5	Output membership functions (not to scale).....	44

Figure 3.6	Fuzzy logic adaptively tuned EKF versus standard EKF and GPS only position estimates. A simulated GPS outage occurs during the last 10 seconds of the trial and is illustrated by the red and green lines.....	46
Figure 3.7	The differences between the fuzzy logic adaptively tuned EKF and GPS position measurements, as well as the difference between the standard EKF and GPS position estimates. A simulated GPS outage occurs during The last 10 seconds of the trial. Commencement of the simulated GPS outage is illustrated by the black line.....	47
Figure 3.8	Fuzzy logic adaptively tuned and standard EKF accelerometer bias estimates. A simulated GPS outage occurs during the last 10 seconds of the trial, commencing at the black line	48
Figure 4.1	Maximum effort controller output	54
Figure 4.2	Simplified canine CMEC flowchart for one waypoint	57
Figure 4.3	Preliminary pedestrian-conducted MEC position trial results for a 45 degree waypoint	65
Figure 4.4	Preliminary pedestrian-conducted MEC heading, orientation deviation, and controller output trial results for a 45 degree waypoint	66
Figure 4.5	Preliminary pedestrian-conducted MEC position trial results for a 90 degree waypoint	67
Figure 4.6	Preliminary pedestrian-conducted MEC heading, orientation deviation, and controller output trial results for a 90 degree waypoint	68
Figure 4.7	Illustration of the environment used for the structured area trials	70
Figure 4.8	Three waypoint field trial types for non-structured environments	71
Figure 4.9	Example two waypoint trial position plot from the autonomously guided canine. The trial took place in a structured environment.....	73
Figure 4.10	Example two waypoint trial velocity, orientation deviation, and controller output plots from the autonomously guided canine. The trial took place in a structured environment.....	74
Figure 4.11	Example two waypoint trial position plot from the autonomously guided canine. The trial took place in a non-structured environment	75
Figure 4.12	Example three waypoint trial position plot from the autonomously guided canine. The trial took place in a structured environment.....	76

Figure 4.13	Example three waypoint trial velocity, orientation deviation, and controller output plots from the autonomously guided canine. The trial took place in a structured environment	77
Figure 4.14	Example three waypoint trial position plot from the autonomously guided canine. The trial took place in a non-structured environment	79
Figure 5.1	Example of a three waypoint trial CTS simulation position plot.....	93
Figure 5.2	Velocity, orientation deviation, and controller output plots for an example three waypoint trial CTS simulation	94
Figure 5.3	Success rate and average trial time simulation results for CTS two waypoint trials.....	96
Figure 5.4	Illustration of the potential effects of larger waypoint distances.....	97
Figure 5.5	Success rate and average trial time CTS simulation results for three waypoint trials.....	99
Figure 5.6	Comparison of two waypoint success rate results for the CTS versus actual trial data.....	100
Figure 5.7	Comparison of two waypoint average trial time results for the CTS versus actual trial data.....	101
Figure 5.8	Comparison of three waypoint success rate and average trial time results for the CTS versus actual trial data.....	102
Figure B.1	North and East axes	134
Figure B.2	Visual illustration of northern and eastern velocity determination.....	136
Figure B.3	Raw accelerometer measurements in automotive applications	144
Figure B.4	Raw accelerometer measurements in pedestrian applications	145
Figure B.5	Raw accelerometer measurements in canine applications	146
Figure B.6	Raw yaw rate measured from the IMU gyroscope in automotive applications	147
Figure B.7	Raw yaw rate measured from the IMU gyroscope in pedestrian applications	148
Figure B.8	Raw yaw rate measured from the IMU gyroscope in canine	

	applications	149
Figure B.9	Velocity profile for GPS measurements versus EKF estimate versus dead reckoning when the canine travels north, turns and travels east, U-turns and travels west, and then turns and travels south	150
Figure B.10	GPS measurement versus EKF estimated position results when the canine travels north, turns and travels east, U-turns and travels west, and then turns and travels south	151
Figure B.11	Velocity, accelerometer bias, heading, and gyroscope bias EKF estimate results when the canine travels north, turns and travels east, U-turns and travels west, and then turns and travels south.....	152
Figure B.12	Velocity, accelerometer bias, heading, and gyroscope bias EKF estimate results when the canine travels north, turns and travels east, turns and travels south, and then turns and travels west. GPS measurements are turned off in the EKF estimate at approximately 15 seconds (marked with a red vertical line)	153
Figure B.13	GPS measurement versus EKF estimated position results when the canine travels north, turns and travels east, turns and travels south, and then turns and travels west. GPS measurements are turned off in the EKF estimate at approximately 15 seconds (marked with a red line)	154
Figure C.1	EKF pedestrian heading results when magnetometers are not used. Simulation GPS outage occurs at approximately the green line	164
Figure C.2	EKF pedestrian heading results when magnetometers are used. Simulation GPS outage occurs at approximately the green line	166
Figure C.3	EKF canine heading results when magnetometers are not used. Simulation GPS outage occurs at approximately the green line	167
Figure C.4	EKF canine heading estimate when magnetometers are used. Simulation GPS outage occurs at approximately the green line	168
Figure D.1	Raw pitch measurement from XSens unit while canine is in odor detection mode.....	176
Figure D.2	Raw pitch measurement from XSens unit while canine is sprinting	177
Figure D.3	Canine motion simulation EKF position estimate results when incline is present after the turn. Approximate initiation of incline is marked with a green line.....	184

Figure D.4	Canine motion simulation EKF velocity and accelerometer bias estimate results when incline is present after approximately 5 seconds. Approximate initiation of incline is marked with a green line	185
Figure D.5	EKF canine velocity estimate when pitch is not accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial	187
Figure D.6	EKF canine position estimate when pitch is not accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial	188
Figure D.7	EKF canine velocity estimate when pitch is accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial	189
Figure D.8	EKF canine position estimate when pitch is accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial	190
Figure D.9	EKF canine velocity estimate when pitch is not accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.....	191
Figure D.10	EKF canine position estimate when pitch is not accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.....	192
Figure D.11	EKF canine velocity estimate when pitch is accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial	192
Figure D.12	EKF canine position estimate when pitch is accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial	193
Figure D.13	Canine position results for various sensor configurations when the canine is in odor detection mode. A simulated GPS outage occurs the final 10 seconds of the trial	194

Nomenclature

Throughout this work, variables, vectors, and matrices are italicized. The acronyms that are used are as follows.

ACURO	Animal Care and Use Review Office
AHRS	Attitude Heading Reference System
C4	Explosive
CBSM	Canine Behavior Statistical Model
CDRI	Canine Detection and Research Institute
CMEC	Canine Maximum Effort Controller
CMS	Canine Motion Simulator
CTS	Canine Trial Simulator
EKF	Extended Kalman Filter
GPS	Global Positioning System
IACUC	Institutional Animal Care and Use Committee
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
k9	Canine
MEC	Maximum Effort Controller
ONR	Office of Naval Research

PDR Pedestrian Dead-Reckoning

UGV Unmanned Ground Vehicle

USAMRMC U.S. Army Medical and Materiel Command

Chapter 1

Introduction

1.1 Prior Work

It is common knowledge that autonomous control is a subject that is of particular interest in the scientific community today. Robots, missiles, unmanned ground vehicles (e.g., military transport vehicles and farming vehicles), unmanned aerial vehicles, satellites, unmanned aquatic vehicles, and many other autonomous control applications are being investigated with notable success [1-30]. However, many engineers are becoming aware of the fact that the world around us is already replete with fully-functional, superior designs that come equipped with natural sensor suites whose designs are beyond the capability of engineering knowledge to date. An insect neurobiologist from the University of Arizona in Tucson said, “There’s a long history of trying to develop micro-robots that could be sent out as autonomous devices, but I think many engineers have realized that they can’t improve on Mother Nature” [31].

With that in mind, the scientific community has become interested in learning how to remotely control **living creatures**—i.e., bio-robots or cyborgs. Cyborg research has been conducted since the 1950’s, when Dr. Jose Delgado of Yale University implanted electrodes into the brains of bulls to stimulate the hypothalamus for control

purposes [31]. Since then, the list of remotely controlled animals using electrode implantation has grown to include sharks (i.e., spiny dogfish), rats, monkeys, mice, pigeons, cats, and even cockroaches [32-39]. Cornell University, the University of California at Berkeley, the University of Michigan, and Arizona State University at Tempe are working on developing flying insect cyborgs, including hawkmoths and green June beetles [40-42]. The University of Florida in Gainesville used electrodes to remotely control rats specifically for detection of humans (for search and rescue scenarios) and explosives [31]. M.I.T. used virtual fencing coupled with GPS for tracking and autonomously herding cows by implementing auditory cues and shock reinforcement to keep cows within a desirable area [46, 47].

Besides bio-robotic examples, there is beginning to be more interest in the prospect of remotely controlling canines [43]. Canines can traverse a variety of terrain more efficiently than humans or robots and are effective at guarding territories, carrying out search and rescue missions, as well as providing guidance for the visually impaired. They also have an amazing sense of smell that makes them capable of detecting explosives, narcotics, tobacco, pipeline leaks, retail contraband, and even cell phones and bed bugs [44]. Since engineers have not developed a device that can compare with a canine's ability to detect odors, the use of canines for these applications is appealing. Although other creatures, such as rats [31], have a keen sense of smell, canines are more appealing, especially due to their innate ability to interact with humans. Thus, using canines for these purposes is attractive to engineers, and the ability to remotely control a canine for many of these purposes is an even more attractive goal for engineers. Many scenarios could be envisioned to illustrate cases where the presence of a handler

alongside a canine would be an impossibility or undesirable (e.g., scenarios where the handler should not be visible or in harm's way). In a recent event in Afghanistan, a bomb detection canine detected an explosive a moment too late. The canine handler lost his left leg and received other serious injuries. The canine had to be euthanized due to severe injuries [43]. Remote control capability or autonomous guidance likely would have significantly altered the outcome of this unfortunate event.

The Canine Detection and Research Institute (CDRI) at Auburn University has demonstrated that canines can be remotely controlled using a canine vest equipped with a tone and vibration generator [45]. However, many cases can easily be envisioned where the canine would be out of sight from the handler (e.g., moving behind a distant building), at which time remote control capability becomes useless. Therefore, the next natural step in this research is to automate this remote control capacity (i.e., autonomous control of the canine). Since canines can traverse a variety of terrains more efficiently than humans, and possess a natural array of "sensors" used to detect and locate items of interest that robots are not readily equipped with, many aspects that pose problems to unmanned ground vehicles are inherently removed with the canine. Canines can execute the low-level decision making necessary for rerouting its local path to avoid obstacles or unfavorable terrain. Auburn University recently proved that autonomous control of canines can be done as well, using the CDRI's canine remote control system, a Global Positioning System (GPS), and machine learning algorithms [48].

1.2 Overview

This dissertation presents a novel system for autonomous control of an odor detection canine to pre-defined waypoints. Figure 1.1 illustrates a high level view of the overall operational objective. The command station determines the end waypoint and develops a path to that waypoint by setting intermediary waypoints for the canine to follow. These waypoints are transmitted wirelessly to an embedded microprocessor onboard the canine. The control system then guides the canine to each successive waypoint.

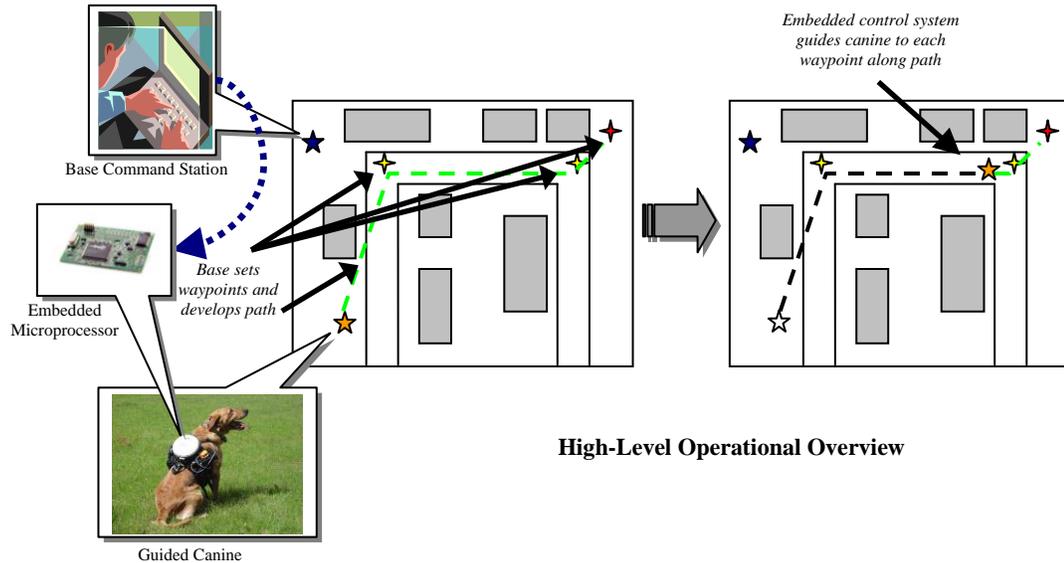


Figure 1.1. High-level operational overview of the project objective.

Figure 1.2 is a block diagram that illustrates the components of the developed control system which will be discussed in this dissertation. The symbol, ψ , represents canine heading. N represents the canine's northern position, and E represents the canine's eastern position. V represents canine velocity, and A represents its acceleration. The symbol, θ , represents canine pitch. Starting in the "Plant," the measurement devices

acquire estimates of the canine's true position, velocity, and orientation. However, due to measurement error, filtering must be done to get an optimal estimate of the canine's true motion status. These estimates are then compared to the desired position, velocity, and orientation of the canine, and the error is fed into the controller. The error is then processed, and a command is issued to the canine to guide it towards the desired waypoint. This process is looped until the canine's location status is sufficiently close to the desired state.

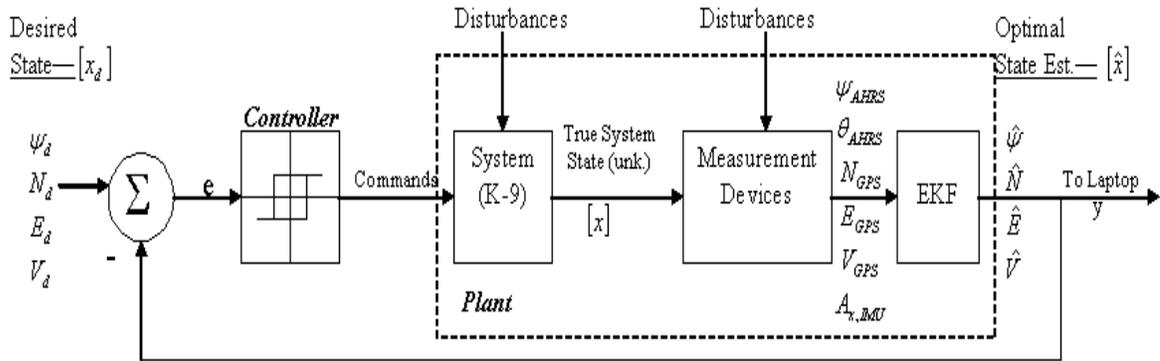


Figure 1.2. Block diagram of the autonomous canine control system.

In order to attain such a control system, several components must be acquired and/or developed. An appropriate canine must be selected and trained to respond to tone and vibrational cues that could be issued from an embedded computer system via a tone and vibration command module. Also, selection and/or development of the hardware necessary for tracking and controlling the canine must ensue. This would include an embedded microcomputer which would be responsible for computations, issuing commands, and perhaps, transferring data to a remote computational device.

Measurement devices must also be selected so that the microcomputer will be able to track the canine's position, velocity, and orientation, and in turn, issue commands for guidance. A canine vest must be developed as well, to allow for a means to house the hardware onboard the canine, and software must be developed to interface all of these hardware components. All of these control system components will be discussed in Chapter 2.

Such a control system must also include software for efficiently tracking the canine at all times, even during brief GPS outages, and a controller which receives the tracking data, processes it, and determines the appropriate command to be issued to the canine via the command module. These considerations will be addressed in Chapters 3 through 6.

With these objectives in mind, Figure 1.3 illustrates a mid-level overview, exhibiting the hardware components and information flow for the guidance system that will be discussed in this dissertation. Before the trial begins, the operator marks the waypoints for the controller (located onboard the microprocessor) through the wireless radios using the lap top. The controller can then be initiated through the lap top as well. After the trial has commenced, the lap top continues to be used as a data sink for saving trial data and allows manual override of the control algorithms. The microprocessor onboard the canine receives data from the sensor suite (i.e., GPS, the Inertial Measurement Unit or IMU, and magnetometers) and processes it, communicates with the lap top via the wireless radios, and issues commands to the command module. The command module then issues tones and vibrations to guide the canine, which in turn changes the output of the sensors, which is fed back into the microprocessor.

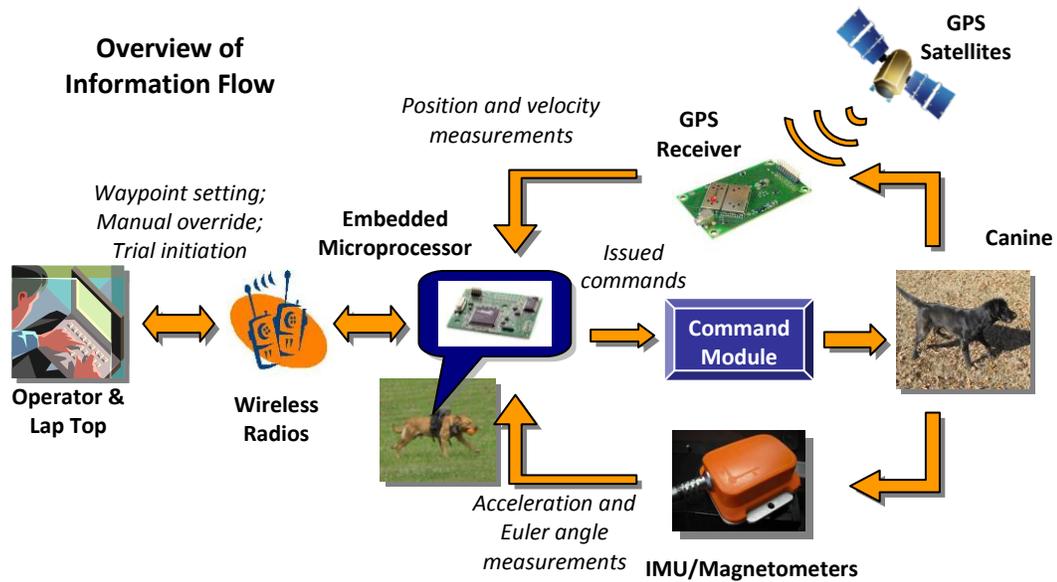


Figure 1.3. Mid-level overview of the hardware and information operational flow during an autonomous canine control trial.

1.3 Contributions

Although some of the aspects of this control system are not novel, there are several novel contributions to the scientific community that this control system affords.

- First and foremost among the contributions is the over-all canine system. As was mentioned earlier, [43] illustrated the need for a remote guidance system to keep handlers out of harm's way, and the potential of visual obstructions between the handler and canine highlights the need for making the guidance system an autonomous one. The approach discussed in this dissertation proved to be an effective one in meeting the objective.

- The system includes specialized, real-time, embedded navigation algorithms that allow for efficiently tracking a canine's position, velocity, and orientation, in spite of the unique motion characteristics that describe canine motion. These algorithms use GPS/Inertial Navigation System (INS)/Magnetometer integration and a specialized Extended Kalman Filter (EKF) that utilizes fuzzy logic to adaptively tune the Process Noise Covariance Matrix to account for different motion characteristics that are not accounted for in the canine motion model. The navigation algorithms allow for effective tracking of the canine's location and motion during brief GPS outages. Besides the canine application, the fuzzy logic tuning approach that was developed could be of benefit for other applications in which the motion model as used in the EKF is somewhat uncertain and carries the potential of significant unmodeled, unanticipated motion characteristics which could corrupt the estimation of relevant biases by the EKF.
- The control system also includes a threshold-based, embedded algorithm for autonomous control of the canine—namely, a Canine Maximum Effort Controller (CMEC). The controller is computationally inexpensive enough to run onboard the canine alongside EKF algorithms, thus eliminating the need for extensive wireless communication with a remote computational device and bypasses the need for a complicated, non-linear dynamic

model of the canine. The control system has potential applications beyond the canine scenario, having already proved to be of interest in pedestrian applications as well [98].

- In order to determine the optimal choice of the **orientation deviation threshold** for the CMEC in future trials, an off-line Canine Trial Simulator was developed based on a statistical model of a canine's behavior which takes into account a canine's average velocity, acceleration, deceleration, right and left turn rates, heading drift while moving, and directional command obedience success rates. The trial simulator allows for estimating the success rates and average trial times for different **orientation deviation threshold** choices. From these statistics, profiles can be developed to help a user quickly determine which **orientation deviation threshold** should be chosen to meet trial objectives. The simulator was found to closely approximate true trial data.
- The control system allows for guiding a canine to geometrically-complicated waypoints and multiple waypoints.
- The control system is robust in its ability to handle behavioral anomalies in canine motion and response due to its anomaly detection features.

- Implementation of the control system results in high rates for trial success for both two and three waypoint trials.

Chapter 2

Materials and Methods

2.1 Introduction

The initial stages of the control system development focused on elements of the plant portion of the block diagram in Figure 1.2—namely, the training of the canines, deciding on the hardware to use for the system, developing the packaging for the hardware, and developing the software to interface the equipment.

2.2 The Canines

2.2.1 Introduction

The dogs that participated in the autonomous control experiments were Labrador Retrievers. Canine Major (Figure 2.1—left) was the primary canine that participated in the navigation and controls portions of the project, and Badealya (Figure 2.1—right) participated in a limited role in the project due to the status of her ongoing training.

Both canines were trained to respond to and be guided by tones and vibrations. Three separate tones were used to tell the canines to go forward, stop, and return to the handler, respectively. Major was initially trained to turn and move in a direction

approximately 90 degrees from his current heading upon feeling a left or right vibrational command. However, upon request, the CDRI commenced training both canines to turn approximately 45 degrees to the right or left upon feeling a vibrational command and await further instruction. At this point, another vibrational command could be given or a forward command could be given. This allowed for finer control of the canines' headings. This training is still in progress, and its effect has not yet been totally realized. Due to the nature of a canine, the success rate in following right and left vibrational commands is not 100%, and the amount that the canine turns upon being given a vibrational command is not necessarily consistent. These factors will be discussed further in Chapter 4.



Figure 2.1. Labrador Retrievers Major (left) and Badealya (right)

2.2.2 Canine Training

The following section is a reproduction of the CDRI training protocols for canine Major that was used in this project. The same training protocols were utilized for Badealya. The protocols are reproduced here verbatim, with the author's permission.

2.2.2.1 The Trained Canine

The dog that participated in these experiments was a male Labrador Retriever named “Major” that was approximately 4-years old and weighed 32 kg. Prior to his acquisition, Major had undergone the initial stages of traditional field/hunt trial training such that he was capable of reliably executing basic “blind retrieves” (i.e., without first seeing the throwing of an object and without obvious visual cues, being directed by handler in the direction of that object through a series of voice, visual, and whistle commands to the object, which the dog then retrieves) of approximately 100 meters.

2.2.2.2 Ethical Approval

The use of Major in this experiment and other ongoing canine detection technology development efforts was approved and monitored by the Auburn University Institutional Animal Care and Use Committee (IACUC), which ensures compliance with the Animal Welfare Act (7 USC, 2131-2156). Auburn’s IACUC is approved by the Office of Laboratory Animal Welfare of the U. S. Public Health Service and Auburn’s animal housing, care, and use is inspected annually and has been approved by the Animal Welfare Division of the Animal and Plant Health Inspection Service of the U. S. Department of Agriculture. Major was housed in IACUC inspected kennels at Auburn University controlled property at Fort McClellan, AL and his care and use were conducted by Auburn University personnel who had successfully completed training in the care and use of dogs used in research or training activities. In addition to the internal IACUC surveillance, the activities in which Major has participated, his housing and care,

and the qualifications of those performing research and training activities with Major were reviewed and approved by the Animal Care and Use Review Office (ACURO) of the U. S. Army Medical and Materiel Command (USAMRMC).

2.2.2.3 Training Procedures

The specifics of the training procedures for the canines are discussed in Appendix A.

2.3 Hardware and Packaging

The goal of the hardware and packaging development phase of the project was to develop a guidance harness that is durable, light weight, and not bulky—durable to be able to withstand outdoor testing and the harsh, jarring motion characteristics of the canine; light weight to not overload the canine and decrease its stamina; not bulky to allow the canine to move into tight spaces with relative ease. The harness also needed to contain the necessary components to allow for determining the canine’s position, orientation, and velocity, as well as issuing commands to guide the canine to the selected waypoints and communicating wirelessly via a remote computational device. Considerable time and effort was put into meeting these goals. The hardware for the final guidance system is comprised of three main components: the command unit, the sensor suite, and a remote computational device.

The command unit consists of an embedded computer system, an embedded wireless radio module, and the command module. The computer system is an RCM 4100 Rabbit microprocessor core module (see Figure 2.2). This microprocessor is equipped with several serial connections, allowing

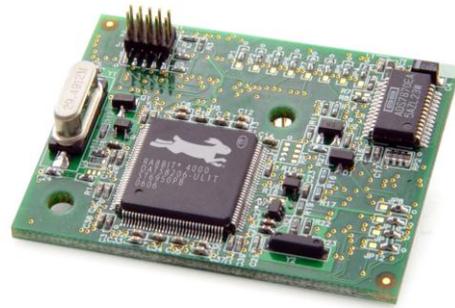


Figure 2.2. Rabbit RCM 4100 microprocessor.

interfacing between sensors and communication devices. It is small, low-cost, and requires relatively low power. However, it has limited speed and memory allocation,

which played a major role in the navigation and control phases of the research (see Chapters 3 and 4).



Figure 2.3. XBee-Pro wireless module.

The wireless radio is an XBee-Pro 802.15.4 module (see Figure 2.3) which communicates wirelessly with an identical module attached to a lap top by means of a USB interface board. The embedded radio and microprocessor are both located in a box alongside the GPS

receiver (discussed below), illustrated in Figure 2.4.

The command module (Figure 2.5) was developed by Wyle labs (www.wyle.com) and issues the commands sent from the microprocessor in the form of tones and vibrations.

The sensor choices were based on a desire to acquire accurate canine position, velocity, and orientation measurements for the control phase of the project, while also maintaining small

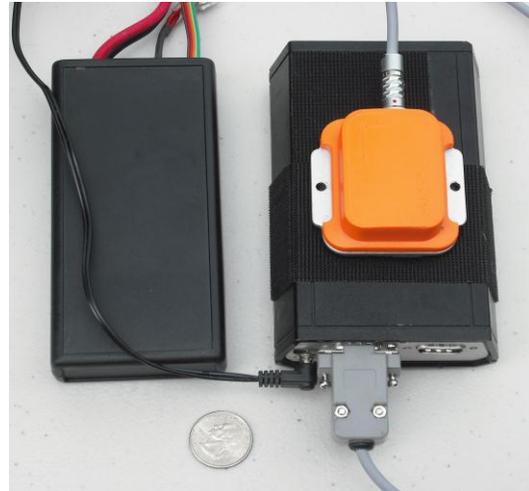


Figure 2.4. The battery for the guidance harness (left), the box containing the microprocessor, wireless radio module, and GPS receiver (right), and the inertial sensors with magnetometers (orange unit).



Figure 2.5. The command module with hand-held radio for remote control capability.

size for the canine's benefit, low cost for production practicality, and low computational expense so that the microprocessor can handle the load. The sensor suite is comprised of a Novatel GPS-702L antenna (the white disk illustrated in Figure 2.7) attached to a U-Blox RCB-4H GPS receiver (see Figure 2.6) and an XSens MTi-X (the orange unit illustrated in Figures

2.4 and 2.7), which contains a six degree of freedom inertial measurement unit (three MEMS grade accelerometers and gyroscopes) plus three magnetometers. The U-Blox GPS receiver acquires satellite data and processes it for the user and comes equipped with several settings to allow the user to choose his/her desired measurement outputs. Computational load on the microprocessor is thus lowered since the receiver does some of the necessary data processing.



Figure 2.6. U-Blox GPS receiver.



Figure 2.7. Canine Major wearing guidance harness.

The XSens unit also comes equipped with different settings for the user. It can output the raw data from each of the sensors, but it also has an Attitude Heading Reference System (AHRS) which processes and filters measurements from the sensors and outputs Euler angles, if so desired. This feature also reduces the computational load on the microprocessor. The command unit and sensor suite were attached to the canine vest, which was developed by Blackthorn K9 (www.blackthornk9.com) based on the canine team's specifications and wishes. The remote computational device is a Dell Inspiron 4150 Laptop computer.

2.4 Software

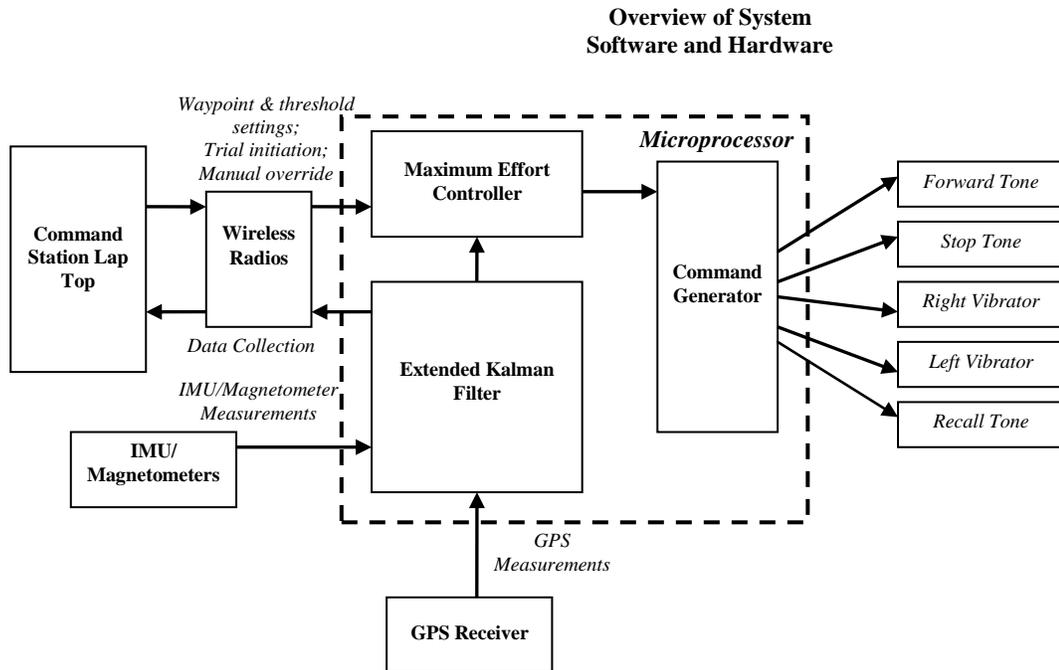


Figure 2.8. High-level overview of the hardware to software interfacing architecture.

There are two programs running simultaneously during any trial. One program runs on the microprocessor onboard the canine. This program utilizes a specialized C-based language known as Dynamic C and is responsible for interfacing all of the components onboard the canine (i.e., the wireless radio module, the GPS receiver, the IMU/magnetometers, and the command module). It also runs navigation filtering algorithms and the control algorithms discussed in Chapter 4. A C++ program runs on the lap top and serves as a data sink to store trial data. It is also used to mark waypoints before trials, initiate trials, and override the controller to allow for manual control of the canine by the handler when such is desirable. Figure 2.8 illustrates a high-level overview of the system architecture—namely, how the software and hardware interface with each other.

Chapter 3

Fuzzy Logic-Tuned Navigation Algorithms for Canine Tracking

3.1 Introduction

The use of Global Positioning System (GPS) and Inertial Navigation System (INS) integration to accurately localize vehicles and robots for remote or autonomous control purposes is very prevalent today [49-54]. However, application of this technology to living systems is much less prevalent. An INS has been used in determining step numbers, stride lengths, and heading for pedestrians [79]. GPS/INS integration has been effectively utilized in tracking the motion of pedestrians and horses [55-57, 78]. GPS/INS/barometer/magnetometer/human pedometer integration has been used for step frequency, length, and direction determination for pedestrian navigation as well [80]. INS pedestrian tracking results in GPS-denied areas have been further improved using Zero Velocity Updating and Pedestrian Dead-Reckoning (PDR) approaches [81]. A hybrid system containing both GPS/INS and GPS/PDR integrations for pedestrian navigation was used in [82]. Accelerometers have also been successfully used to analyze the motion behaviors of cats [60], as well as sow [61] and goat motion behaviors during grazing

[62], but very little work has been done in using this technology to track the position, velocity, and orientation of canines, which tend to move in more aggressive and abrupt ways.

Accelerometers have been used with wireless radios to detect canine pose for search and rescue applications [58, 59]. Although inertial sensors (i.e., accelerometers and gyroscopes) do not require GPS satellite visibility and have high frequency sampling rates (i.e., 25 Hertz for the canine system), small, low cost accelerometers are notorious for significant long-term error due to their inherent biases, scale factors, and sensitivity to noise. Also, since these canines are trained to scour an entire area for odors of interest when they reach the general destination that a handler guides them to, a certain amount of error in estimating the dog's position, velocity, and orientation may be acceptable. However, the tracking estimate errors for canine motion using inertial sensors without any correction from GPS are not insignificant, as will be discussed later in this chapter.

The use of GPS to track canines is becoming more common [63-66]. However, although GPS measurements do not contain biases, single antenna GPS has its own issues, as well. For instance, the sampling rate is relatively slow—4 Hertz for the canine system. The tracking solution is not smooth and is sensitive to surrounding environmental factors (e.g., the number of visible satellites, satellite geometry, and multi-pathing). Also, course measurements are unavailable when the antenna is not moving. Besides these factors, many of the scenarios in which the use of canines would be of interest would likely require tracking during brief GPS outages or loss in visibility of several satellites. Since GPS outages are likely in many canine tracking scenarios, it is of interest to improve the dead reckoning solution of the canine's position, velocity, and orientation

estimates for such cases. Thus, as with typical Unmanned Ground Vehicle (UGV) applications, GPS and INS can be coupled to “take the best of both worlds” and improve the tracking solution.

The Kalman Filter is an effective tool that can be used to integrate acquired measurements from a GPS sensor with acquired measurements from an Inertial Measurement Unit (IMU), as well as magnetometers. The aforementioned issues with GPS and inertial sensors when they stand alone can be eliminated or reduced by using the Kalman Filter, which helps to estimate errors that are inherent in inertial sensor measurements using the highly accurate GPS measurements. So, integrating the measurements from the different sensors with a Kalman Filter can help to achieve more accurate tracking results during GPS outages [49-53].

3.2 The Extended Kalman Filter

The Extended Kalman Filter (EKF) allows for filtering of non-linear systems, such as those found in typical navigation filtering scenarios, and is described in detail elsewhere [67, 68]. Briefly, the EKF is composed of a measurement update and a time update. When GPS measurements are available, the following standard measurement update equations are applied.

$$L_k = P_k^- C_d^T [C_d P_k^- C_d^T + R_k]^{-1} \quad (3.1)$$

$$\hat{x}_k^+ = \hat{x}_k^- + L_k (y_k - C_d \hat{x}_k^-) \quad (3.2)$$

$$P_k^+ = (I - L_k C_d) P_k^- \quad (3.3)$$

Where: P^\pm = State Estimation Covariance Matrix before or after update
 C_d = Output Observation Matrix
 R_k = GPS Sensor Noise Covariance Matrix
 \hat{x}^\pm = State estimate vector at current or next time step
 y_k = Measurement vector
 I = Identity matrix
 L = Kalman gain matrix
 k = Current time index
 d = System is discretized

The P matrix is initialized at the onset of the filter, as well as the state estimate vector.

In this dissertation, a linear measurement model is assumed. The general output or measurement equation is the following:

$$y_k = C\hat{x} + \nu \quad (3.4)$$

Where: ν = Sensor noise

The EKF time update is described by:

$$P_{k+1}^- = \Phi_c P_k^+ \Phi_c^T + Q_k \quad (3.5)$$

$$\hat{x}_{k+1}^- = \hat{x}_k^+ + (\dot{x}_{k+1} \Delta t) \quad (3.6)$$

Where: $\Phi_c =$ Continuous Jacobian Matrix

$Q_k =$ Process Noise Covariance Matrix

$(\dot{x}_{k+1}\Delta t)$ is calculated from the non-linearized state equations, using numerical integration.

The general state space equation is the following:

$$\dot{x} = \Phi x + Bu + B_\omega \omega \quad (3.7)$$

Where: $B =$ Input observation matrix

$B_\omega =$ Noise input observation matrix

$u =$ Input vector

The Jacobian (Φ) is used for the covariance prediction in the EKF to linearize the system about the operating point. The Jacobian is found utilizing the following:

$$\Phi = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \quad (3.8)$$

Where: $f_{1..n} =$ State equations

$x_{1..n} = \text{States}$

The GPS Sensor Noise Covariance Matrix (R) and the Process Noise Covariance Matrix (Q) constitute the primary EKF tuning parameters. As explained in [69-71] and in the following sections, the unique motion characteristics of live biped and quadruped species, such as canines, require specialized tuning of Q .

Figure 3.1 illustrates a high-level view of the Extended Kalman Filter. As mentioned above, the EKF consists of a time update or prediction step and a measurement update.

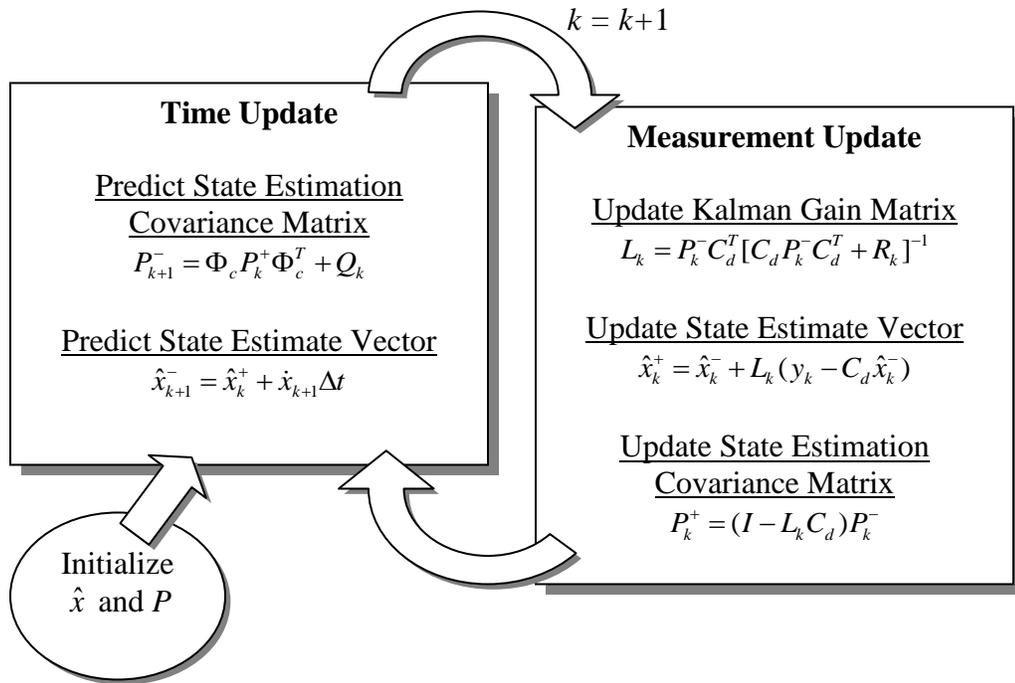


Figure 3.1. Overview of the Extended Kalman Filter.

According to [67], there are different variations of the EKF which can be used, dependent upon the application. A hybrid version of the Extended Kalman Filter was

utilized for the tracking algorithm described in this dissertation. This filter is partially discrete and partially continuous. The state time update is calculated from the continuous, non-linear equations using 4th Order Runge-Kutta numerical integration, rather than from the linearized, discretized dynamic matrix (i.e., the Jacobian). The covariance matrix time update is calculated with the linearized, continuous Jacobian, using numerical integration. Using the continuous Jacobian proves to cut down on the computational expense of the EKF. The measurement updates (i.e., the filter gain, state estimate, and covariance estimate) are calculated with the discrete algorithms from the EKF.

3.3 Adaptive Tuning of the Process Noise Covariance Matrix using Fuzzy Logic

It is important to capture an accurate estimate of the accelerometer bias and to do so quickly so that the canine tracking solution will be able to adequately hold during brief GPS outages. The accelerometer bias is used to correct accelerometer measurements, which are used in determining velocity and position solutions during GPS outages and in between GPS measurement arrivals. It has also been discussed in depth in [69] and [70] that the canine motion characteristics are such that the typical motion models that are used for automotive EKF's simply are not adequate in approximating a canine's behavior. It has been shown that using magnetometers [70] in the motion model improves the heading estimate during GPS outages. However, the ultimate EKF solution from these approaches was found to be less than ideal, especially since the EKF noise characteristics typically had to be manually tuned on a trial-by-trial basis. [See Appendices B and C for further explanation.]

It was verified in [71] that the standard EKF motion model, as used in vehicular applications, does not account for other motion characteristics that are significant in canine behavior. In that study, it was discovered that pitch plays a significant role in canine motion and accounting for it in the motion model immensely improves the canine tracking solution during brief GPS outages. However, other unmodeled canine motion characteristics are still likely present which add noise to the tracking solution, cause error to be lumped into the accelerometer bias estimate, and ultimately slow down or completely hinder the ability to accurately capture an estimate. It would be of interest to determine other canine motion characteristics that are significant enough to be added to the motion model for an improved positioning solution, but an addition of elements in the canine's motion model would add more computational burden to the already stressed microprocessor onboard the canine.

Observation of the actual canine and the EKF's tracking solutions of the canine for different trials highlights the fact that the motion characteristics of the canine change based upon its velocity. More specifically, the higher the canine's velocity, the more "noise" (i.e., unmodeled motion characteristics) there is in the accelerometer measurements. The approach presented in this section involves, in effect, adaptively adjusting the canine's motion model based on its velocity using a Fuzzy Logic Controller, thus bypassing the option of utilizing a higher-order state model as well as the need for further investigation of specific canine motion characteristics.

3.3.1 Prior Work

In 1965, Lotfi Zadeh introduced the now popular concept of fuzzy logic by publishing a paper entitled, “Fuzzy Sets” [86]. Since then, the field has grown exponentially and is now being utilized in hundreds of various applications. Scenarios in which fuzzy control theory has been used for guidance purposes have especially multiplied in the last few years. In the area of missile guidance, fuzzy logic has been used in guiding missiles in surface-to-air and air-to-surface navigation [4], including scenarios utilizing vertical launching with back turning capability [28], and tracking moving targets [21], including those moving at high speeds [1], while minimizing computational expense and miss distance and ensuring stability [27] and also controlling the impact angle [19]. Fuzzy logic has also been used for guidance in autonomous aquatic vehicle navigation [3], including autonomous underwater vehicle maneuvering [5] and depth control [24]. In the area of autonomous aerial vehicle guidance, fuzzy logic has been used in gain scheduling [13], control of planetary landing vehicles [25], satellite attitude control [7, 26], and altitude and attitude control for unmanned aerial vehicle autopiloting scenarios [30].

Perhaps the most prevalent area in which fuzzy logic is being used for guidance is in autonomous ground vehicle applications, such as navigation using magnetic markers [2], leader-follower scenarios [12], navigation in unknown environments using wall-following [17], and path planning using potential fields [15]. Fuzzy logic is being used in autonomous ground vehicle guidance for cruise control adaptation based on a lead vehicle’s velocity, stop and go maneuvering, and steering control [6, 11, 20], as well as for obstacle and target detection and pursuit or avoidance [14, 16], and in order to allow an autonomous vehicle to pass a lead vehicle [20]. It is even being used in farming

applications, where fuzzy logic is used for obstacle detection and row guidance [22], as well as automatic adjustment of harvester settings in order to minimize grain loss [29]. [9] is using fuzzy logic for object tracking in airports using video images for advanced surface movement, guidance, and control systems, and [10] is using fuzzy logic for real-time demand-responsive bus dispatching control. Fuzzy logic has been coupled with Kalman Filtering as well. Fuzzy logic and the residuals in the Kalman Filter were used in [5, 83-85] to adjust the noise covariance matrix values—in effect, re-modeling the noise characteristics of their systems in real-time. The approach presented in this chapter involves using fuzzy logic to adaptively re-model the canine’s motion characteristics based on its velocity.

3.3.2 Benefits of Fuzzy Logic Control

The development of fuzzy logic is an attempt to improve the interface between humans and computers. Computers operate in a very Boolean, “crisp,” “black and white” manner, with no “gray area.” For example, the number “two” is either greater than one or it is not greater than one. However, humans sometimes communicate in a more ambiguous or “fuzzy” manner. When a person is said to be “tall” or “short,” it is done without computing the person’s numerical height and comparing it to a specific threshold number, above which a person is considered tall and below which a person is considered short. Communication is further clouded when vague degrees are added into the equation. For example, “Tom is **very** tall. Sue is **pretty** short.” Fuzzy logic is an attempt to create a bridge in the communication between humans and computers by creating an interfacing mechanism which allows humans to use such ambiguous terminology when

communicating with a computer, while allowing the computer to still translate the language into the crisp information it requires for operation. In essence, fuzzy logic is a means to model the human brain and establish “a harmony in communication between a computer and a human being on the levels of cooperative thinking, logic, language” [87]. This approach allows operators to easily customize a fuzzy logic controller to different applications and adjust the rule base for the controller using natural human language.

One of the primary benefits that fuzzy logic provides is the ability to implement a control system effectively without the need for a precise model of the system that is to be controlled. Fuzzy logic proves to require less computational effort than other complicated non-linear control approaches and can be implemented with relative ease. Fuzzy logic controllers by-pass temporal issues (e.g., delay modeling and time-varying systems) and measurement uncertainty issues, and eliminate the issues inherent in multi-variable, multi-loop systems (e.g., “complex constraints and dependencies” [87]). They can be considered more robust than PID controllers, since they can “cover a much wider range of operating conditions and can operate with noise and disturbances of different natures” [87]. However, the fuzzy rule base must be comprehensive enough to consider such factors.

3.3.3 Fuzzy Controller Components

Figure 3.2 is a modified version of the fuzzy controller block diagram illustrated in [88]. As explained in [88], fuzzy controllers consist of four components: the fuzzy rule base, condition interface, inference engine, and action interface. These are discussed briefly below.

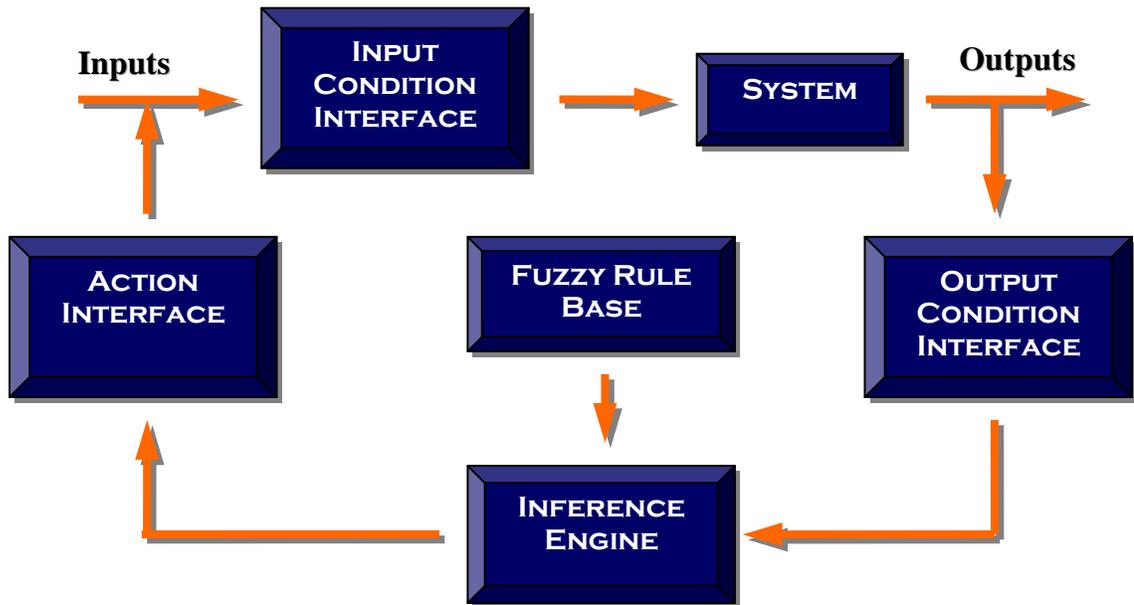


Figure 3.2. Fuzzy Logic Controller diagram.

3.3.3.1 Fuzzy Rule Base

The rule base is comprised of rules that represent the expert human knowledge about the system. The rules are typically in the form of “if-then” statements containing inputs, outputs, and linguistic variables which help to “fuzzify” the inputs. Rules are usually developed based on the knowledge and experience of experts and are fixed [88]. However, the rules can be obtained through “building operator models,” “basing them on a fuzzy model of the plant,” and “deriving them through learning” [89]. Fuzzy logic-based learning algorithms have been developed for rule adaptation using neural networks or evolutionary algorithms [90-91].

An example rule for autonomous control of vehicle speed might be the following: “If **TRAFFIC** is *heavy*, then **SPEED** is *slow*.” “**TRAFFIC**” and “**SPEED**” in the rule

are input and output fuzzy sets, respectively, that take on variable values. “**TRAFFIC**” is an input to the fuzzy control system, in this case based on a user input describing the condition of traffic or perhaps, visual sensors, whereas “**SPEED**” is an output from the fuzzy control system, in this case controlled by the accelerator and/or braking system. The degree to which “**TRAFFIC**” and “**SPEED**” are “*heavy*” and “*slow*” in the above rule is based on membership functions, described below.

3.3.3.2 Condition Interface/Fuzzifier

The condition interface fuzzifies crisp inputs to the system and/or crisp outputs from the system. Associated with each fuzzy set is a set of membership functions which transform these crisp values into a membership degree (i.e., they fuzzify the crisp values) or firing strength. The membership functions can be triangular, trapezoidal, Gamma, S, Logistic, Exponential-like, Gaussian, and step functions and are commonly determined based on expert knowledge [88]. Triangular and trapezoidal membership functions are common and are often preferred due to their simplicity and lack of computational expense [87].

For the above example, membership functions for the “**TRAFFIC**” input fuzzy set might include “*heavy*,” “*light*,” and “*average*,” while membership functions for the “**SPEED**” output fuzzy set might include, “*zero*,” “*slow*,” “*medium*,” and “*fast*.” Based on the actual value of “**TRAFFIC**” and the membership functions, the fuzzified input could be, for example, part “*heavy*,” part “*average*,” and not “*light*” at all. There would be a membership degree for each function which would determine how likely it is that the fuzzified input belongs in each category. As with how humans communicate, there will

be some variability in whether or not a person considers traffic to be, for instance, “heavy” or “average.” Thus, the membership degrees will reflect that the traffic is partially “average” and partially “heavy.” Continuing with the traffic example above, assume that the membership degree for “heavy” is found to be 0.8, and the membership degree for “average” is 0.2. This indicates that the traffic is 80% heavy and 20% average.

3.3.3.3 Inference Engine

The inference engine takes the fuzzified inputs (i.e., membership degrees from above) and infers fuzzified outputs by mapping the inputs to the rule base. From this process, it is determined which rules will fire (i.e., which rules will be applicable) and the appropriate output membership functions based on which rules fire. The input degrees directly correlate to the input degree for the output membership functions. If there are multiple inputs, fuzzy operators (e.g., min and max functions, complement, product, and summation) are used to determine the applicable firing strength for the combined functions.

For the traffic example, “heavy” is found to be the dominant “**TRAFFIC**” characteristic, since it has a higher degree than “average.” However, “**TRAFFIC**” is not 100% “heavy,” but rather, 80%. Thus, the rule (i.e., “If **TRAFFIC** is *heavy*, then **SPEED** is *slow*.”) fires with a strength of 0.8. This value would be inserted into the output function and the membership function for “*slow*” would be considered.

3.3.3.4 Action Interface/Defuzzifier

The action interface defuzzifies the output and thereby determines the crisp output which will be used in the control system. This can be done in different ways, including the max-min method, averaging method, root-sum-square method, and the clipped center of gravity method [88]. For the clipped center of gravity method, which is used in the current filtering system used in this work, each output membership function is clipped at its rule firing strength, as determined by the inference engine. The centroid of the resulting shape is subsequently determined and its horizontal coordinate is used as the output of the fuzzy logic controller for use by the control system.

3.3.4 Guidance System

The navigation system described in Chapter 2 is used as the guidance system in this work. North and East positions, as well as Velocity measurements were used from the GPS receiver. The longitudinal accelerometer measurement, as well as the pitch and heading estimates from the XSens unit are used as inputs to the filtering algorithms.

3.3.5 Tracking Algorithms

3.3.5.1 Standard EKF Components

Accurate estimates of canine position, velocity, and orientation are needed for the canine control problem. The navigation system needs to continue to provide a tracking solution during brief GPS outages, and it needs to be as computationally inexpensive as possible so that the algorithms will be able to run real-time on the small, low-cost microprocessor located onboard the canine alongside the control algorithms. Using an EKF in this dissertation has proved to be effective in meeting the first goal (i.e., yielding

a navigation solution that will hold during brief GPS outages; see [69-71]) and is used during this phase of the project as well. In order to meet the second goal, a loosely-coupled, four state EKF is used for the tracking algorithm, and the Attitude Heading Reference System on the XSens unit is used to provide a drift-free measurement of the Euler angles. This allows for an accurate estimate of heading and pitch, regardless of the presence of GPS satellite visibility and without adding several extra estimated states to the EKF, thus significantly lowering the computational expense of the algorithms. These choices proved to meet the desired goals.

The four state estimate vector, initialized with zeros, is the following:

$$\hat{x} = [\hat{V} \quad \hat{b}_a \quad \hat{N} \quad \hat{E}]^T \quad (3.9)$$

Where: \hat{V} = Estimated velocity

\hat{b}_a = Estimated accelerometer bias

\hat{N} = Estimated northern position

\hat{E} = Estimated eastern position

Figure 3.3 visually illustrates the notations used in determining northern and eastern velocities. As is common in some vehicle navigation applications, the side slip, β , is assumed to be zero at this time. The symbol, ψ , represents canine heading.

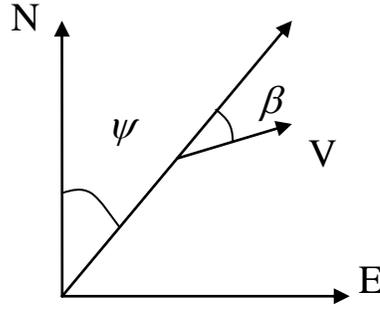


Figure 3.3. Visual illustration of northern and eastern velocity determination.

The state equation for acceleration is illustrated in Equation (3.10) and will be explained in further detail in Appendix D.

$$\hat{V} = A_x^m - g \sin(\theta_m) - \hat{b}_a \quad (3.10)$$

Where: \hat{V} = Acceleration estimate

A_x^m = Measured longitudinal acceleration from the accelerometer

\hat{b}_a = Accelerometer bias estimate

θ_m = Pitch measurement

g = Acceleration due to gravity ($9.81 \frac{m}{s^2}$)

The bias for this system is assumed to be a constant and is modeled as a random walk. The process noise is assumed to be zero mean, white noise (i.e., $\omega \sim N(0, \sigma_\omega^2)$).

Therefore, the following equation represents the change in bias for the accelerometer:

$$\hat{b}_a = 0 + \omega_a \frac{m}{s^2} \quad (3.12)$$

Where: ω_a = the process noise of the accelerometer

The estimated changes in northern and eastern directions are calculated with the following equations, according to Figure 3.3.

$$\hat{N} = \hat{V} \cos \hat{\psi} \quad (3.11)$$

$$\hat{E} = \hat{V} \sin \hat{\psi}$$

Note that the northern and eastern velocities are non-linear. Therefore, the Jacobian (Φ) is used for the covariance prediction in the EKF to linearize the system about the operating point. This results in the Jacobian matrix for the present system, shown below:

$$\Phi = \begin{bmatrix} 0 & 0 & 0 & -1 \\ \cos \hat{\psi} & 0 & 0 & 0 \\ \sin \hat{\psi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.12)$$

The longitudinal acceleration is measured by an accelerometer, and pitch and heading estimates are provided by the XSens unit. Thus, the input vector is as follows:

$$u = \begin{bmatrix} A_x^m \\ \theta_m \\ \psi_m \end{bmatrix} \quad (3.13)$$

Where: ψ_m = Heading measurement

Velocity (i.e., ground speed), northern position, and eastern position measurements are provided by GPS. Thus, the output vector, according to Equation (3.4), is as follows:

$$y = \begin{bmatrix} V_{GPSM} \\ N_{GPSM} \\ E_{GPSM} \end{bmatrix} \quad (3.14)$$

The GPS Sensor Noise Covariance Matrix (R) and the Process Noise Covariance Matrix (Q) constitute the primary EKF tuning parameters. The R matrix values are relatively standard and can be acquired from sensor information.

$$R = \begin{bmatrix} \sigma_{GPSvel}^2 & 0 & 0 \\ 0 & \sigma_{GPSnorth}^2 & 0 \\ 0 & 0 & \sigma_{GPSeast}^2 \end{bmatrix} \quad (3.15)$$

Where: $\sigma_{GPSvel} = 0.001 * S_{acc} \frac{m}{s}$

$$\sigma_{GPSnorth} = \sigma_{GPSeast} = 0.0005 * H_{acc} m$$

S_{acc} and H_{acc} are speed and position accuracy measurements, respectively, provided by the GPS receiver.

The Q matrix values are “tuned” based on canine motion characteristics through experience and are discussed in Appendices B through D, with the exception of the accelerometer noise values, which are calculated adaptively and are the subject of this chapter.

$$Q_k = \begin{bmatrix} \sigma_{Ax}^2 * T_s & 0 & 0 & 0 \\ 0 & \sigma_{bA}^2 & 0 & 0 \\ 0 & 0 & \sigma_{north}^2 & 0 \\ 0 & 0 & 0 & \sigma_{east}^2 \end{bmatrix} \quad (3.16)$$

Where: $\sigma_{bA} = 0.001 \frac{m}{s^2}$

$$\sigma_{north} = \sigma_{east} = 1 * e^{-6} m$$

The state estimation covariance matrix was initialized as follows:

$$P = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.17)$$

3.3.5.2 Fuzzy Logic Controller Components

Three prevalent fuzzy controller types are table-based controllers, Mamdani fuzzy controllers, and Takagi-Sugeno controllers [88]. A Mamdani fuzzy controller was used for this system, which involves identification of input and output linguistic variables and defining their numerical ranges (commonly based on expert knowledge), determining the desired membership functions for those variables, constructing a rule base for the control strategy, fuzzifying the input values, conducting inferencing to determine the firing strengths of activated rules, and defuzzifying to determine the action to be executed by the control system [88].

3.3.5.2.1 Fuzzy Rule Base

The fuzzy controller rule base for this phase is comprised of three rules, which are as follows.

- 1) If **VELOCITY** is *stopped*, **NOISE** is *low*. (3.18)
- 2) If **VELOCITY** is *walking/detection*, **NOISE** is *medium*.
- 3) If **VELOCITY** is *running*, **NOISE** is *high*.

As was discussed earlier, the higher the velocity of the canine, the more unmodeled motion is present in the tracking algorithms which is dumped into the accelerometer bias estimate. Therefore, the process noise on the accelerometer is adjusted accordingly using these rules.

3.3.5.2.2 Condition Interface

Figure 3.4 illustrates the membership functions that were chosen for the velocity input. The membership function that corresponds to when the canine is stopped is μ_s . The μ_w membership function corresponds to canine motion that would be considered walking or odor detection mode (i.e., motion characteristics present when the canine is searching for odors). μ_R corresponds to canine motion that would be considered running or high velocity. The shape of and values corresponding to each of the membership functions was determined through extensive observation. One of the benefits of using a fuzzy logic controller is that “designers have much freedom in selecting appropriate membership functions.... It is the task of the human expert of the domain to define the function that captures the characteristics of the fuzzy set” and to define relevant “numerical ranges” for the membership functions [88].

The equations that describe the membership functions in Figure 3.4 are as follows:

$$\mu_s(x) = \begin{cases} \frac{0.5-x}{0.5}, & 0 \leq x < 0.5 \\ 0, & \textit{otherwise} \end{cases} \quad (3.19)$$

$$\mu_w(x) = \begin{cases} \frac{x}{0.5}, & 0 \leq x < 0.5 \\ 1, & 0.5 \leq x < 3.0 \\ 4.0-x, & 3.0 \leq x < 4.0 \\ 0, & \textit{otherwise} \end{cases}$$

$$\mu_R(x) = \begin{cases} 0, & x < 3.0 \\ x-3.0, & 3.0 \leq x < 4.0 \\ 1.0, & x \geq 4.0 \end{cases}$$

The velocity estimate is inserted in the membership functions and degrees are determined for membership in each function (i.e., the crisp velocity input is fuzzified).

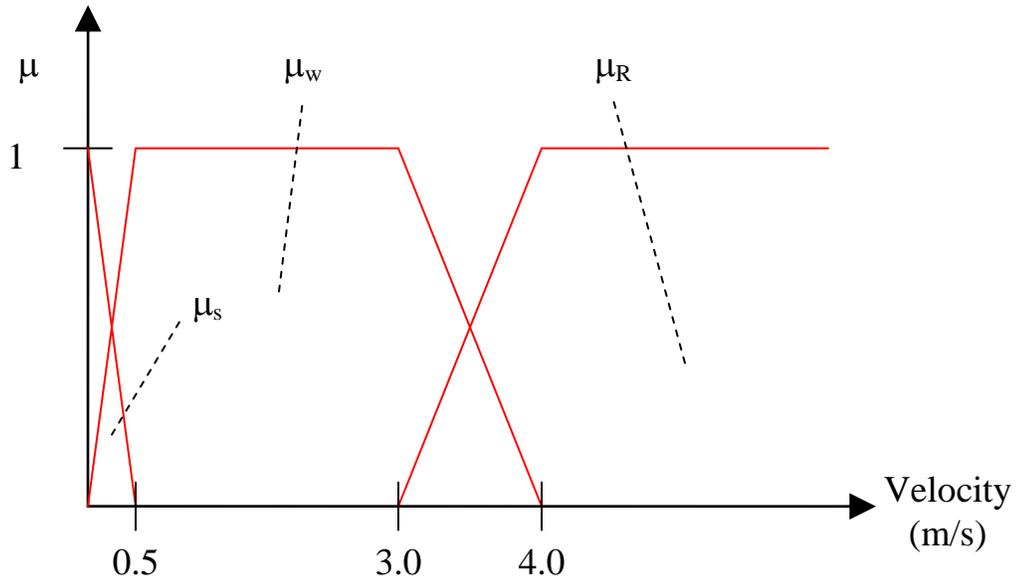


Figure 3.4. Input (Velocity) membership functions.

The walking/odor detection motion was of special interest and was studied more extensively than the other motion scenarios due to the assumption that GPS outages are more likely to occur when the canine is in this motion mode or in stopped mode. When the canine is sitting still, the position and velocity estimates are less likely to be corrupted as quickly since the canine motion in this mode contains fewer unmodeled characteristics. Thus, studying the stopped motion mode is of less interest. When the canine is running, he is likely to be in more open areas where satellite visibility is higher and GPS outages are less likely. Thus, the running mode is of less interest as well. However, odor detection is more likely to occur in areas where satellite visibility would be less. So, walking/odor detection motion was studied extensively.

Figure 3.5 illustrates the membership functions for the output, which is used in the process noise covariance matrix. The membership function that corresponds to low noise motion characteristics is μ_L . The μ_M membership function corresponds to motion characteristics that are considered medium noise, and μ_H corresponds to canine motion that is considered high in noise. As with the input membership functions, the shape of and values corresponding to each of the membership functions was determined through extensive observation.

The equations that describe the membership functions in Figure 3.5 are as follows:

$$\mu_L(x) = \begin{cases} \frac{0.2-x}{0.2}, & 0 \leq x < 0.2 \\ 0, & \textit{otherwise} \end{cases} \quad (3.20)$$

$$\mu_M(x) = \begin{cases} \frac{3-x}{2.8}, & 0.2 \leq x < 3 \\ 0, & \textit{otherwise} \end{cases}$$

$$\mu_H(x) = \begin{cases} 0, & 0 \leq x < 3 \\ x-3, & 3 \leq x < 4 \\ 1, & 4 \leq x < 300 \end{cases}$$

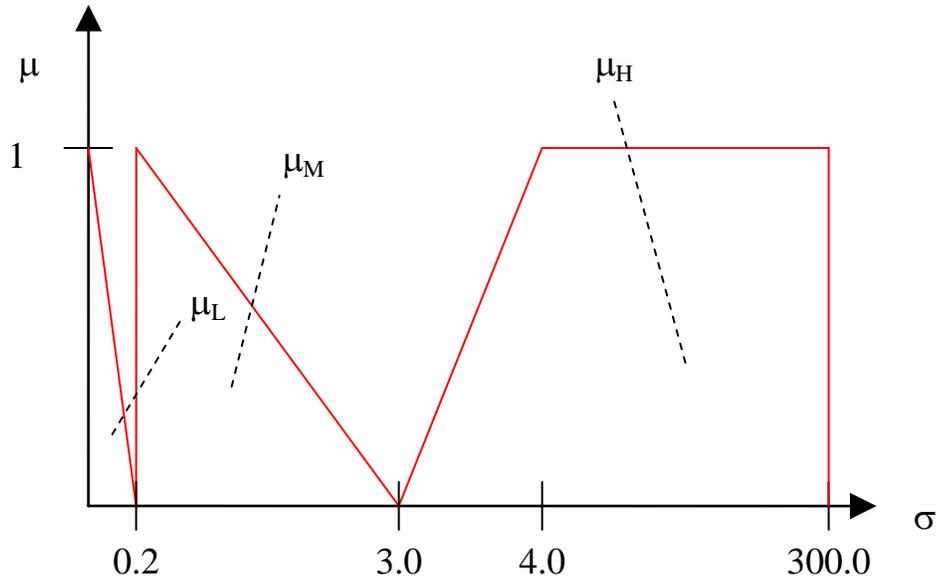


Figure 3.5. Output membership functions (not to scale).

3.3.5.2.3 Inference Engine

The inference engine receives the input membership degrees (i.e., $\mu_R(x_i)$, $\mu_w(x_i)$, and $\mu_s(x_i)$) and examines the fuzzy rules to determine which rules will fire. Membership degrees that are non-zero will fire their respective rules. After determining which rules will fire, it is determined which output membership functions will be applicable.

3.3.5.2.4 Action Interface

Defuzzification was carried out using the “clipped center of gravity” method. The applicable output membership functions, which were determined above, are clipped at their corresponding firing strengths and the centroid of the composite area is calculated. The horizontal coordinate of the centroid is the output of the controller. In discrete form, the horizontal coordinate is determined using Equation (3.21), below.

$$output = \frac{\sum_{i=1}^{n_x} x_i \mu_{composite}(x_i)}{\sum_{i=1}^{n_x} \mu_{composite}(x_i)}, x_i = (0, 0.2, 0.4, \dots, 300.0) \quad (3.21)$$

The output in this case is σ_{Ax} , the noise on the longitudinal accelerometer, which is used in the process noise covariance matrix. The greatest computational burden in the fuzzy logic controller occurs in determining the composite, clipped membership function at each time step which is used in Equation (3.21).

3.3.6 Experimental Results

Figures 3.6-3.8 illustrate the canine tracking results from an example trial using the fuzzy logic, adaptively tuned Extended Kalman Filter. Figure 3.6 is a plot of the canine's position. The trial length was about 43 seconds, and a simulated GPS outage is initiated 10 seconds before the end of the trial. The GPS position results are considered to be the "true" solution and are plotted with blue dots for the sake of comparison. The position estimate using the adaptive scheme without a GPS outage is shown with a cyan line. The position estimates from standard and adaptive EKF schemes with the GPS outage are plotted in green and red lines, respectively. By the end of the ten second outage, the adaptive position results are within two meters of the GPS solution, while the standard position results yield an error of approximately six and a half meters by the end of the GPS outage. Note that on the eastern and western legs of the trial, the GPS solution is slightly "crunched" together, rather than being spread out similar to the northern and

southern legs. This likely indicates unfavorable satellite geometry on the day of the trial and could have deleteriously affected the accelerometer bias estimate. However, the adaptive EKF solution was robust enough to maintain an accurate estimate of the position during the GPS outage.

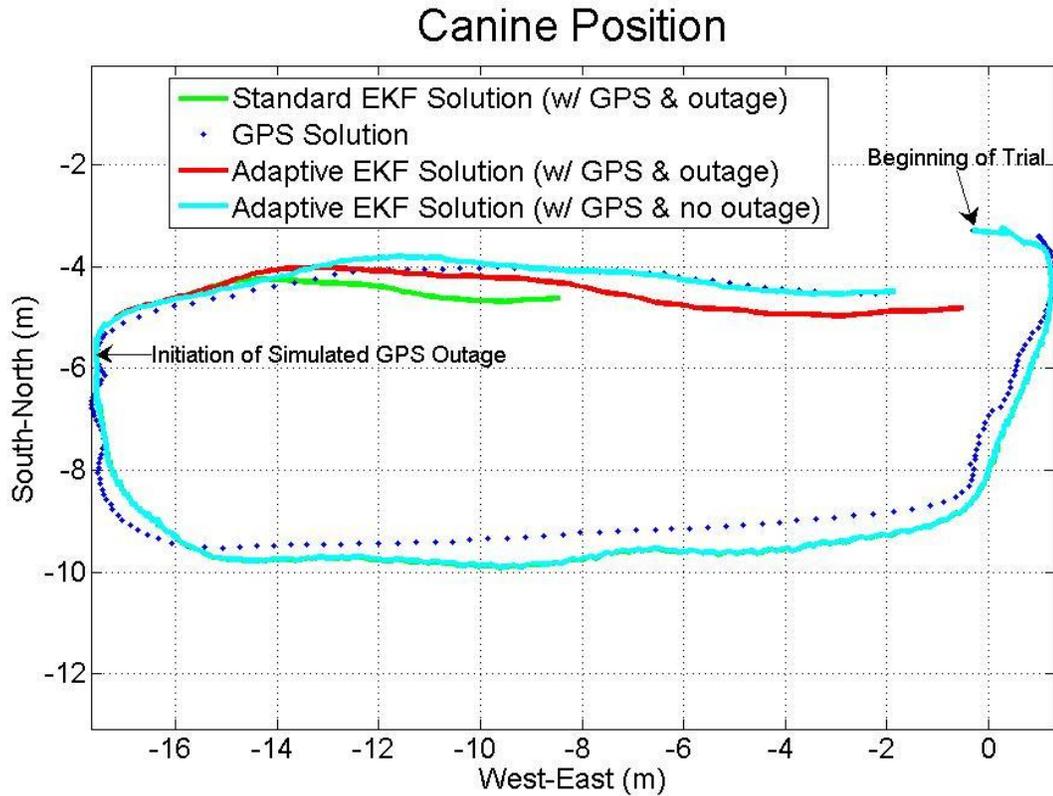


Figure 3.6. Fuzzy logic adaptively tuned EKF versus standard EKF and GPS only position estimates. A simulated GPS outage occurs during the last 10 seconds of the trial and is illustrated by the red and green lines.

Difference Between GPS and EKF Position Solutions

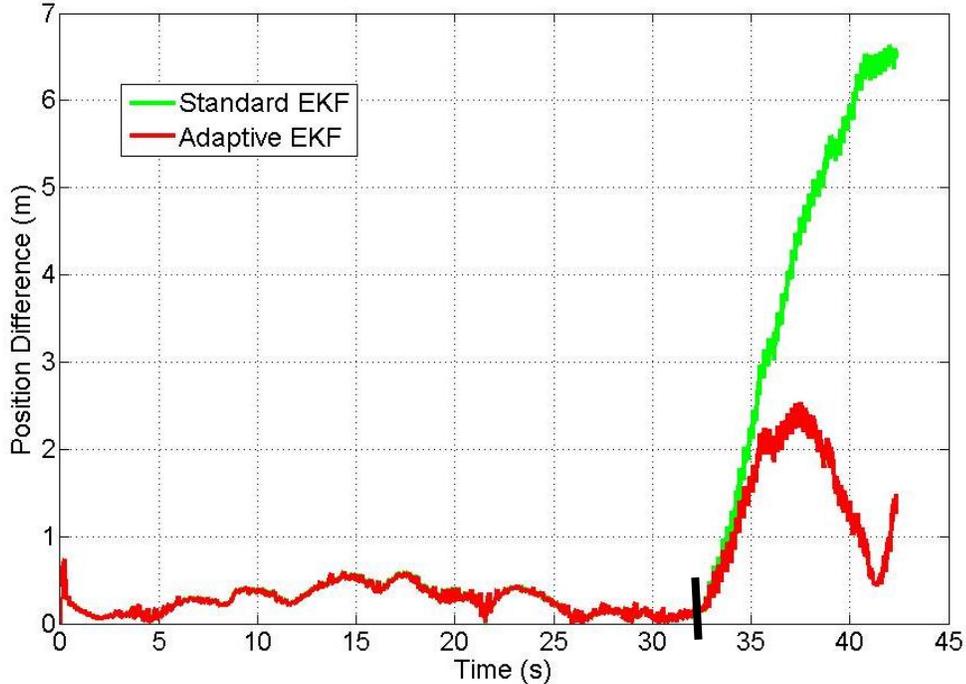


Figure 3.7. The differences between the fuzzy logic adaptively tuned EKF and GPS position measurements, as well as the difference between the standard EKF and GPS position estimates. A simulated GPS outage occurs during the last 10 seconds of the trial. Commencement of the simulated GPS outage is illustrated by the black line.

Figure 3.7 illustrates the difference between the position solutions using the standard/adaptive EKF schemes and GPS position solution throughout the trial. As discussed above, the approaches deviate more from the GPS position solution between approximately 5 through 26 seconds due to satellite geometry issues. After the GPS outage at approximately 33 seconds, the standard EKF scheme difference begins a steep climb through the rest of the trial, ending at about six and a half meters difference between the EKF solution and the GPS solution. However, the fuzzy logic, adaptive approach yields position results that are less than two and a half meters in difference

throughout the outage, ultimately ending with one and a half meters in difference between the EKF solution and the GPS measurements.

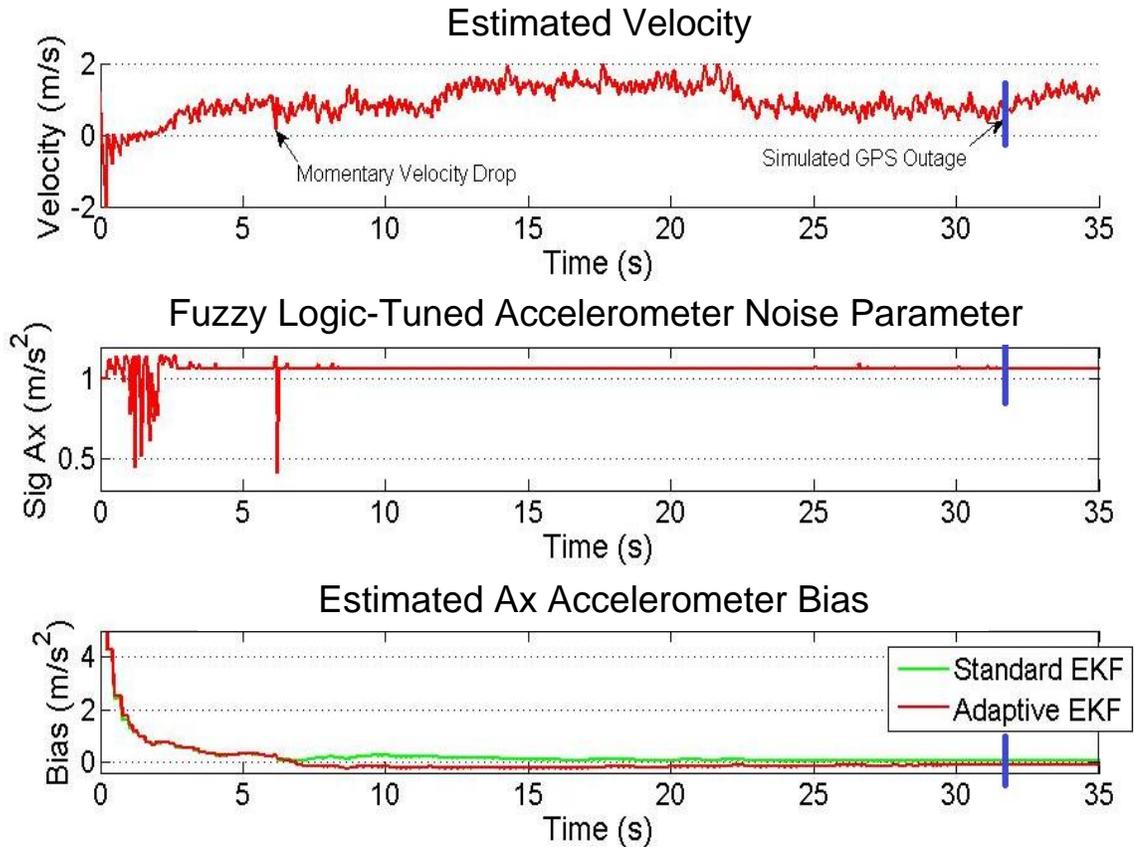


Figure 3.8. Fuzzy logic adaptively tuned velocity, adaptive and standard EKF accelerometer bias estimates, and fuzzy logic adaptively tuned noise characteristics for use in the Q matrix. A simulated GPS outage occurs during the last 10 seconds of the trial, commencing at the blue line.

Figure 3.8 illustrates the canine velocity for the same trial using the adaptive scheme. Accelerometer bias estimate results using the adaptive and standard schemes are shown, as well as the adaptively tuned noise characteristics for use in the Q matrix. An

event happens at approximately six and a half seconds which causes a notable divergence in the bias estimations using the two approaches. The event is marked by an abrupt, momentary drop in velocity. A study of the heading, accelerometer, and pitch measurements reveals that the canine likely jerked at that moment (which happens frequently), injecting motion behavior that is not accounted for in the model of the canine that is used by the standard EKF. The fuzzy logic EKF adjusted the noise characteristics in the Q matrix accordingly to account for the momentary injection of excess “noise” and continues to move towards the true bias, while the standard EKF bias estimate climbs for a couple seconds in response to the noise before re-commencing the necessary decline towards the desired bias estimate. The standard EKF approach does not recover quickly enough. By the GPS outage, the standard EKF’s estimate of the accelerometer bias is 0.07 m/s^2 , while the adaptive scheme reaches -0.09 m/s^2 , which ultimately results in a superior position solution over the standard approach during the outage, as illustrated in Figure 3.6.

Table 3.1. Average canine motion tracking error comparisons using standard and adaptive EKF approaches after a 10 second GPS outage, as well as two-tailed, unpaired t-test results. The canine is in odor detection mode.

	Position Error (m)		Velocity Error (m/s)	
	Standard	Adaptive	Standard	Adaptive
Mean	11.6	7.5	2.8	1.2
St. Dev.	7.2	5.7	1.8	0.9
2-Tailed P Value	0.0115		0.0001	
Significance	Statistically Significant		Extremely Statistically Significant	

n = 31 and 36 for constant and adaptive, respectively
www.graphpad.com/quickcalcs/ttest1.cfm?Format=SD

Table 3.1 illustrates the average motion tracking results from several trials spread out over three days for the canine using the adaptive, fuzzy logic EKF. The trials were each between 40 and 45 seconds. A simulated GPS outage occurs the last 10 seconds of each trial, thus allowing the EKF at least 30 seconds to estimate the accelerometer bias in each trial. The results from [71], where the process noise covariance matrix remained constant throughout each trial, are illustrated as well, for the sake of comparison. The navigation system using the fuzzy logic-based adaptive EKF scheme results in a 35% improvement in position error (i.e., the difference between the filtered results and GPS results) after the simulated GPS outage. The adaptive scheme resulted in a 57% improvement in velocity error. These results were found to be statistically significant using the appropriate T-tests.

Notable is the fact that the standard deviations on the position and velocity results decreased significantly as well, namely 21% and 50%, respectively. This indicates that not only is the error smaller on average for the position and velocity results with the adaptive scheme, but that even when higher errors occur in trials, the results will still likely be closer to the average than they would have been using the previous approach. Also notable is the fact that the GPS solution is only known to be accurate to within a circle with a radius of approximately 3.5 meters (based on the horizontal accuracy measurement provided by the GPS receiver). If the GPS solution is off by 3.5 meters in the opposite direction of the EKF solution, it implies that the error in position (i.e., the difference between the EKF solution and the **true** solution, rather than the GPS solution)

illustrated in Table 3.1 could actually be up to 3.5 meters less than what is exhibited in the chart.

3.4 Conclusions

The algorithms presented in this chapter present a method to locate the canine's position, velocity, and orientation for use in the autonomous control problem in a computationally inexpensive way—especially without having to investigate the potential of other unmodeled canine motion characteristics than those discussed in [71].

It is evident from the results that the use of fuzzy logic-based adaptive tuning of the process noise covariance matrix produces excellent tracking results for the canine application. The results would likely be even better if the trials were longer. If each trial were longer, the EKF would have more time to hone in on an estimate of the accelerometer bias to a higher degree of accuracy. In a real-world application, a GPS outage would not likely occur during the first minute of the navigation system's initiation. Therefore, longer trials would be a reasonable adjustment. However, in order to be able to compare the current scheme with the schemes presented in the appendices, the trial lengths were kept approximately the same.

Suggested further work would include a more in depth analysis of canine running motion and subsequent improved tuning of the running and high noise membership functions. Also, it is always of interest to improve the GPS solution, since in the EKF it is assumed that GPS is “true,” and biases are estimated based on GPS measurements. It was determined while developing the hardware packaging for this project that the radio transmission frequency from the Xbee affected how well the GPS antenna was able to

read data being transmitted from satellites in orbit. This issue affected how many satellites the GPS receiver was able to lock onto. In the end, the Novatel antenna was chosen due to its ability to read satellite data well in spite of the radio transmission issues. However, a different radio choice may improve the GPS solution even more.

The use of differential GPS algorithms could also significantly improve the GPS solution [92], as well as moving to a tightly coupled GPS solution. However, such a change would also likely require switching to a more powerful microprocessor. Given such a change, it would also be of interest to investigate the effect of roll and slip in canine motion, and how they affect the EKF solution. While stopping and starting, the vest on the canine exhibits roll characteristics, and while the canine is moving, the vest tends to slide slightly to one side or the other, thus resulting in roll-like behavior. While stopping from a sprint, the canine also tends to exhibit slip motion, which could inject excessive noise into the filtering algorithms.

Also, assuming a more powerful microprocessor was used, it could be beneficial to develop a Canine Attitude and Heading Reference System tuned to canine motion. This could allow for a more accurate determination of the Euler angles rather than those provided by the XSens, which has pedestrian and automotive settings but no setting which correlates well with the erratic motion characteristics of a canine. It would also be beneficial to upgrade to the XSens unit (or equivalent accelerometers and gyroscopes) which uses inertial sensors capable of reading higher magnitude accelerations and rates of turn to insure that the sensors are not saturated by the canine's motion.

Chapter 4

Control Algorithm

4.1 The Maximum Effort Controller

4.1.1 Introduction

As was discussed in Chapter 1, although there has been research conducted in developing remote control systems for living creatures, little work has been done in attempting to control biological entities autonomously. M.I.T. used virtual fencing coupled with GPS for tracking and autonomously herding cows [46, 47]. However, as was also discussed in Chapter 1, the autonomous control of canines is an attractive endeavor to reach for.

After having trained a canine to be remotely controlled and developing the hardware for tracking and controlling it (see Chapter 2), as well as developing algorithms for tracking the canine's motion (see Chapter 3), the next step is to develop algorithms for autonomously guiding the canine to pre-determined waypoints using the microprocessor as the handler or brain for issuing commands. The maximum effort controller (MEC) was chosen as the decision maker for the autonomous canine control system.

4.1.2 Description

In the literature, the MEC is also known as a bang-bang controller, on-off controller, binary controller, two position controller, or a hysteretic controller [67, 93-97]. In control theory, the MEC is a non-linear controller that switches between two states (e.g., on and off) when a certain threshold or switching point has been reached (see Figure 4.1).

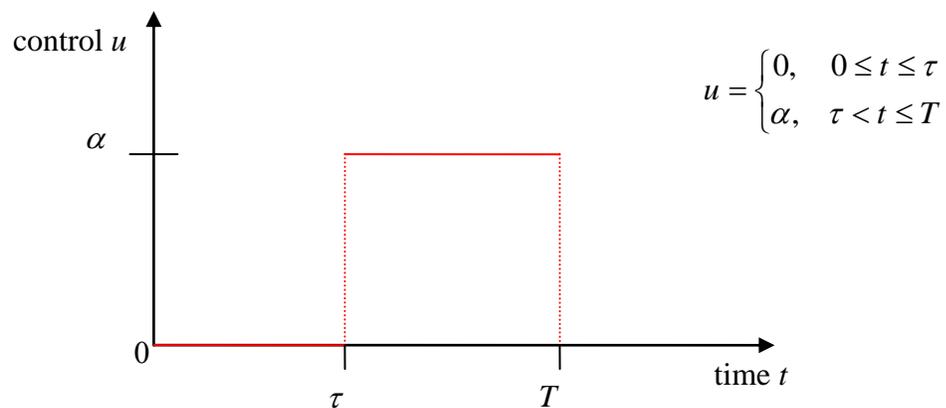


Figure 4.1. Maximum effort controller output.

A commonly cited MEC example is the thermal control system operating in a household thermostat. The oil in a furnace burns at a specific temperature. It cannot burn hotter or colder. There is no “middle ground” on how much heat is released by it. When the furnace is on, it is in a state of **maximum effort**. Systems that allow for middle ground in output, like, for instance, a pressure valve, could be controlled with a fuzzy logic controller. However, for the case of the furnace, or the command unit that was used for the autonomous canine control problem, there is simply two options—on and off. In order to effect a particular temperature in a house, the thermostat control unit “decides”

when to turn the furnace on or off based on a user-defined threshold. The furnace will burn its oil at its specific temperature until that threshold value is reached, at which time the furnace will turn off.

4.1.3 Benefits

Perhaps the most attractive benefit that the MEC offers is its relative ease in implementation. It operates without the need of an exact mathematical model for the canine motion. Thus, the MEC is useful for non-linear systems, such as canine motion. The use of the MEC for the canine application, in effect, seeks to mimic handler behavior and decisions (i.e., a handler's experience and command decisions), rather than following the complicated, non-linear mathematical equations that would describe canine motion. The canine control application lends to the use of the MEC since continuous control resolution is unattainable. The MEC is also an attractive choice for the canine control problem as it bypasses computationally expensive approximation models—a feature which is necessary for simultaneously embedding interfacing code, radio communication code, navigation and control algorithms, and potentially extensive anomaly detection and path planning code onboard the canine on the small, low-cost microprocessor.

4.1.4 Maximum Effort Controller for the Canine

4.1.4.1 Canine MEC Description

Figure 4.2 illustrates a simplified version of the MEC, as tailored for the canine application (CMEC). The flowchart shows the basic decision structure for guiding the canine to one waypoint and then back to the starting point. The controller commences by

gathering the necessary filtered data from the EKF tracking solution and calculating “*d*” and “*h*”—the **distance to the waypoint** (in meters) and the **orientation deviation** (in degrees), respectively. The **distance to the waypoint** is defined as the distance between the canine’s current position, which is constantly changing, and the waypoint position coordinates, which are stationary. Thus, the **distance to the waypoint** is constantly recalculated each time a new EKF solution is presented (i.e., at 25 Hertz). The **orientation deviation** is recalculated at the same update rate as well. The **orientation deviation** is defined as the difference between the canine’s orientation with respect to true north (i.e., “true” vs. “magnetic” north) and “*do*”—the **desired orientation** (i.e., the orientation that would be required to move straight towards the desired waypoint). Equations (4.1) through (4.4) illustrate the necessary calculations for determining the **distance to waypoint** and **orientation deviation**.

$$\begin{aligned} NoDiff &= wpts.Bn - filt.Npos \\ EaDiff &= wpts.Be - filt.Epos \end{aligned} \quad \text{(in meters)} \quad (4.1)$$

Where: *wpts.Bn* = the northern coordinate of the desired waypoint

wpts.Be = the eastern coordinate of the desired waypoint

filt.Npos = the EKF northern position solution

filt.Epos = the EKF eastern position solution

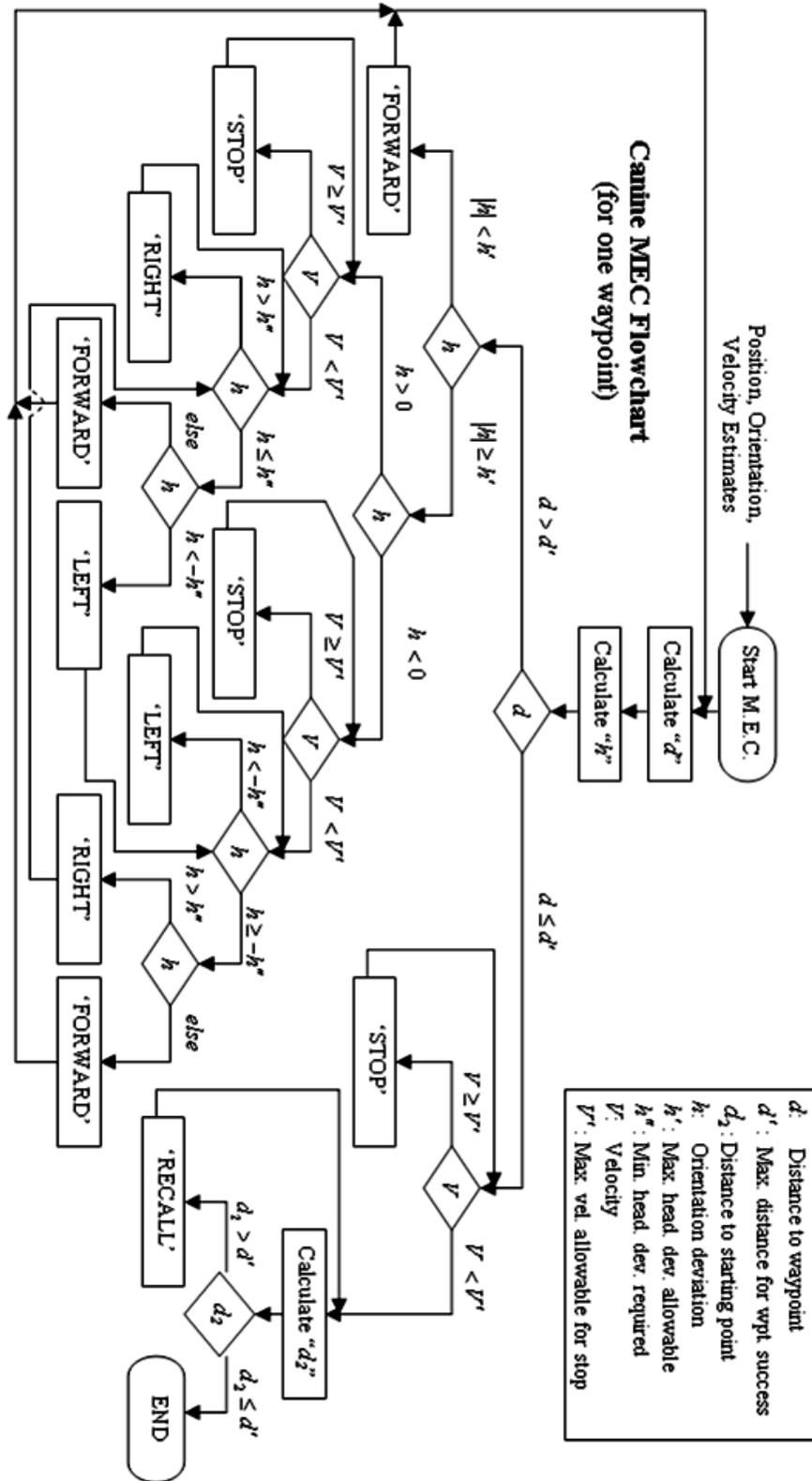


Figure 4.2. Simplified CMEC flowchart for one waypoint.

$$d = \sqrt{NoDiff^2 + EaDiff^2} \text{ (meters)} \quad (4.2)$$

$$do = a \tan 2 \left(\frac{EaDiff}{NoDiff} \right) \text{ (radians)} \quad (4.3)$$

$$h = do - filt.Head \text{ (radians)} \quad (4.4)$$

Table 4.1 illustrates the possible tonal and vibrational commands that can be issued from the microprocessor to the canine via the command module.

Table 4.1. Possible commands that can be issued to the canine.

Potential Canine Commands		
Command	Type	Description/Location
Forward	Tone	1500 Hz
Stop	Tone	3500 Hz
Recall	Tone	9500 Hz
Right	Vibration	Vest interior; Front right; Shoulder
Left	Vibration	Vest interior; Back left; Ribs

Upon determining d , if it is found to be less than or equal to d' (i.e., the maximum distance (in meters) between the canine and the waypoint allowable to be considered a waypoint success), the stop command is initiated. When the canine's velocity is under V' m/s (i.e., the maximum canine velocity allowable to be considered stopped), d_2 (defined as the distance (in meters) between the canine and the starting

point) is calculated, and the recall command is issued until the canine's position is within d' of the starting point.

If d is found to be greater than d' , the canine is not near enough to the desired waypoint, and h is determined. If the absolute value of h is under the moving **orientation deviation threshold**, h' , the canine is considered to be headed sufficiently in the correct direction and the forward command is issued or continued. If the absolute value of h is found to be greater than or equal to h' , the canine is headed sufficiently in the wrong direction to warrant a directional change. If h is above 0, the right command loop is initiated, and if h is less than 0, the left command loop is initiated.

The stop command is subsequently issued until the canine's velocity crosses below the V' threshold, as discussed above. Note that 0 is not the V' threshold due to the fact that the velocity will rarely be 0, even when the canine is not moving, due to the noisy nature of the velocity measurements. After the canine has stopped, h is checked again, just in case the canine's orientation changed during the stopping process, which is common. If h is still sufficiently high after stopping (i.e., greater than h'' radians, the **orientation deviation threshold** while stopped) to warrant a directional change, the relevant directional command is initiated. If the canine has fixed its orientation, a directional command will not be issued, but rather, the forward command will be issued. If the canine has changed direction in the opposite direction to such a degree that the opposite command should be given, the CMEC will switch commands and the appropriate directional command loop will be initiated. The CMEC is looped until the waypoint is reached, at which time, it is shut off until re-initiated.

4.1.4.2 CMEC Thresholds

Table 4.2 illustrates the thresholds that are used in the CMEC. Their meanings are described in depth in section 4.1.4.1. The values for d' , DELAYTHRESH, DELTHRESH2, DELTHRESH3, HEADCOUNTTHRESH, and MAXTURN were determined based on the wishes of the CDRI. MAXTURN serves as the trial failure criteria. If the canine reaches a waypoint having been given two or less directional changes, a waypoint success has been achieved. The directional command count is then reset, and the CMEC attempts to guide the canine to the next waypoint, if applicable. If the canine requires more than two directional changes before reaching a waypoint, the canine is recalled and the trial is considered a failure.

According to the CDRI, after being given more than two directional commands, the canine becomes more confused and begins ignoring commands, partly because there have been too many commands issued since the canine was “rewarded” by finding the bumper or C4 it is searching for. At that point, the canine is less likely to reach the waypoint in a reasonable amount of time and is more likely to start developing the bad habit of ignoring direction commands in the future. Therefore the canine is recalled, and the mission is re-attempted. Some of the reasons a canine would fail to respond properly to a command will be discussed in section 5.1.3.

The V' , h'' , and DISTTHRESH thresholds are determined intuitively based on experience. However, the h' threshold is not necessarily intuitive. Various threshold values were tested in the field. A Canine Trial Simulator (CTS) was developed to assist in selecting this parameter in the future, which will be discussed in Chapter 5.

Table 4.2. Thresholds and descriptions for the CMEC.

CMEC Thresholds			
Variable	Value	Units	Description
d'	7	m	Distance between canine and wpt., under which is considered wpt. Success
V'	0.3	m/s	Canine velocity, under which is considered a stop.
h'	49	deg.	Maximum allowable orientation deviation between the moving canine's heading and the heading required for movement in the precise direction of the wpt.
h''	20	deg.	Minimum orientation deviation that must be present after the canine stops in order for a directional command to be carried out
DELAYTHRESH	1.5	sec.	Minimum amount of time the stop command must be on when initiated
DELTHRESH2	3	sec.	Maximum amount of time a directional command will stay on before stopping and re-issuing the command
DELTHRESH3	3	sec.	Maximum amount of time the forward command will stay on before stopping and re-issuing the command
DISTTHRESH	1	m	Minimum amount of canine movement required during a forward command in order to bypass a command re-issue
HEADCOUNTTHRESH	1	sec.	Amount of time spent in the 'no command' loop when the MEC enters it
MAXTURN	2	turns	Maximum amount of issued directional commands allowable for each wpt.

4.1.4.3 Other Features of the CMEC

The CMEC also has other features which are not illustrated in Figure 4.2. First, the designed CMEC has the capability of guiding the canine to multiple waypoints. However, a visualization of the flowchart for multiple waypoints is rather large and complex. So, the chart was developed to illustrate the basic CMEC algorithm that would

be used for guiding the canine to one waypoint. Secondly, to alleviate some of the confusion that could be caused in the mind of the canine by being oversaturated with commands, the CDRI requested that the stop command, when issued, be on for a period of at least DELAYTHRESH seconds. This provided time to allow the canine to stop and catch its breath for a moment so that it would be less likely to miss the next command out of, for instance, fatigue.

Table 4.3. Summary of CMEC anomaly detection types.

CMEC Anomaly Detection Features	
Type	Description
FORWARD FAILURE	Canine fails to respond to forward command
TURN FAILURE	Canine fails to respond to directional command
DIRECTION SWITCH	Canine turns too far or in opposite direction
'NO COMMAND'	Canine changes state while in no command period
TRIAL FAILURE	Canine required more than MAXTURN directional commands before reaching wpt.

Anomaly detection schemes were also found to be important in allowing the CMEC to be able to robustly respond to atypical scenarios. Table 4.3 exhibits a summary of the anomaly detection features of the designed CMEC. The FORWARD FAIL and TURN FAIL anomalies occur if a command is issued and the canine is found to have not moved sufficiently during a certain amount of time (i.e., it has not moved at least DISTTHRESH meters by DELTHRESH3 seconds for the forward command and has not adequately adjusted its orientation in DELTHRESH2 seconds for the directional commands). If these anomalies are detected, the CMEC stops issuing the command; a

period of no command ensues (i.e., a period lasting HEADCOUNTTHRESH seconds); and then the command is re-issued. This anomaly detection feature was found to be needed for the directional commands frequently and for the forward command rarely.

Secondly, if a directional command is issued and the canine turns too far or turns in the opposite direction sufficiently enough to warrant the opposite directional command, the CMEC turns off the current directional command for a period of HEADCOUNTTHRESH seconds. The CMEC subsequently issues the opposite command. This anomaly detection feature was found to be beneficial at times as well since the CMEC would continue to issue the same turn command until the canine fixed its heading without the feature. This would result, for instance, in the canine having to turn all the way around in the opposite direction (e.g., 300 degrees) instead of turning towards the desired direction (e.g., 60 degrees).

Thirdly, when the CMEC enters one of the periods of no command that are noted above, the next command that will be issued after the dead period is logged. However, the CMEC continues to keep track of d and h during the dead period, which allows the CMEC to initiate command changes from the pre-planned, anticipated command to a new command during the dead period. This feature was found to be helpful in cases where the canine fixed its state error during the dead period (thus eliminating the need for a directional command) or in cases where the canine moved in such a way that an opposite directional command was necessary.

Finally, as discussed above, the CDRI determined that the canine should be recalled after having been issued more than MAXTURN turns between each waypoint. The designed CMEC contains the feature of keeping a count of how many directional

commands have been issued since the previous waypoint in order to meet the CDRI's wishes. After reaching a waypoint, the directional command count is reset to allow the canine an additional two directional cues before the next waypoint. This anomaly detection feature was the means by which a trial failure was determined.

4.1.5 Preliminary Pedestrian-Conducted Trial Results

4.1.5.1 Trial Setup and Description

Preliminary tests were conducted in order to eliminate potential “bugs” in the CMEC algorithms. The moving **orientation deviation threshold** was set to 30 degrees, and two types of three waypoint trials were set up. One trial type contained a waypoint geometry that included a 90 degree directional change, and one trial type required a 45 degree directional change. In order to test the CMEC algorithms, a pedestrian (i.e., me) carried the command unit and sensor suite while being guided by the control system to each waypoint. The following section illustrates sample results from these preliminary trials.

4.1.5.2 Preliminary Sample Results and Discussion

Figures 4.3-4.6 illustrate example preliminary results for the CMEC when a pedestrian is carrying the control system. In order to demonstrate the control system's robustness in handling geometrically complicated waypoints, a trial was conducted in which a 45 degree waypoint is present. Figure 4.3 illustrates the position results from that trial.

The pedestrian begins the trial at the starting point (i.e., the magenta triangle) facing east. When the control system is initiated, the forward command is given and the pedestrian begins moving in the direction he was facing. The green line illustrates the portion of the trial in which the forward command was issued. When the pedestrian has moved within the waypoint threshold, d' (i.e., the yellow line), a waypoint success has been achieved, the stop command is issued (illustrated by the red line), and the next waypoint is set.

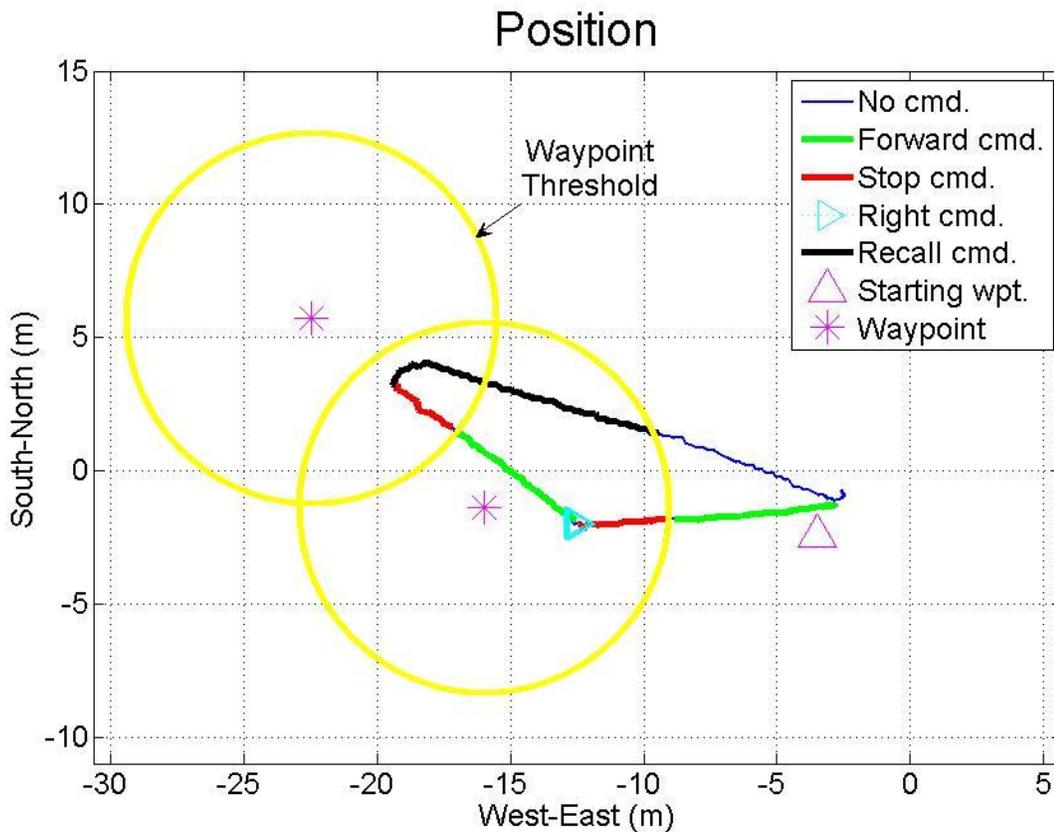


Figure 4.3. Preliminary pedestrian-conducted MEC position trial results for a 45 degree waypoint.

Note that the waypoint threshold was set based on the wishes of the CDRI. When the canine is within 7 meters of the waypoint, the canine will typically be able to detect the desired odor. After stopping, the right command is issued (illustrated by the cyan colored triangle facing to the right). When the pedestrian has adjusted his orientation sufficiently enough (i.e., when the orientation deviation is under the stopped **orientation deviation threshold**), the forward command is issued and the pedestrian moves forward again. Upon crossing the next waypoint threshold, the stop command is issued again, and after stopping, the recall command is issued until the pedestrian returns to the starting point.

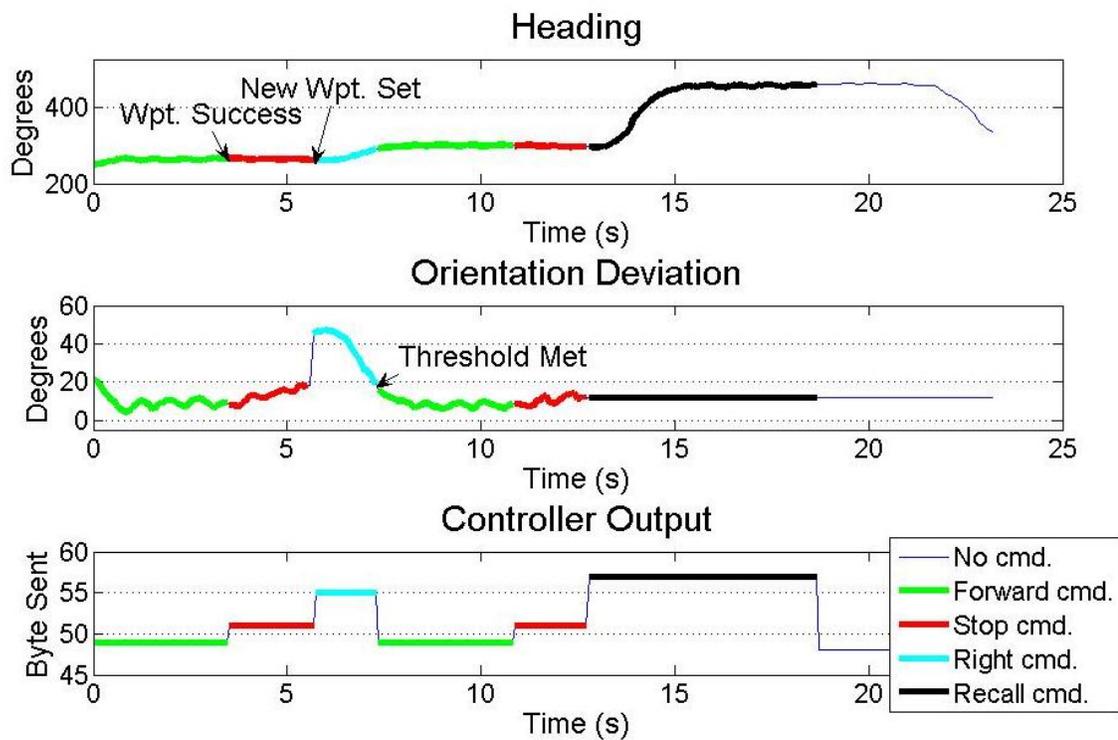


Figure 4.4. Preliminary pedestrian-conducted MEC heading, orientation deviation, and controller output trial results for a 45 degree waypoint.

Figure 4.4 shows the heading, orientation deviation, and command output results for the same trial. From the orientation deviation results, it is evident that the pedestrian begins the trial at around 20 degrees of orientation deviation (i.e., under the threshold). Therefore, the forward command is issued (see the Controller Output). As soon as the pedestrian achieves waypoint success, the stop command is issued (see the Heading subplot). After coming to a halt, the new waypoint is set, which results in an orientation deviation jump to approximately 45 degrees (at approximately six seconds), which results in the right command being initiated (see Controller Output). As soon as the orientation has been sufficiently adjusted to meet the deviation threshold, the forward command is re-issued and the trial continues.

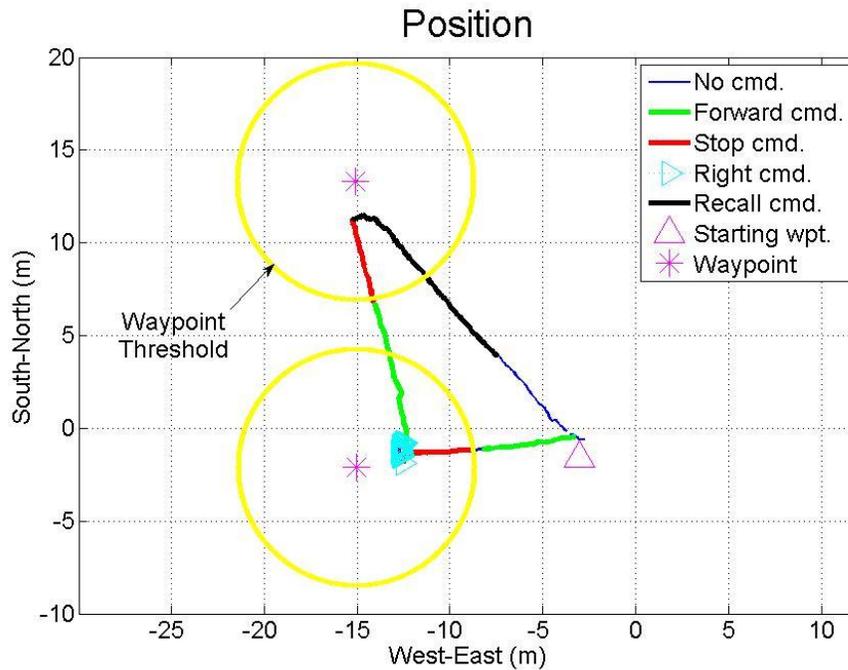


Figure 4.5. Preliminary pedestrian-conducted MEC position trial results for a 90 degree waypoint.

Figures 4.5 and 4.6 illustrate similar results, but for a 90 degree waypoint. Again, the MEC is shown to function properly. According to Figure 4.6, the orientation deviation jump is closer to 90 degrees after the first waypoint has been reached. Thus, the right command stays on longer, until the orientation deviation has dropped below the threshold, before the trial continues.

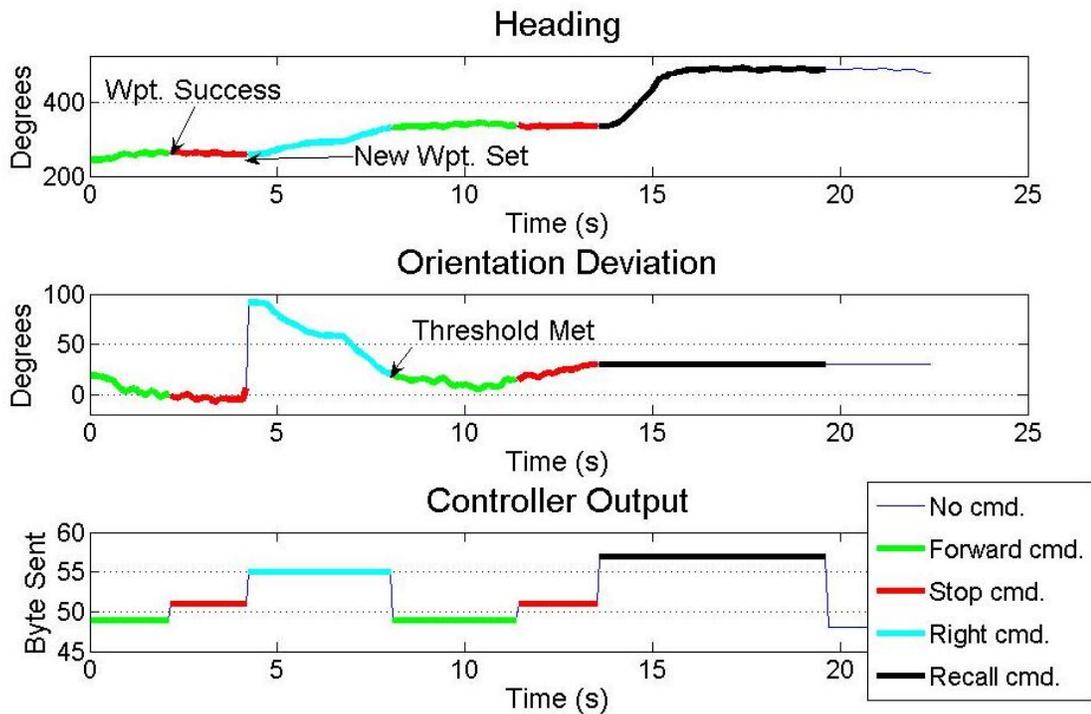


Figure 4.6. Preliminary pedestrian-conducted MEC heading, orientation deviation, and controller output trial results for a 90 degree waypoint.

4.2 Autonomous Canine Control Results and Discussion

4.2.1 Field Trial Descriptions

In order to test the effectiveness of the CMEC autonomous control system, several two and three waypoint field trials were conducted with canine Major. Some of

the trials were conducted in a structured environment, similar to what would be expected in the downtown area of a city (i.e., with buildings side-by-side, intersecting streets, sidewalks at odd angles, etc.), and some of the trials were conducted in an open field. Some of the three waypoint trials contained 90 degree waypoints, and some contained approximately 45 degree waypoints. These variables provided a broad variation for testing the robustness of the CMEC in handling different types of real-world scenarios.

The set up for the structured trials is illustrated in Figure 4.7. In order to mimic a structured, downtown environment, the CDRI chose to pick a large field with high grass and cut “streets” in the grass, while leaving the “building” areas uncut. This emulation allowed for testing of the CMEC without the added potential of GPS outages due to large structures that could occur in real urban scenarios. Structured, two waypoint trials were run from the waypoints along the bottom of the illustration to the other various waypoints in the middle and very top of the illustration in Figure 4.7.

Notice from the illustration that multiple options were available for possible field trials, including various 45 degree and 90 degree turns, so that Major would not become biased in his decision making but would rather be forced to heed commands from the control system. The terrain at each of the intersections was slightly different, visually-speaking. Straight line trials (i.e., two waypoint trials) were interspersed between trials requiring directional changes so that Major would not become too accustomed to stopping at certain intersections without a stop command. The straight line trials forced Major to run through intersections that he would often be stopped at as an intermediate waypoint and be given directional changes.

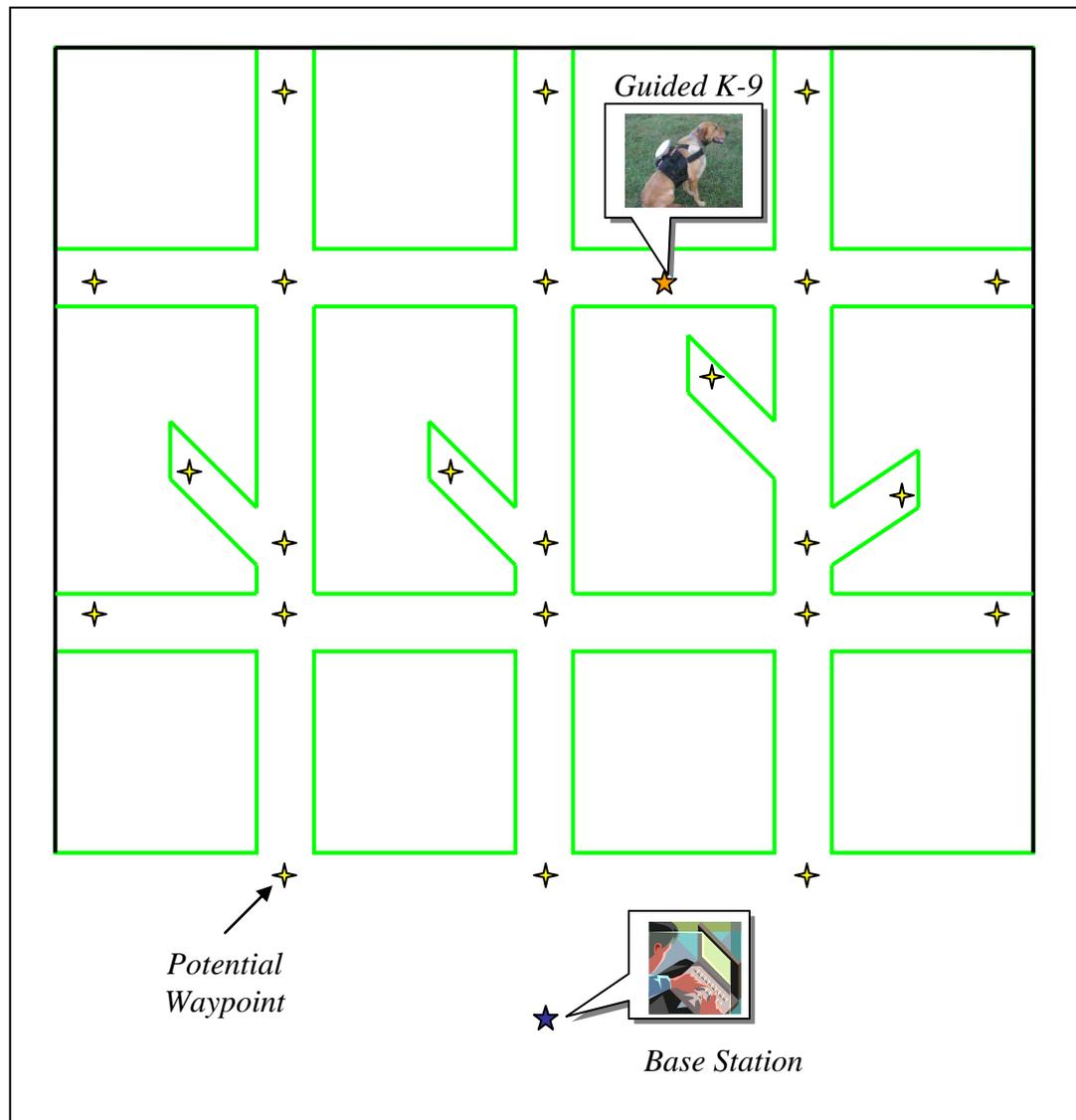


Figure 4.7. Illustration of the environment used for the structured area trials.

After discussions with the CDRI, it was determined that two waypoint trials (i.e., a starting point and a remote waypoint) would be set up so that the waypoints would be separated from each other by approximately 25 meters. This distance is sufficiently large to be able to gage whether the canine is responding appropriately to commands, but small enough to allow multiple trials in one day without excessive canine fatigue. Further, following the protocol of the CDRI, the canine handler always oriented the canine in the

general direction of the first waypoint before beginning a trial, which would be a realistic action in real-world applications and that would significantly increase the potential that the canine will reach the desired waypoint.

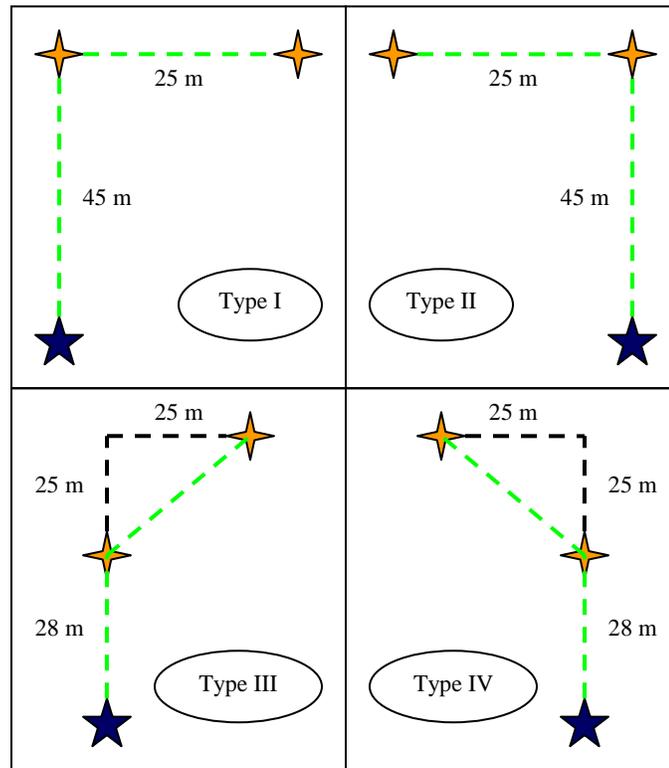


Figure 4.8. Three waypoint field trial types for non-structured environments.

In keeping with Major's training, visual cues were used to highlight potential waypoints of interest. Poles were placed at different places with bumpers lying at their base. One of the poles served as the desired waypoint, and the others served as decoys.

The unstructured three waypoint trials were also conducted in keeping with the wishes of the CDRI. Again, decoy poles were placed in the field in the midst of the

desired waypoint poles. Four different types of trials were used in order to investigate different possible waypoint geometry scenarios. Figure 4.8 illustrates the four trial types.

4.2.2 Trial Examples

4.2.2.1 Example Two Waypoint Autonomous Canine Guidance Trial

Figures 4.9-4.11 illustrate typical example results from autonomously guided canine field trials with two waypoints present. Figures 4.9 and 4.10 illustrate results from a trial in a structured environment. Other potential waypoints (i.e., decoys) are marked on the position plot (Figure 4.9) besides the desired waypoints. However, the CMEC guides the canine to the appropriate waypoint efficiently, in spite of the “decoy” waypoints, and recalls it.

Notice from the velocity plot (Figure 4.10, top) that the velocity plateaus at approximately 4 m/s and then climbs again starting at about 13 seconds. This is because the canine held back on its velocity slightly until passing the first potential waypoint, just in case a stop command was going to be given. After passing the waypoint, its velocity climbs until the stop command is given. This was common for canine Major. Notice also from the orientation deviation plot (Figure 4.10, middle), the deviation spikes to nearly 180 degrees as the canine moves past the waypoint, as should be expected since the new waypoint is not set until after the canine comes to a stop. The canine then turns around to grab the bumper and is facing towards the starting point when the next waypoint (i.e., the starting point) is set. Therefore, no significant “jump” in orientation is visible between the stop and recall commands.

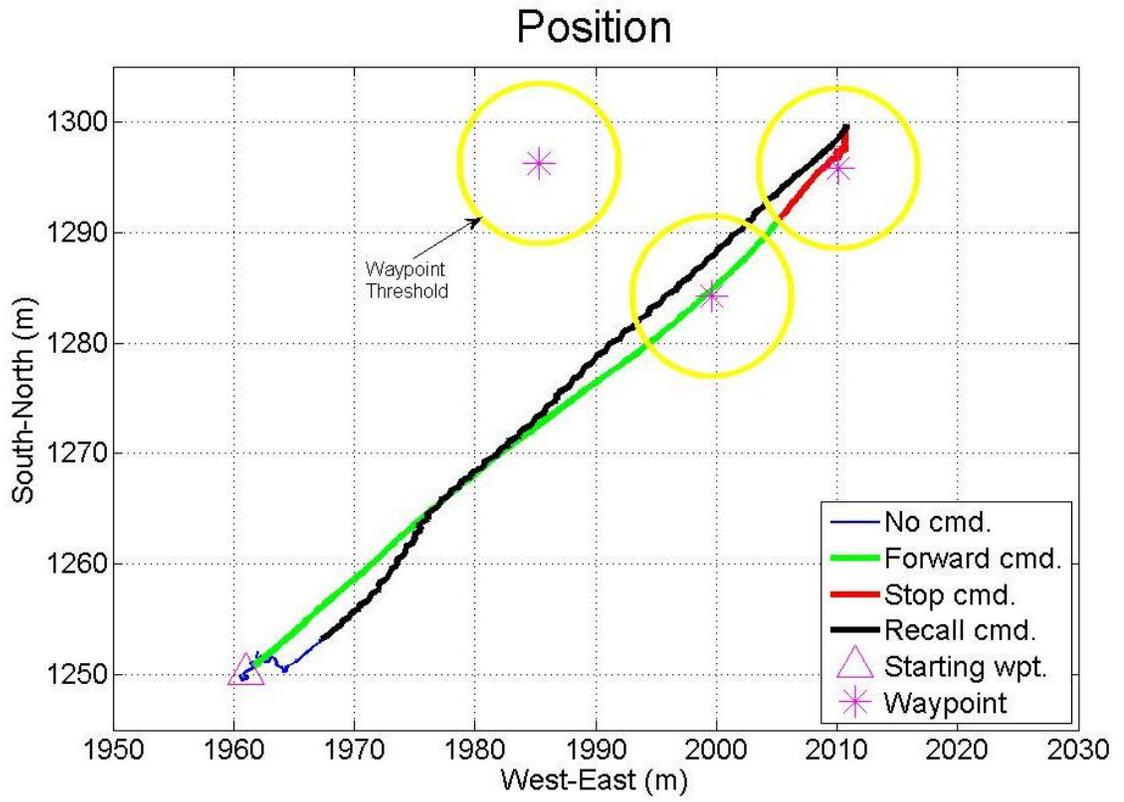


Figure 4.9. Example two waypoint trial position plot from the autonomously guided canine. The trial took place in a structured environment.

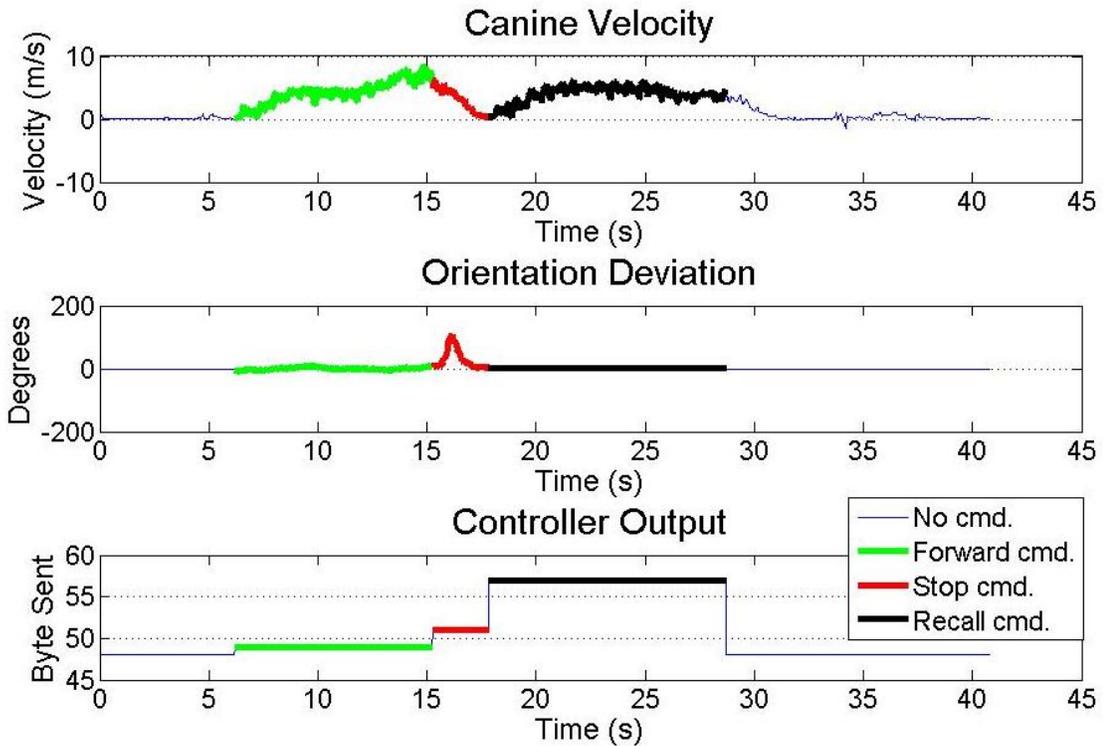


Figure 4.10. Example two waypoint trial velocity, orientation deviation, and controller output plots from the autonomously guided canine. The trial took place in a structured environment.

Figure 4.11 illustrates another example of a two waypoint field trial. The trial was conducted in a non-structured environment. Notice that the canine immediately begins to drift off course, not having locked in on the ultimate goal visually. The canine is subsequently stopped by the CMEC, and its orientation is adjusted before the trial continues with success.

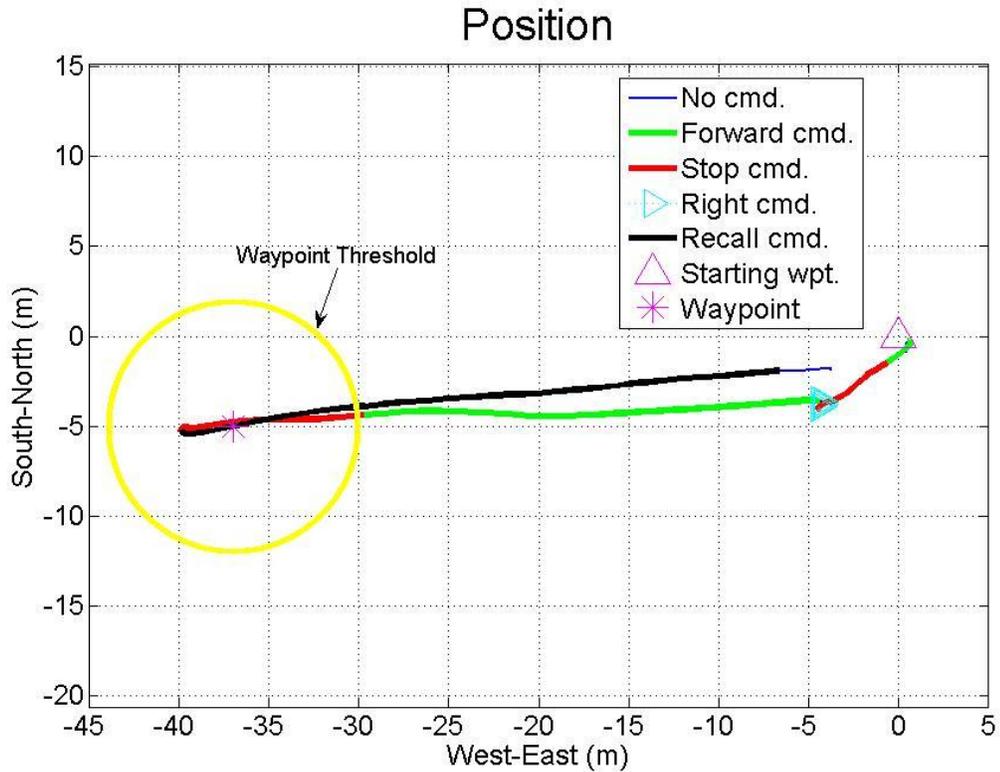


Figure 4.11. Example two waypoint trial position plot from the autonomously guided canine. The trial took place in a non-structured environment.

4.2.2.2 Example Three Waypoint Autonomous Canine Guidance Trial

Figures 4.12-14 illustrate common examples of three waypoint trials. Figures 4.12 and 4.13 illustrate the results from a trial in a structured environment and represent a scenario in which the canine would be required to move along the side of a building and then turn at the corner to reach the desired waypoint. The canine is first given the forward command and begins to head in a different direction from the desired heading. So, the stop command is issued and a directional change is made, which the canine obeys perfectly.

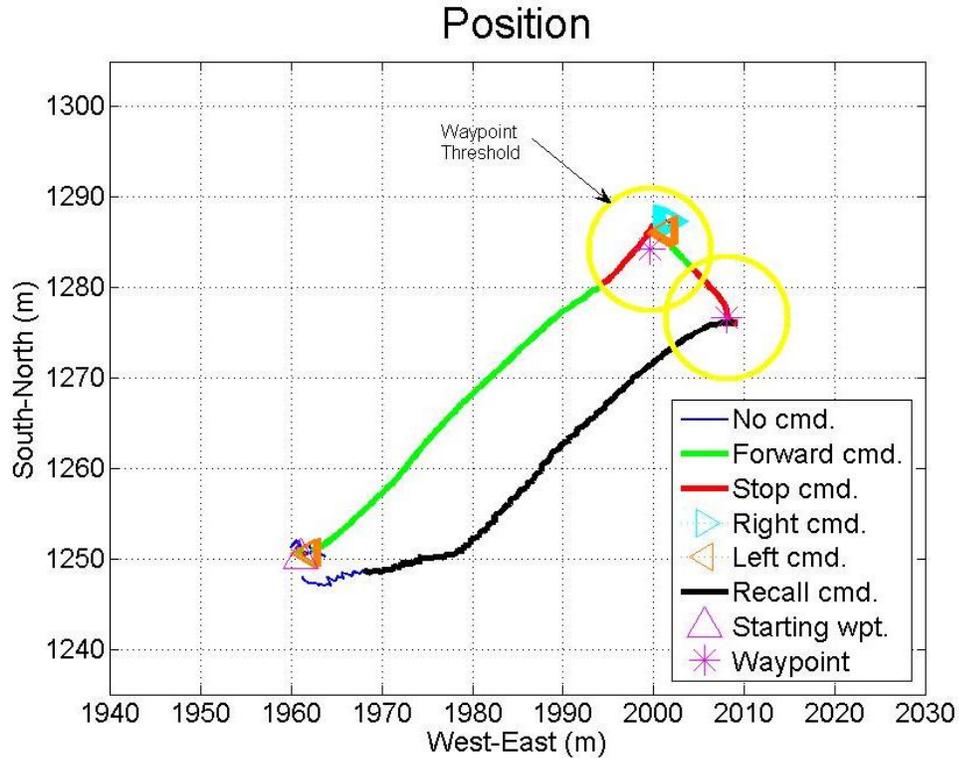


Figure 4.12. Example three waypoint trial position plot from the autonomously guided canine. The trial took place in a structured environment.

After correcting its heading, the canine is given a forward command. Notice that the canine moves beyond the first waypoint some distance before coming to a complete stop. This was a factor that sometimes proved to be an issue in the structured trials. The canine would take different lengths of time to come to a halt throughout a day of trials depending on different factors (e.g., fatigue and pre-stop command velocity). If the corridor (representing an alley, sidewalk, or road) was relatively narrow, the canine would sometimes not be sitting precisely at the intersection after coming to a stop. At such times, after receiving a directional change, the canine could become confused since it would be facing, essentially, a wall after adjusting its heading. This would not be an

issue in real world applications, as long as the corridors were wide enough. The corridors developed for these trials were relatively narrow.

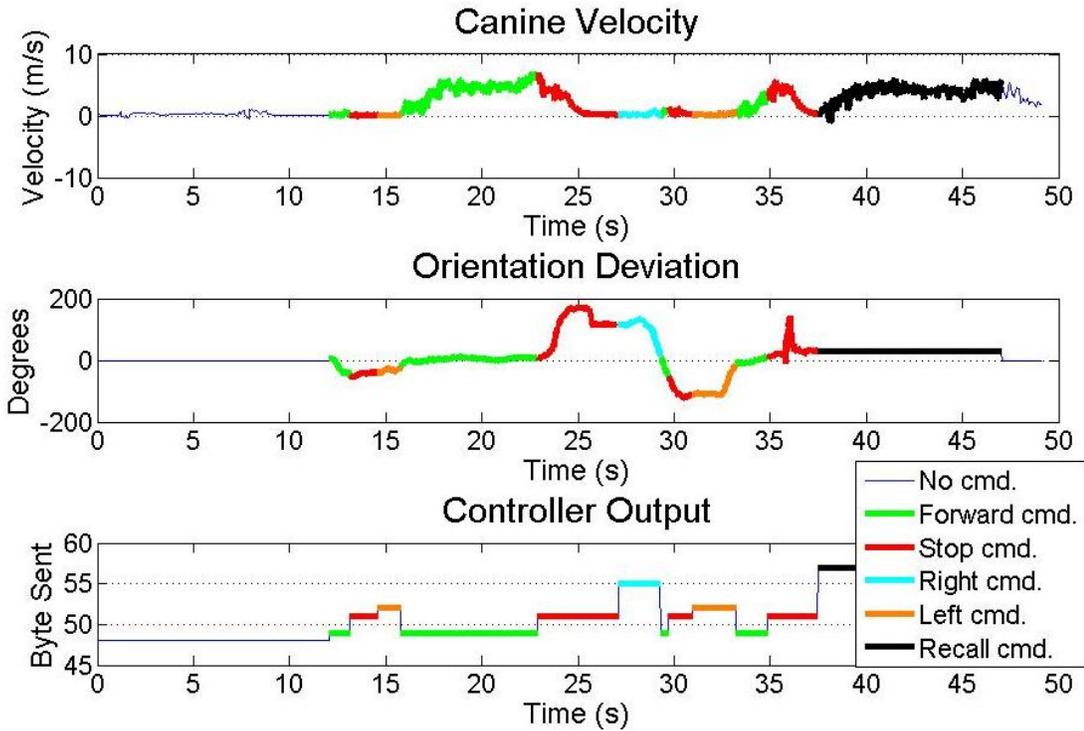


Figure 4.13. Example three waypoint trial velocity, orientation deviation, and controller output plots from the autonomously guided canine. The trial took place in a structured environment.

The accuracy of previously recorded waypoints generated from the GPS solution can cause an issue, especially when precise positioning in narrow corridors is required. This is due to the satellite geometry dependent random drift in GPS solution when sky view is limited. Since signal blockages drastically reduce the available geometry, the solution accuracy degrades. Research experience shows this issue when trying to stop the canine precisely at an intersection in narrow corridors. This complication, though

potentially significant in the long run of the project, did not prove to hinder the canine from reaching its objective in the described trials, as the example illustrated in Figures 4.12 and 4.13 shows. This issue could be resolved using GPS techniques that allow for more precise GPS solutions when setting waypoints and when tracking the canine.

Again, in spite of this issue, this trial illustrates that the CMEC is still able to guide the canine to the desired waypoint. After moving past the waypoint and “missing” the corridor, the canine is given a right directional command. The canine turns to the right as commanded and is given a forward command. Since moving forward would require that the canine run through the high grass, the canine turns and moves back down the corridor towards the starting point and is stopped again due to its orientation deviation passing the threshold. This placed the canine in precisely the right spot for a left turn command, which the canine obeys, and then the forward command is re-issued, since the canine’s orientation deviation is low enough to “suit” the CMEC. The trial then continues without incident. The canine reaches the desired waypoint, and after receiving the recall command, the canine moves back to the starting position, using the most direct path that it can find.

The prototype control system and canine training discussed in [48] allowed for right and left commands that resulted in the canine turning and moving in approximately 90 degree angles. Figure 4.14 illustrates a typical three waypoint trial with a 45 degree angle present. The trial was conducted in a non-structured environment. Notice that the CMEC is able to maneuver the canine with ease to more geometrically-complicated waypoints.

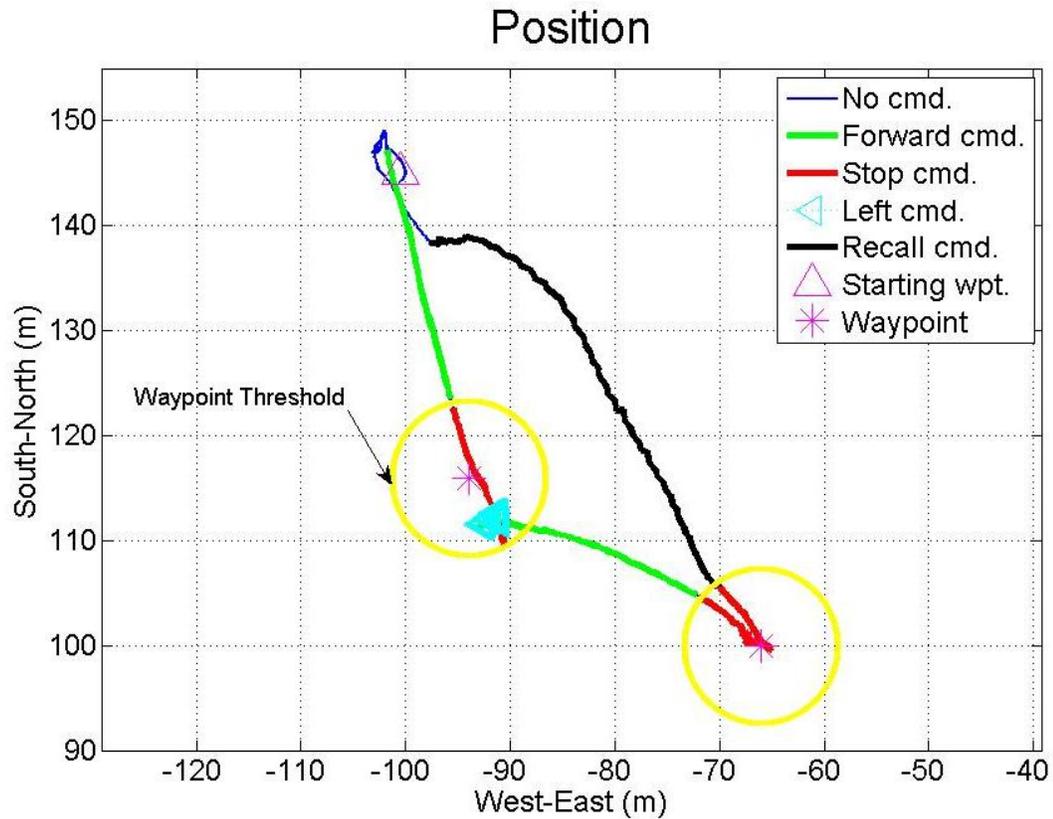


Figure 4.14. Example three waypoint trial position plot from the autonomously guided canine. The trial took place in a non-structured environment.

4.2.3 Results and Conclusion

Table 4.4 illustrates a summary of the autonomously guided canine field trial results. Various **orientation deviation threshold** values were selected for use in the CMEC. Trials in structured (i.e., “Str.”) and non-structured (i.e., “Non-Str.”) environments (see Section 4.2.1 for further descriptions), as well as two and three waypoint trials are shown. It should be noted that the CMEC presented in this dissertation worked with 100% accuracy for the trial types discussed. The navigation system, whose accuracy is highly contingent upon GPS (see section 4.2.2.2), only slightly affected the

success rates when GPS visibility was lessened. Therefore, the success rates illustrated in Table 4.4 are almost entirely based, not on the adequacy or inadequacy of the control system, but on the ability of the canine to respond correctly to the directional commands. The canine proved to respond with virtually 100% success to all tonal commands. However, vibrational command (i.e., left and right commands) adherence was significantly less and proved to be the limiting factor in trial success rates. Further discussion of this issue will take place in Chapter 5.

Table 4.4. Success rate results from the autonomously guided canine trials.

Canine Success Rate Trial Results					
h'(deg.)	2 Wpt. (Non-Str.)	2 Wpt. (Str.)	3 Wpt. (Non-Str.)	3 Wpt. (Str.)	Total
23	94.3%				
30	97.1%				
37	100.0%		77.1%		
49		100.0%		65.4%	
Total	97.7%		70.1%		86.6%
n	105	24	35	52	216

For the two waypoint trials, the success rates are very high for structured and non-structured environments, since very few directional changes are required. The success rate for the non-structured environment was around 97%. Having decoy waypoints present in the open field would sometimes move the canine off course, at which time the potential for failed directional changes increases. Also of note is the fact that the lower orientation deviation threshold settings resulted in lower success rates, as the canine

would have to be corrected more often in order to keep it under the orientation deviation threshold.

The success rate for the structured environment was 100%. Since the structured environment contains a corridor that requires that the canine move in the correct direction, one would expect the canine to have 100% accuracy, as long as the canine obeys the tonal commands (which the canine does with essentially 100% accuracy) and is in the correct corridor, which should not be a problem since the handler is with the canine at the commencement of a trial.

The composite success rate for the two waypoint trials was 97.7%. The success rate is extremely high for at least two reasons. First, two waypoint trials do not typically require a directional change, unless the canine drifts off course or is visually distracted by another potential waypoint (in non-structured environments). The canine is near perfect in responding to tonal commands (i.e., the forward, stop and recall commands), which are the primary commands needed during a two waypoint trial. Secondly, unlike with the prototype approach discussed in [48], even when a directional command is needed, the heading of the canine is still known when the canine stops or is moving very slow. This means that the appropriate directional command will be given 100% of the time, and the canine will respond correctly based on the percentages presented in the Canine Behavior Statistical Model, which will be discussed in Chapter 5.

The non-structured three waypoint trials had a success rate of 77.1%, while the structured trials had a success rate of 65.4%. Although given more trials, the non-structured trial success rate may drop to some extent, it is likely that the non-structured environment would still result in a higher success rate for three waypoint trials than the

structured environments using the same **orientation deviation threshold**. In the structured environments that were used in these trials, it was harder for the canine to see the possible end waypoint, due to visual obstructions. Since the canine used in these trials was trained to respond to visual cues, this factor could play a significant role in success rates. The composite success rate for the three waypoint trials was 70.1%. Again, the primary factor in success or failure in a three waypoint trial was ultimately the canine's success rates in correctly responding to directional commands. The overall success rate of the autonomously guided canine control system, over all of the field trials, was 86.6%.

Chapter 5

Canine Trial Simulator for CMEC Threshold Selection

5.1 Introduction

5.1.1 Canine Trial Simulator Broad Description

A Canine Trial Simulator (CTS) was developed based upon the results presented in Chapter 4 to assist future canine handlers in selecting the **orientation deviation threshold** for the CMEC. The CTS is a MATLAB-based simulator that simulates a canine's decisions and movements while being autonomously guided by a CMEC to successive waypoints. A Canine Behavior Statistical Model (CBSM) is used to help mimic the canine's behavior. The objective of the CTS is to simulate a large amount of canine guidance trials in order to determine the optimal moving **orientation deviation threshold** choice for use in the CMEC. The threshold should maximize trial success rate and/or minimize trial time. Thus, the CTS also provides a tool to assess the potential field trial performance of the canine in applications where the CMEC is being used.

5.1.2 Discussion of CTS Purpose

As was discussed in Chapter 4, the MEC is a threshold-based controller. Although some thresholds can be gathered from expert experience and from intuition, some may not be determined as easily. For the CMEC, the optimal choice of the moving **orientation deviation threshold** was not necessarily intuitive. If the threshold is chosen to be too small, the canine will be stopped and corrected too often, which will gradually saturate its mind and lead to failed responses and lead to longer trial times. If the threshold is chosen to be too large, the canine could potentially move many meters (or even kilometers, if the waypoints are spread out far enough) in the wrong direction before being given a directional command. This will lead to longer trial times, canine fatigue, and likely, added directional commands. Success rate is of the essence. However, in many real-world applications of the autonomously controlled canine, trial time could be of prime importance as well. A few seconds could be the difference between catastrophic carnage from an explosion, and the saving of lives.

Unfortunately, gathering enough live, canine field trial data to find the true optimal **orientation deviation threshold** choice would likely take years and possibly the life of the canine. With these considerations in mind, the CTS was developed based on a CBSM to determine a range of optimal choices that should lead to high success rates and a range of optimal choices that should lead to low trial times.

5.1.3 Canine Behavioral Inconsistency

Recall from the block diagram of the control system discussed in this dissertation (i.e., Figure 1.2) that disturbances are injected into the system (i.e., the canine) which will alter its output. Although many systems have disturbances which can affect their output,

the canine is a unique system. Unlike with many systems (e.g., robots, automobiles, etc.), a canine's response to a command can be, and often is, unpredictable. Due to the nature of the disturbances, the canine may or may not even respond to a command, and when it does respond, it may or may not respond in the same way each time the same command is given. The effectiveness of the canine's response is a function of many factors. The canine species and breeding factors can affect consistency and success rates, as well as the trainer and the training protocol. However, in a more specific sense, command obedience can be affected by the following:

- Did the canine see someone it knew or another canine that caused a distraction?
- Is the canine tired?
- Is the canine out of practice in using the control system?
- Is the canine hungry?
- Has the canine been trained sufficiently to respond appropriately to particular commands?
- Has the canine detected an odor while in the middle of receiving command sequences that will override the control system's commands? Odors are of high interest to a canine due to their keen sense of smell and may override the commands that are being given to it.
- Is the canine in heat or aroused? This factor played a major role in testing results at least one day.
- Has the canine been over-stimulated with commands? If too many commands are issued, the canine will stop recognizing them.

- Is the canine's breathing or heart rate high enough to cause it to not notice a directional command after stopping from a sprint?

Due to these factors and others, a simple statistical model of the canine was developed for use in the CTS. The model allows for determination of how the canine will respond to given commands and especially the likelihood that the canine will respond correctly to a given command. From said model, the CTS is utilized to determine how the model parameters will affect trial success rates and trial times in the CTS.

5.2 Canine Behavior Statistical Model

Several live canine autonomous control trials were conducted in order to determine a rough estimate of the canine's actual behavioral characteristics for use in the CTS. Table 5.1 illustrates those average canine behavioral characteristics. *Velocity* is the average velocity that the canine maintains while "cruising" (i.e., not accelerating or decelerating). The *Heading Drift* characteristic accounts for the fact that the canine tends to not run straight, but rather, gradually drifts to the right while running. *Left* and *Right Success Rate* give the likelihood that the canine will turn when given a directional command. The rest of the characteristics are self-explanatory.

A relatively high standard deviation exists for some of the characteristics due to the fact that the characteristics change throughout a day of testing based especially on the canine's rising level of fatigue. It may be of interest in the future to gather statistics on individual canine behavior throughout its months of training and how those statistics correlate to the factors discussed in section 5.1.3 and others for use in an upgraded

CBSM in the future. For the current CBSM prototype, the statistics presented here are assumed to be Gaussian in distribution. The statistics were intentionally taken from a mixture of trials from throughout the testing days, in order to capture this fluctuation and get a more precise average for the simulator.

Table 5.1. Canine behavioral statistics for simulator use.

Characteristic	Average	n
<i>Velocity (m/s)</i>	4.86 ± 1.49	66
<i>Acceleration (m/s²)</i>	1.78 ± 0.67	73
<i>Deceleration (m/s²)</i>	-2.72 ± 1.31	40
<i>Right Turn Rate (deg/s)</i>	57.92 ± 37.66	26
<i>Left Turn Rate (deg/s)</i>	-77.67 ± 41.13	26
<i>Heading Drift (deg/s)</i>	2.62 ± 1.62	35
<i>Left Success Rate</i>	0.64	45
<i>Right Success Rate</i>	0.6	35

Also of note is the fact that the forward, stop, and recall success rates were not included in the behavioral statistics since they are virtually 100%, according to the CDRI and my observations. The canine’s sense of hearing is exceptional, and it appears to be able to process auditory commands with ease. The canine’s response to directional commands from the vibration motors are the primary inconsistency in the canine’s behavior, and when directional changes are necessary (primarily in multiple waypoint scenarios), the entire control system ultimately succeeds or fails based on the canine’s likelihood in responding appropriately. The CDRI was in a constant process of improving the canine’s consistency in adhering to vibration commands throughout the period of data collection. Thus, the turn success rates were in a constant state of flux and will likely

improve more in time. However, for the sake of getting a rough estimate to be used in the CTS, the turning success rates were determined and fixed approximately half-way through the data collection period.

The primary canine motion characteristic that is not (and perhaps **cannot** be) accounted for in a simulator such as this, is the effect of visual cues on the canine's motion characteristics. If the canine sees an object of interest or if it tries to anticipate what command it will be given based on its visual surroundings, its behavior is affected, whether beneficially or deleteriously. However, even without accounting for the points discussed above, the CTS was found to be effective in producing positive results. Potential future improvements to the CBSM could include varying the CBSM characteristics based upon the handler's analysis of such factors in the environment he/she will be conducting trials in, as well as the other factors listed in section 5.1.3.

5.3 The Canine Trial Simulator

5.3.1 CTS Features

The CTS is equipped with several features that make it beneficial and attractive for use with the CMEC. First, as discussed above, the CTS utilizes actual behavioral statistics that describe the canine's motion (i.e., the CBSM). Secondly, it uses MATLAB's random number generation functions coupled with the CBSM to mimic probable canine behavior over multiple trials. For each trial, the probable heading drift while the canine is running, the average turn rate, the turn success rates, and the initial canine orientation are all estimated using random numbers and the CBSM. The CTS also accounts for canine fatigue and focus failure by using the same trial failure scheme used

in the CMEC. The CTS records the actual elapsed time for each successful trial, as well, and records the number of total commands that were issued to the canine during each successful trial. The CTS allows for simulation of the canine's behavior over a mass number of user-defined trials—numbers which would be impractical to carryout in actual field tests with the canine. The CTS also allows for an easy change of the **orientation deviation threshold** and waypoint distances/trial profiles to determine its effect in trial performance over time.

5.3.2 CTS Description

The CTS uses the CMEC as the “brain” for each Monte-Carlo-type simulated trial. The CMEC uses the Rabbit's Dynamic C programming language and was converted for use in MATLAB. After making the necessary changes to the CMEC code, the CBSM was used to determine the canine's likely response to the CMEC commands. As discussed previously, if a forward, stop, or recall command are given, the canine responds correctly with virtually 100% accuracy on the first attempt. Thus, random numbers are not needed to determine whether the canine will or will not respond to these commands. Rather, the CBSM is used to determine the canine acceleration, deceleration, and cruise velocity, and the appropriate calculus is used to determine what the canine's movement response would be to the commands issued from the CMEC code. The equations used to determine position, orientation, and velocity are described in the CMS (section D.6). The canine's initial position and velocity are set to zero, and its initial orientation is based on random numbers (see Equation (5.1)).

$$\psi_o = \overline{\psi_o} + \sigma_{\psi_o} * randn(1000,1) * \left(\frac{\pi}{180}\right) \quad (rad) \quad (5.1)$$

Where: $\psi_o(1)$ = Initial heading for a simulation

$\overline{\psi_o}$ = Average initial heading

σ_{ψ_o} = Average initial heading standard deviation

$randn(\cdot)$ = MATLAB normally-distributed random number function

The heading drift rate during forward movement is also determined using random numbers (see Equation (5.2)).

$$\delta = \overline{\delta} + \sigma_{\delta} * randn(1000,1) * \left(\frac{\pi}{180}\right) \quad \left(\frac{rad}{s}\right) \quad (5.2)$$

Where: $\delta(1)$ = Heading drift rate for the first simulation

$\overline{\delta}$ = Average heading drift rate

σ_{δ} = Average heading drift rate standard deviation

If a directional command is given from the CMEC, the canine is stopped. Then, the likelihood of the canine responding to the turn command is considered using a random number function (see Equation (5.3)). Equation (5.4) shows the MATLAB code used in the CTS for determining whether or not the simulated canine will respond to a

right directional command. A similar equation is used for the left directional command. If $Rt(i)=1$, the canine will respond to the right command at time step, i .

$$ranRT = rand(1000,1) \quad (5.3)$$

```
for i=1:1000
    if ranRT(i) <= RtSR
        Rt(i) = 1;
    else
        Rt(i) = 0;
    end
end
```

(5.4)

- Where: $ranRT$ = Random number used for right command response
 $RtSR$ = Success rate for the canine in responding to the right command
 $Rt(i)$ = Answer to the right response determination (1 = yes; 2 = no)

If the simulated canine is found to be responsive to a directional command, the turn rate is used to initiate a directional change. Equation (5.5) shows the equation used for determining the right turn rate. A similar equation is used for a left turn.

$$\dot{\psi}_R = \overline{\dot{\psi}_R} + \sigma_{\dot{\psi}_R} * randn(1000,1) * \left(\frac{\pi}{180} \right) \left(\frac{rad}{s} \right) \quad (5.5)$$

- Where: $\dot{\psi}_R(1)$ = Right turn rate for a simulated trial
 $\overline{\dot{\psi}_R}$ = Average right turn rate

$$\sigma_{\dot{\psi}_R} = \text{Average right turn rate standard deviation}$$

5.3.3 Simulated Trial Setup

The two waypoint trial simulations were set up like the field trials, as discussed in section 4.2.1. The waypoints were separated by 25 meters—namely a starting point and a point 25 meters north of the starting point. The initial heading deviation was set to 0 degrees with a standard deviation of 10 degrees.

The three waypoint trial simulations were set up like the non-structured field trials discussed in section 4.2.1. The four types of trials were developed so that the total distance traveled in a trial (i.e., the perimeter of the triangle created by the three waypoints) would equal approximately 122 meters, as set by the CDRI. Two types of trials contained a waypoint geometry that required 90 degree turns, while two types of trials contained a waypoint geometry that required 45 degree turns. Figure 4.8 illustrates the four trial types that were used for the simulations.

5.3.4 Example of a CTS Three Waypoint Trial

Figures 5.1 and 5.2 exhibit an example of a CTS-simulated three waypoint trial (Type I). As with the plots illustrated in Chapter 4, the green line indicates that the forward command is being issued during that portion of the trial. When the red line is present, the stop command is being issued. When the black line is present, the recall command is being issued, and when the blue line is present, no command is being issued. The cyan triangles indicate directional commands; the magenta-colored objects represent waypoints; and the yellow lines represent the waypoint thresholds.

In the position plot (Figure 5.1), the simulated canine is given a forward command and begins traveling approximately north. Its heading gradually drifts east in keeping with the CBSM until its orientation deviation crosses the **orientation deviation threshold**, at which time a stop command is given, and a directional change is made. When the waypoint threshold is crossed, a stop command is given until the simulated canine stops, and the new waypoint is set. The trial continues until the simulated canine crosses the starting waypoint threshold.

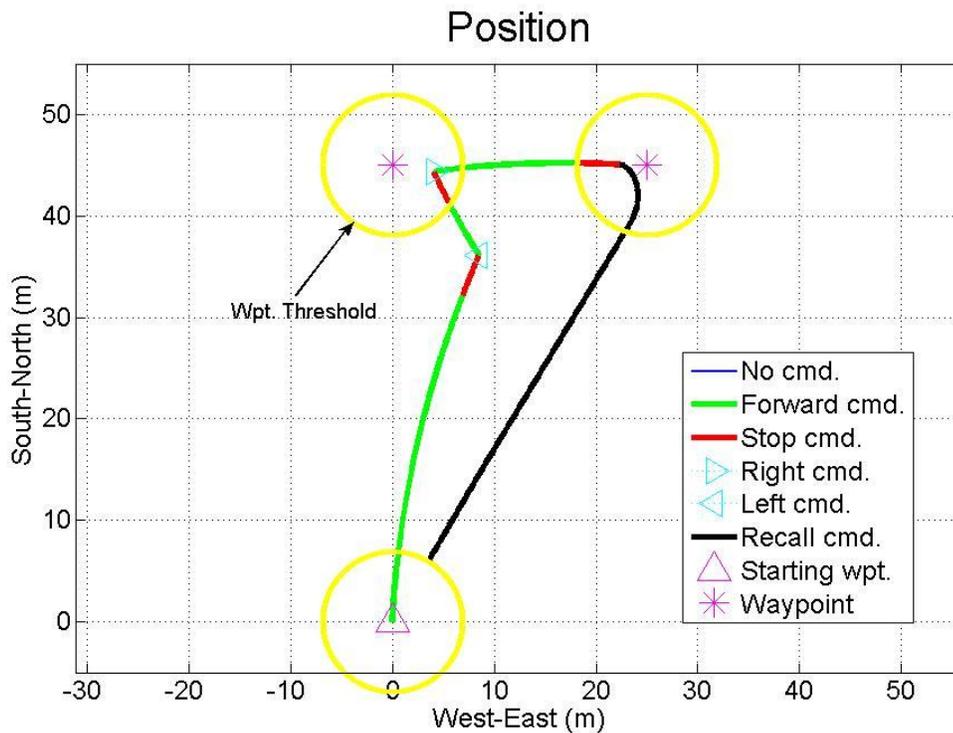


Figure 5.1. Example of a three waypoint trial CTS simulation position plot.

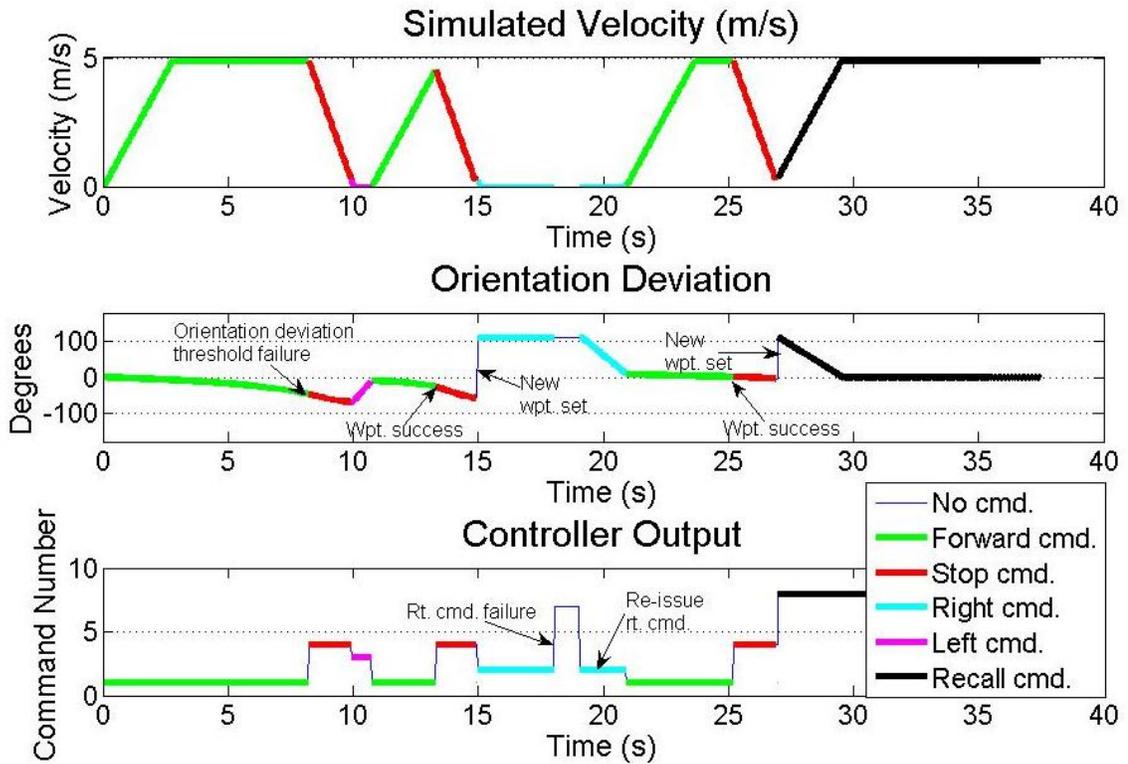


Figure 5.2. Velocity, orientation deviation, and controller output plots for an example three waypoint trial CTS simulation.

Figure 5.2 illustrates other information from the trial example illustrated in Figure 5.1—namely the simulated canine velocity throughout the trial, the orientation deviation, and the commands which are being issued at each moment of the simulated trial. Notice that the **orientation deviation threshold** is crossed at around 8 seconds, and the stop command is issued. After coming to a complete stop (i.e., after crossing the V' threshold; see the velocity portion of the plot), the left command is initiated until the h'' is crossed, at which time the forward command is initiated again until crossing the waypoint distance threshold. Notice that at approximately 15 seconds, the right command is initiated (see the Controller Output) in order to turn the simulated canine towards the next

waypoint. However, the simulated canine fails to obey the command due to the success rate from the CBSM and the random number generation functions. After the command has been given for three seconds (i.e., DELTHRESH2) with no response, no command is issued for one second (i.e., HEADCOUNTTHRESH), and then the command is re-issued. The simulated canine obeys the command on the second try, and when the orientation deviation is low enough, the forward command is issued again and the trial continues.

5.3.5 Two Waypoint CTS Results

Figure 5.3 illustrates the CTS results for various **orientation deviation thresholds** for two waypoints. The success rate and average trial time were calculated after carrying out 1000 Monte-Carlo simulations at each **orientation deviation threshold** (i.e., 15, 16, ..., and 90 degrees). A best fit trend line for the success rate data was produced, and the trend line was found to have an R^2 value of 89.1%. As is shown, the simulated canine achieves a 100% success rate when the orientation deviation is at approximately 23 degrees or above (26 degrees and above according to the trend line). This would be expected, as long as the canine is set to face in approximately the right direction from the beginning of the trial. However, when the orientation deviation is set lower, the simulated canine is forced to stop more often in order to adjust its heading to stay under the smaller threshold. This causes the trial time to rise and the success rate to decrease since more directional commands must be given to keep the canine under the orientation deviation threshold while in pursuit of the desired waypoint. Despite this drop, the success rate still remains above 85%. These results would imply that a choice of

a threshold approximately 25 degrees or above will result in excellent results. However, the three waypoint results adjust these findings and will be discussed later.

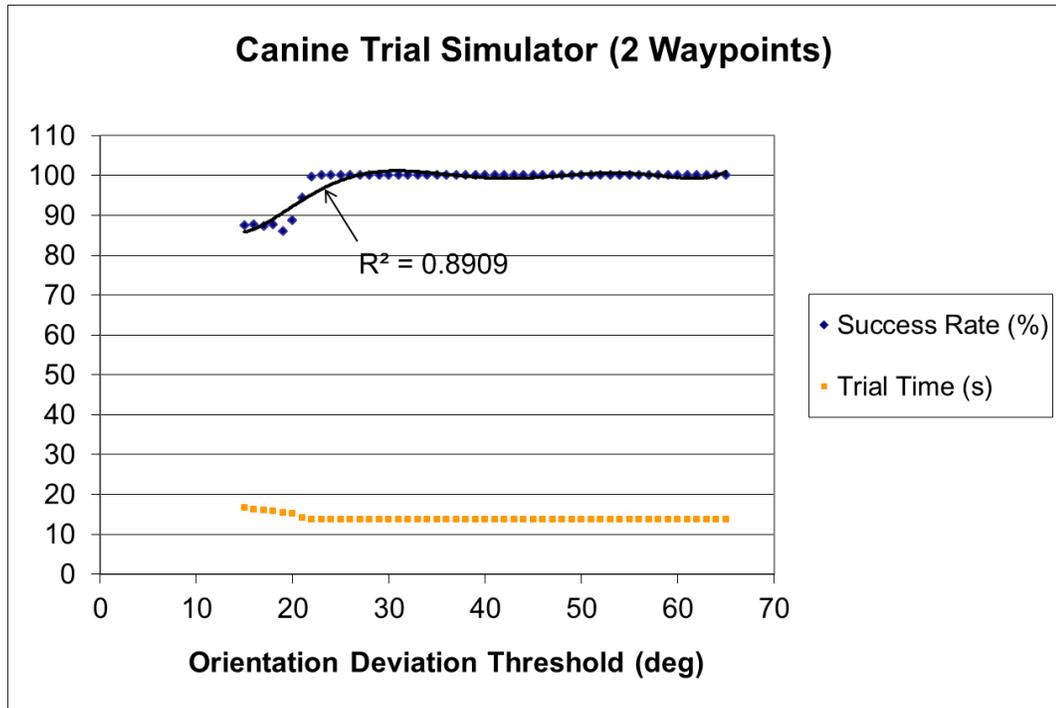


Figure 5.3. Success rate and average trial time simulation results for CTS two waypoint trials.

As was previously discussed, these simulated results are based on the actual waypoint distances that were used in canine trials. Note that if the distance to the waypoint is increased, it would be expected that there would likely be more directional commands required before reaching the waypoint. This would increase the average trial time and lower the success rate solely due to the increased distance. The diagram in Figure 5.4 helps to illustrate this concept. As “y” increases, “x” increases, which in turn increases the trial time. In the illustrated scenario, for a waypoint distance of 1000

meters, the canine could drift as far away from the waypoint to the east as approximately 200 meters without being given stop and directional change commands, assuming the **orientation deviation threshold** is about 45 degrees. This would likely result in more commands being issued to the canine before reaching the waypoint on average.

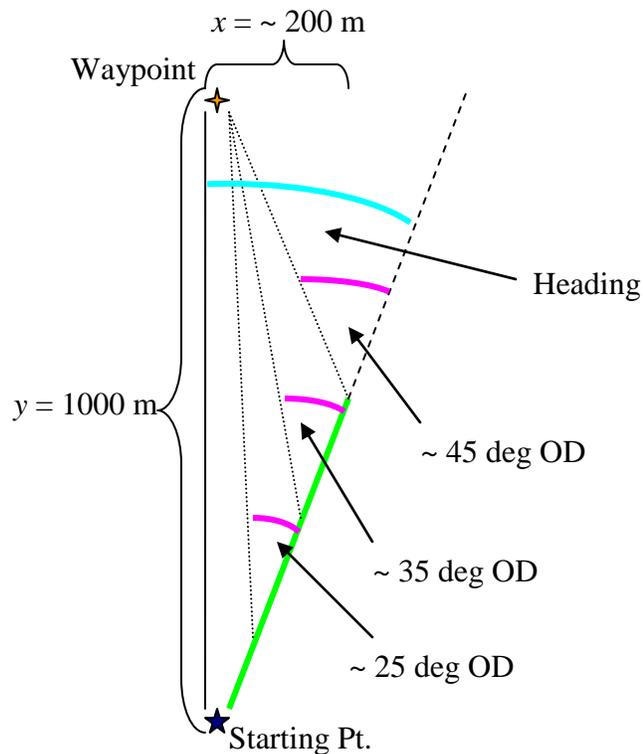


Figure 5.4. Illustration of the potential effects of larger waypoint distances.

Further, it would be expected that the average trial time would increase even more for the higher **orientation deviation thresholds**, due to the fact that the simulated canine would be “allowed” more leeway to travel several more meters off course before a directional command would be issued. Again examining Figure 5.4, if the **orientation deviation threshold** is larger, the canine would be allowed to travel further down the

dashed line before given a stop command. This fact would increase the average trial time that much more for the higher **orientation deviation thresholds**, and as was discussed earlier, every second counts in many real world scenarios. This growth in average trial time as the orientation deviation threshold is raised begins to be realized in the three waypoint simulations, which contain longer waypoint distances, and will be discussed below.

Note also that the longer a waypoint distance is, the more likely it may be that the canine will have to make an unexpected deviation from the pre-planned course momentarily, which could require a turn. In making the turn, the **orientation deviation threshold** could be passed, thus initiating unwanted correction procedures. Initial path development would account for many such scenarios, but not all possibilities could be planned for initially. Such scenarios could be addressed in future work.

5.3.6 Three Waypoint CTS Results

For the three waypoint simulations, 1000 trials were again conducted at each **orientation deviation threshold** (i.e., 15, 16, ..., 65). Of the 1000 trials, there were 250 simulations performed for each trial type (shown previously in Figure 4.8) in order to get a good mixture of possible three waypoint scenarios. Figure 5.5 illustrates the simulation results. A best fit trend line for the success rate data was produced, and the trend line was found to have an R^2 value of 95.4%. As expected, the average trial time gradually gets higher as the **orientation deviation threshold** increases (see section 5.3.5 for further explanation).

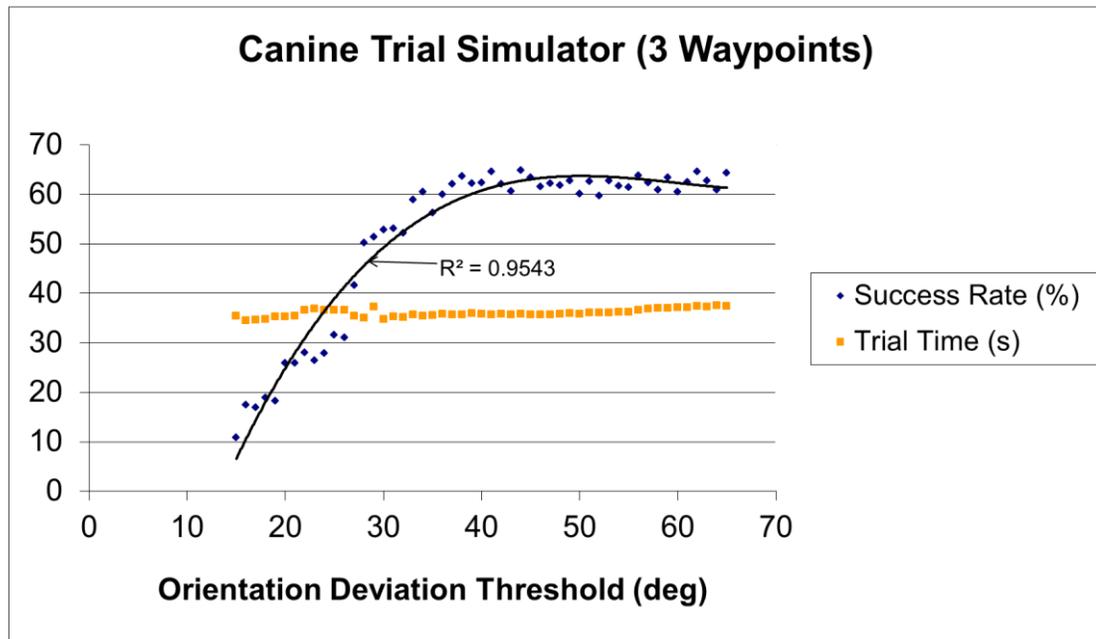


Figure 5.5. Success rate and average trial time CTS simulation results for three waypoint trials.

5.3.7 CTS Results Verification

5.3.7.1 Two Waypoint CTS Verification

The field trial results from Chapter 4 were used to check the accuracy of the CTS. In the field trials, three different **orientation deviation thresholds** were used while running two waypoint trials with an actual canine. Recall that for each of the three **orientation deviation thresholds**, 35 field trials were run, and for each **orientation deviation threshold** in simulation, 1000 trials were performed.

Figure 5.6 illustrates the success rate results for the three **orientation deviation thresholds** using the CTS versus the actual trial data. Since few orientation corrections will be necessary in a two waypoint trial, intuition would lead to the conclusion that the success rates should be very high, as long as the **orientation deviation threshold** is not

chosen to be too small. Note from Figure 5.3, once again, that according to the CTS, the success rate remains at 100% for upper **orientation deviation thresholds**, then begins to decrease for **orientation deviation thresholds** below 23 degrees (below 26 degrees, according to the trend line). The actual trial data also indicated very high success rates above 23 degrees—100% at 37 degrees and 97% at 30 degrees (i.e., one trial failure). The 23 degree threshold actual data (94%) is in keeping with the trend line (95%) of the CTS results, as shown in Figure 5.3. Considering the fact that the CTS does not account for certain effects that occur in real world canine trials (e.g., navigation system issues and visual disturbances), the CTS success rate results line up nicely with the actual data.

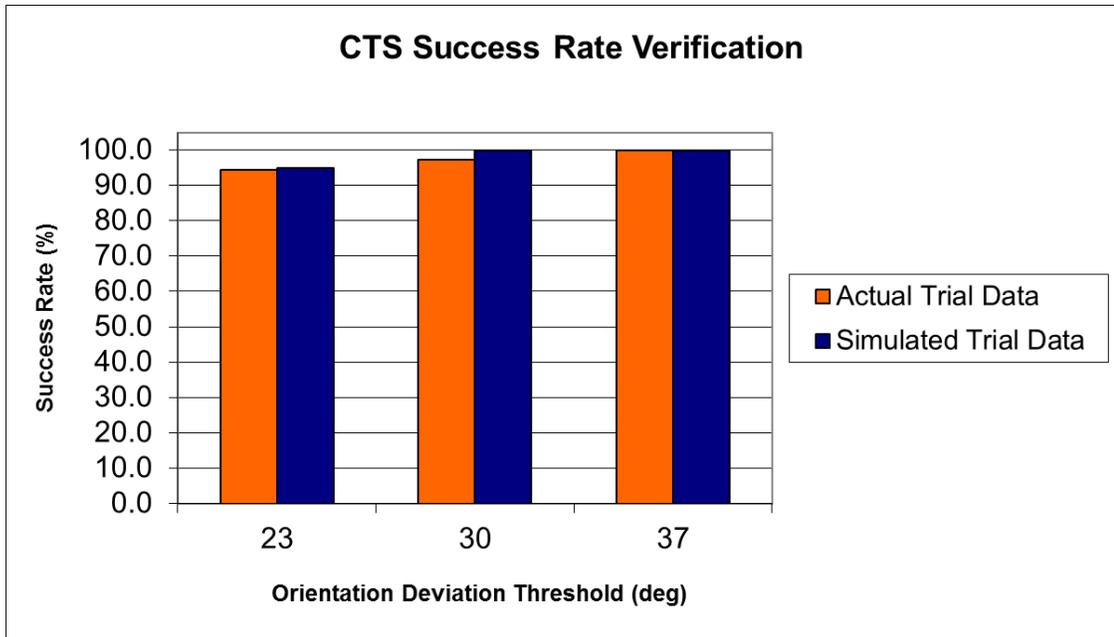


Figure 5.6. Comparison of two waypoint success rate results for the CTS versus actual trial data.

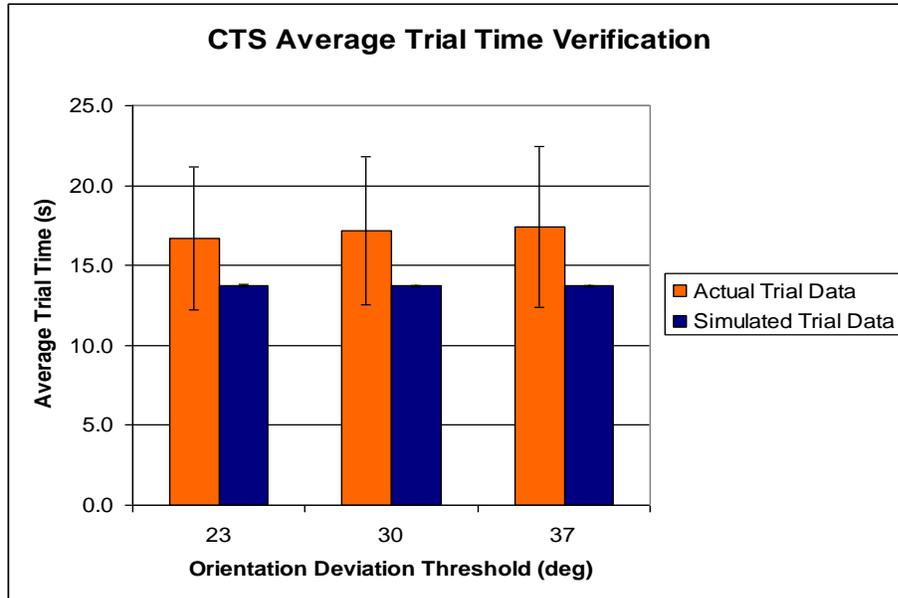


Figure 5.7. Comparison of two waypoint average trial time results for the CTS versus actual trial data.

Figure 5.7 shows the average trial time results. The actual average trial times are higher than the simulated results by about 3 to 3.5 seconds. This difference is still within the actual trial standard deviation on all three orientation deviations, and with more trial data, this difference may decrease. However, the primary reason there is a difference between the actual and simulated trial times could be that the following factors are not taken into account in the CTS. These factors could be considered in later versions of the CTS.

- In the actual canine trials the canine takes between 1 and 5 seconds to find and grab the bumper (i.e., the item the canine has been trained to find at the final destination) out of the grass before obeying the recall command.

- Unlike with the simulator, there is a small delay in actual canine response to each command. The delay could be a function of fatigue as well.
- The actual trials were timed by human hand using an egg timer and are therefore often slightly higher.

These factors considered, the CTS results, once again, line up with the actual results excellently.

5.3.7.2 Three Waypoint CTS Verification

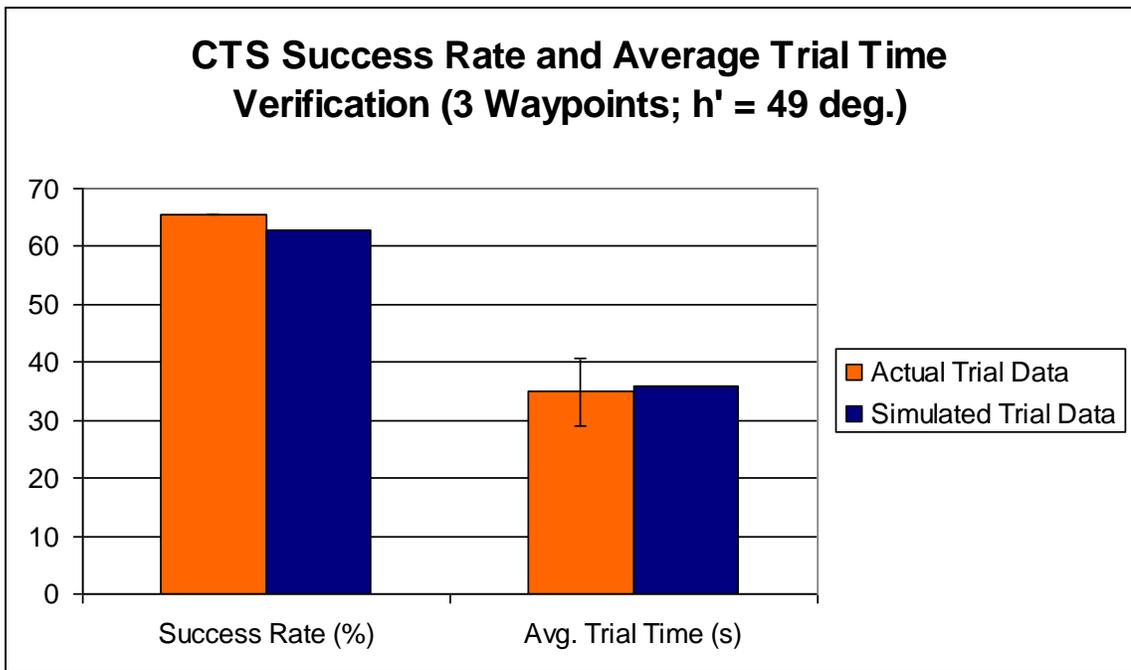


Figure 5.8. Comparison of three waypoint success rate and average trial time results for the CTS versus actual trial data.

As with the two waypoint trials, the field data was utilized to check the accuracy of the three waypoint CTS simulations. In order to gather more data under constant constraints, a single **orientation deviation threshold** was used in the three waypoint field trials—namely, 49 degrees—and 52 field trials were conducted using that threshold. Figure 5.8 presents the CTS versus the actual results of the success rate and the average trial time, which again proved to be extremely close to each other. Notice that the average trial times are closer together in the three waypoint trials. The simulated trial results are deep within the standard deviation of the actual trial data. This could be due to the fact that the trials are longer. Therefore, there is more opportunity for the canine to “make up” the lost time throughout each trial. It is also likely that the hand timing of each trial was more accurate, since the author became more accustomed to the timing procedures by the time the two waypoint trials took place. Plus, the three waypoint trials used for comparison were conducted in the structured environment, which in this case allowed for easier retrieval of the bumper by the canine.

5.3.8 Conclusion

As was discussed earlier, if the **orientation deviation threshold** is chosen too small, the canine will be forced to stop and receive directional changes more often. This will result in longer trial times and more potential for failure, since the canine does not respond as readily to directional commands. If the **orientation deviation threshold** is chosen to be too large, the canine success rate may be relatively high. However, the trial time will also potentially be much higher, since the canine will be allowed more “leeway” to get farther off course. Taking into account these considerations, intuition and

expert experience says that the optimum **orientation deviation threshold** should be somewhere between 40 and 50 degrees. The CTS results verify the results expected from intuition.

Chapter 6

Conclusions

6.1 Conclusion

This dissertation presents a novel approach for tracking and autonomously controlling a guided canine to prescribed waypoints using an Extended Kalman Filter and a Canine Maximum Effort Controller. In order to autonomously control a canine, a platform was developed that would allow for the retrieval of measurements for tracking the canine and issuing commands to the canine. As discussed in Chapter 2, this platform was composed of the command unit, sensor suite, remote computational device, and a canine vest. This platform included the software necessary for interfacing electronic hardware components.

The next task was to develop algorithms that would grant position, velocity, and orientation information describing the canine's behavior for use in the autonomous control algorithms. As is shown in Figure 1.2, the control system retrieves measurements from inertial sensors, magnetometers, and a GPS receiver and filters the data in order to attain optimal state estimates of the canine's position, orientation, and velocity. Utilizing fuzzy logic to adaptively tune the Process Noise Covariance Matrix of the EKF based on the canine's motion allowed for an excellent canine tracking system during brief GPS

outages. After simulated 10 second GPS outages, the adaptive EKF scheme resulted in average position solutions 7.5 meters away from the GPS solution and average velocity solutions 1.2 meters per second away from the GPS solution. This proved to be a significant improvement over the tracking approach discussed in [71].

After arriving at an optimal tracking solution, a Canine Maximum Effort Controller was developed to autonomously guide the canine to pre-planned, successive waypoints using the tracking solution. The controller proved to work effectively, achieving an overall 97.7% trial success rate on two waypoint trials, a 70.1% trial success rate on three waypoint trials, and an overall trial success rate of 86.6%.

In order to determine an optimal choice for the **orientation deviation threshold** to be used in the CMEC in future work without resorting to trial and error, a Canine Behavior Statistical Model was developed, which is comprised of actual statistical information that describes the canine's behavior over time. A Canine Trial Simulator was then developed to simulate different types of autonomous control trials using the CBSM and the CMEC and thereby determine the best **orientation deviation threshold** choice for use in the CMEC, given the statistical information about the canine provided by the behavioral model.

6.2 Future Work

Several avenues are available for possible future work to make improvements on the control system presented in this dissertation.

- ⇒ Hardware upgrades could be made, including moving to an IMU that is capable of reading higher magnitude accelerations and rates of turn. An XSens is available which would meet these needs, although developing an IMU with magnetometer integration in house would also be an attractive option.

- ⇒ Less bulkiness on the canine harness is always a goal worthy of attention. Moving to a smaller GPS antenna would be ideal. However, this would also likely require switching to radios that do not create satellite tracking interference.

- ⇒ The tracking algorithms could likely be improved if a microprocessor upgrade was made, as well.
 - ◆ With more processing power and memory, a tightly coupled EKF could be used which would make GPS outages less likely.

 - ◆ GPS position accuracy improvements could be made using, for instance, differential GPS [92].

 - ◆ A canine AHRS could be developed that would grant Euler angle estimates without relying on the AHRS solution provided by the XSens, since the XSens was tuned for pedestrian and/or automotive applications.

- ◆ A higher order EKF could be used that considers the effect of roll and slip in the canine motion model.

- ⇒ It may also be of interest to further investigate canine running motion to finely tune the fuzzy logic adaptively tuned EKF scheme presented in Chapter 3.

- ⇒ Other approaches could be investigated to improve the navigation solution as well, other than using the adaptive tuning approach, including developing a scheme to estimate two separate accelerometer biases “simultaneously”—one that would represent detection mode bias and one that would represent running mode bias. The EKF would switch states depending upon the GPS velocity measurement.

- ⇒ As was discussed earlier, promising results have surfaced from navigation solutions based on stride lengths. Investigating the effectiveness of that approach for the canine application may be beneficial coupled with a frequency analysis of the accelerometer measurements to analyze canine step frequency.

- ⇒ More effective means to guide the canine are constantly being considered. As was discussed above, the limiting factor in the canine control problem is the canine’s success rate in following directional commands. It may be of benefit to increase the strength of the vibration and/or use vibrations on the opposite side of the canine from the direction of the turn in order to tell the canine which way to **not** go, instead of

- which way to go. The trainers currently use a similar approach in some of their training protocols.
- ⇒ It also should be noted that the canine's success rate in adhering to tonal commands is virtually flawless. Moving to tonal commands for directional cues may be an option worth investigating first.
 - ⇒ Determining the canine's success rate when being guided to 4 or 5 waypoints would be of interest in the long run of the project, since real world applications may require more turns in order to reach the final destination. However, it would likely be more fruitful for this investigation to be carried out after one of the training changes mentioned above, which could significantly increase the canine's success rate in obeying directional commands.
 - ⇒ The CDRI is currently still investigating the possibility of eliminating the need for visual cues altogether, which would make the canine more like a robot in following commands. This is attractive to engineers. However, it may be determined that the canine's ability to see potential or likely targets that handlers/computers cannot sense is a benefit that the canine possesses which would not be of benefit to dampen in many real-world applications.
 - ⇒ Several potential avenues for future work were discussed in Chapter 5 concerning the Canine Trial Simulator. Of particular interest would be developing a CBSM comprised of many more parameters during a canine's training phase. This could

allow the CTS to consider other factors such as environment and canine fatigue when running simulations. However, it may be found that data gathered during the training phase would not represent the canine's behavior post-training.

The field of autonomous creature control is certainly a novel area in the engineering community. Autonomous control of canines is even more new. The possible avenues of future work in this field are many and exciting. More importantly, I am thankful to have had the opportunity to be a part of this work due to the potential good that can be done for humanity through such endeavors.

References

- [1] C. Lin, H. Hung, Y. Chen, and B. Chen. “Development of an Integrated Fuzzy – logic-based Missile Guidance Law Against High Speed Target.” *IEEE Transactions on Fuzzy Systems*, Vol. 12, No. 2. 2004. pp. 157-169.
- [2] L.C. Bento and U. Nunes. “Autonomous Navigation Control with Magnetic Markers Guidance of a Cybernetic Car using Fuzzy Logic.” *Machine Intelligence & Robotic Control*, Vol. 6, No. 1. 2004. pp. 1-10.
- [3] S. Lee, K. Kwon, and J. Joh. “A Fuzzy Logic for Autonomous Navigation of Marine Vehicles Satisfying COLREG Guidelines.” *International Journal of Control, Automation, and Systems*, Vol. 2, No. 2. 2004. pp. 171-181.
- [4] G.M. Dimirovski, S.M. Deskovski, and Z.M. Gacovski. “Classical and Fuzzy-system Guidance Laws in Homing Missiles Systems.” *IEEE Aerospace Conference Proceedings*. 2004.
- [5] D. Loebis, R. Sutton, J. Chudley, and W. Naeem. “Adaptive Tuning of a Kalman Filter via Fuzzy Logic for an Intelligent AUV Navigation System.” *Control Engineering Practice*, Vol. 12. 2004. pp. 1531-1539.
- [6] J.E. Naranjo, C. Gonzalez, R. Garcia, T. Pedro, and R.E. Haber. “Power-Steering Control Architecture for Automatic Driving.” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 6, No. 4. 2005. pp. 406-415.

- [7] P. Guan, X. Liu, and J. Liu. "Adaptive Fuzzy Sliding Mode Control for Flexible Satellite." *Engineering Applications of Artificial Intelligence*, Vol. 18. 2005. pp. 451-459.
- [8] C. Hsu and C. Lin. "Fuzzy-Identification-Based Adaptive Controller Design via Backstepping Approach." *Fuzzy Sets and Systems*, Vol. 151. 2005. pp. 43-57.
- [9] J. Garcia, J.M. Molina, J.A. Besada, and J.I. Portillo. "A Multitarget Tracking Video System Based on Fuzzy and Neuro-fuzzy Techniques." *EURASIP Journal on Applied Signal Processing*, Vol. 14. 2005. pp. 2341-2358.
- [10] J. Sheu. "A Fuzzy Clustering Approach to Real-time Demand-responsive Bus Dispatching Control." *Fuzzy Sets and Systems*, Vol. 150. 2005. pp. 437-455.
- [11] J.E. Naranjo, C. Gonzalez, R. Garcia, and T. Pedro. "ACC+Stop&Go Maneuvers With Throttle and Brake Fuzzy Control." *IEEE Transactions on Intelligent Transportation Systems*, Vol. 7, No. 2. 2006. pp. 213-225.
- [12] Z. Avdagic, E. Cernica, and S. Konjicija. "Longitudinal Vehicle Guidance using Fuzzy Logic." *IEEE Conference on Industrial Technology*. 2006. pp. 893-898.
- [13] M. Oosterom and R. Babuska. "Design of a Gain-scheduling Mechanism for Flight Control Laws by Fuzzy Clustering." *Control Engineering Practice*, Vol. 14. 2006. pp. 769-781.
- [14] M.M. Elkhatab and D.J. Hamilton. "A Modified Proportional Navigation Based Fuzzy Logic Controller for a Nonholonomic Car-like Robot." *Proc. of the IEE International Conf. of Control*, Glasgow. 2006.
- [15] K. Tu and J. Baltes. "Fuzzy Potential Energy for a Map Approach to Robot Navigation." *Robotics and Autonomous Systems*, Vol. 54. 2006. pp. 574-589.

- [16] A. Zhu and S.X. Yang. “Neurofuzzy-based Approach to Mobile Robot Navigation in Unknown Environments.” *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews*, Vol. 37, No. 4. 2007. pp. 610-621.
- [17] M. Mucientes, D.L. Moreno, A. Bugarin, and S. Barro. “Design of a Fuzzy Controller in Mobile Robotics using Genetic Algorithms.” *Applied Soft Computing*, Vol. 7. 2007. pp. 540-546.
- [18] B. Innocenti, B. Lopez, and J. Salvi. “A Multi-agent Architecture with Cooperative Fuzzy Control for a Mobile Robot.” *Robotics and Autonomous Systems*, Vol. 55. 2007. pp. 881-891.
- [19] S. Chabra and S.E. Talole. “Fuzzy Logic-based Terminal Guidance with Impact Angle Control.” *Defence Science Journal*, Vol. 57, No. 4. 2007. pp. 351-360.
- [20] J.E. Naranjo, C. Gonzales, R. Garcia, T. Pedro, and M.A. Sotelo. “Using Fuzzy Logic in Automated Vehicle Control.” *IEEE Intelligent Systems*. 2007. pp. 36-45.
- [21] C. Shieh. “Nonlinear Robust Digital Controller Based on Fuzzy Model for Homing Missiles.” *Chaos, Solitons and Fractals*, Vol. 38. 2008. pp. 192-198.
- [22] A. Zogheib. “Autonomous Navigation Tool for Real and Virtual Field Robots.” *1st International Conf. on Mach. Control and Guidance*. 2008. pp. 1-11.
- [23] H.M. Omar. “Designing Multi-Stage Fuzzy Guidance Law via Genetic Algorithms.” *8th World Congress on Computational Mechanics, 5th European Congress on Computational Methods in Applied Sciences and Engineering*. 2008.

- [24] W.M. Bessa, M.S. Dutra, and E. Kreuzer. "Depth Control of Remotely Operated Underwater Vehicles using an Adaptive Fuzzy Sliding Mode Controller." *Robotics and Autonomous Systems*, Vol. 56. 2008. pp. 670-677.
- [25] H.H. Afshari, J. Roshanian, and A.B. Novinzadeh. "A Perturbation Approach in Determination of Closed-loop Optimal-fuzzy Control Policy for Planetary Landing Mission." *Proc. of the IMechE*, Vol. 223, Part G: J. Aerospace Engineering. 2009. pp. 233-243.
- [26] C. Cheng, S. Shu, and P. Cheng. "Attitude Control of a Satellite Using Fuzzy Controllers." *Expert Systems with Applications*, Vol. 36. 2009. pp. 6613-6620.
- [27] S. Li and L. Yuan. "Design of Fuzzy Logic Missile Guidance Law with Minimal Rule Base." *IEEE 2009 6th International Conference on Fuzzy Systems and Knowledge Discovery*. 2009. pp. 176-180.
- [28] C. Lin, Y. Lin, and T. Wang. "A Fuzzy Guidance Law for Vertical Launch Interceptors." *Control Engineering Practice*, Vol. 17. 2009. pp. 914-923.
- [29] M. Omid, M. Lashgari, H. Mobli, R. Alimardani, S. Mohtasebi, and R. Hesamifard. "Design of Fuzzy Logic Control System Incorporating Human Expert Knowledge for Combine Harvester." *Expert Systems with Applications*. 2010.
- [30] Y. Lai and F. Hsiao. "Application of Fuzzy Logic Controller and Pseudo-attitude to the Autonomous Flight of an Unmanned Aerial Vehicle." *Journal of the Chinese Institute of Engineers*, Vol. 33, No. 3. 2010.

- [31] J. Marshall. "The Cyborg Animal Spies Hatching in the Lab." *New Scientist*, iss. 2646. 6 Mar., 2008. <http://www.newscientist.com/article/mg19726461.800-the-cyborg-animal-spies-hatching-in-the-lab.html?full=true>. 27 Mar., 2010.
- [32] W.J. Gomes, D. Perez, and J.A. Catipovic. "Autonomous Shark Tag with Neural Reading and Stimulation Capability for Open-ocean Experiments." *Eos Trans. AGU*. 87(36), Ocean Sci. Meet. Suppl., Abstract OS45Q-05. 2006.
- [33] Shandong University of Science and Technology. "SDUST Created Remote-controlled Pigeon." 2 Sep., 2007. http://www.sdkd.net.cn/en/news_show.php?id=65. 12 Sep., 2007.
- [34] S. Brown. "Stealth sharks to patrol the high seas." *New Scientist*. 4 Mar., 2006. pp. 30-31.
- [35] Y. Li and S. Panwar. "A Wireless Biosensor Network Using Autonomously Controlled Animals." *IEEE Network*. 2006. pp. 6-11.
- [36] S. Talwar, S. Xu, E. Hawley, S. Weiss, K. Moxon, and J. Chapin. "Rat navigation guided by remote control." *Nature*, Vol. 417(6884). 2002. pp. 37-38.
- [37] R. Holzer, I. Shimoyama, and H. Miura. "Locomotion Control of a Bio-Robotic System via Electric Stimulation." *International Conference on Intelligent Robots and Systems*. Grenoble, France. 1997.
- [38] W. Song, J. Chai, T. Han, and K. Yuan. "A Remote Controlled Multimode Micro-stimulator for Freely Moving Animals." *Acta Physiologica Sinica*, 58(2). 2006. pp. 183-188.
- [39] VNUnet UK. "Researchers Develop 'Robo-roach.'" *UNU-MERIT: I&T Weekly*, Iss. 7. 28 Sep., 2001. United Nations University.

http://www.merit.unu.edu/i&tweekly/i&tweekly_previous.php?issue=0107&issue_show=7&year=2001. 13 Sep., 2007.

- [40] H. Sato, C.W. Berry, B.E. Casey, G. Lavella, Y. Yao, J.M. Vandenbrooks, and M.M. Maharbiz. "A Cyborg Beetle: Insect Flight Control Through an Implantable, Tetherless Microsystem." *IEEE MEMS 2008 Conf.* 2008. pp. 164-167.
- [41] H. Sato, Y. Peeri, E. Baghoomian, C.W. Berry, and M.M. Maharbiz. "Radio-controlled Cyborg Beetles: A Radio-frequency System for Insect Neural Flight Control." *IEEE MEMS 2009 Conf.* 2009. pp. 216-219.
- [42] A. Bozkurt, R. Gilmour, D. Stern, and A. Lal. "MEMS Based Bioelectronic Neuromuscular Interfaces for Insect Cyborg Flight Control." *IEEE MEMS 2008 Conf.* 2008. pp. 160-163.
- [43] "Grand Challenge: Smart Vest for Detector Dogs." *National Aerospace & Electronics Conference*. <http://www.naecon.org/challenge.htm>. 18 May, 2010.
- [44] "Detection Services." *Amdetech: Protection Through Detection*. <http://www.amdetech.com>. 18 May, 2010.
- [45] W.R. Britt, J. Miller, P. Waggoner, D.M. Bevly, and J.A. Hamilton. "An Embedded System for Real-time Navigation and Remote Command of a Trained Canine." DOI 10.1007/s00779-010-0298-4. 2010.
- [46] N. Correll, M. Schwager, and D. Rus. "Social Control of Herd Animals by Integration of Artificially Controlled Congeners." *Proc. of the 10th International Conference on Simulation of Adaptive Behavior*. 2008. pp. 437-447.

- [47] M. Schwager, C. Detweiler, I. Vasilescu, D.M. Anderson, and D. Rus. “Data-driven Identification of Group Dynamics for Motion Prediction and Control.” *Journal of Field and Service Robotics*, Vol. 25(6-7). 2008. pp. 305-324.
- [48] W. Britt. “A Software and Hardware System for the Autonomous Control and Navigation of a Trained Canine.” *Ph.D. Dissertation*, Auburn University. Summer, 2009.
- [49] D.M. Bevly. “Evaluation of a Blended Dead Reckoning and Carrier Phase Differential GPS System for Control of an Off-Road Vehicle.” *Proc. of the ION GPS*, Nashville, TN. 14-17 Sep., 1999. pp. 2061-2069.
- [50] S. Godha. “Performance Evaluation of Low Cost MEMS-Based IMU Integrated with GPS for Land Vehicle Navigation Application.” *MS Thesis*, University of Calgary. Feb., 2006.
- [51] J. Ryu and J.C. Gerdes. “Integrating Inertial Sensors with GPS for Vehicle Control.” *Jour. of Dynamic Systems Measurement, and Control*, Vol. 126, Issue 2. June, 2004. pp. 243-254.
- [52] D.M. Bevly. “Global Positioning System (GPS): A Low-Cost Velocity Sensor for Correcting Inertial Sensor Errors on Ground Vehicles.” *Jour. of Dynamic Systems Measurement, and Control*, Vol. 126, Issue 2. June, 2004. pp. 255-264.
- [53] J.B. Derrick, D.M. Bevly, and A.K. Rekow. “Model-Reference Adaptive Steering Control of a Farm Tractor with Varying Hitch Forces.” *Proc. of the American Control Conference*, Seattle, WA. June, 2008.

- [54] T.N. Upadhyay, S. Cotterhill, and A.W. Deaton. "Autonomous GPS/INS Navigation Experiment for Space Transfer Vehicle." *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 29(3). Jul., 1993. pp. 772-785.
- [55] V. Gabaglio. "GPS/INS Integration for Pedestrian Navigation." *Ph.D. Dissertation*, Institute of Geomatics of the Swiss Federal Institute of Technology in Lausanne. 2003.
- [56] O. Ladetto, V. Gabaglio, and B. Merminod. "Combining Gyroscopes, Magnetic Compass and GPS for Pedestrian Navigation." *Proc. Int. Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS 2001)*. 2001. pp. 205-213.
- [57] A.J. Hill, A. Slamka, Y.T. Morton, M. Miller, and J. Campbell. "A Real-Time Position, Velocity, and Physiological Monitoring and Tracking Device for Equestrian and Race Training." *Proc. of the ION GNSS*. Sept. 2007.
- [58] A. Ferworn, A. Sadeghian, K. Barnum, D. Ostrom, H. Rahnama, and I. Woungang. "Rubble Search with Canine Augmentation Technology." *IEEE International Conference on System of Systems Engineering*, DOI 10.1109/SYSOSE.2007.4304328. 2007. pp. 1-6.
- [59] C. Ribeiro, A. Ferworn, M. Denko, J. Tran, and C. Mawson. "Wireless Estimation of Canine Pose for Search and Rescue." *IEEE International Conference on System of Systems Engineering*, DOI 10.1109/SYSOSE.2008.4724172. 2008. pp. 1-6.
- [60] S. Watanabe, M. Izawa, A. Kato, Y. Ropert-Coudert, and Y. Naito. "A New Technique for Monitoring the Detailed Behaviour of Terrestrial Animals: a Case

- Study with the Domestic Cat.” *Appl. Animal Behav. Sci.*, Vol. 94. 2005. pp. 117-131.
- [61] C. Cornou and S. Lundbye-Christensen. “Classifying Sow’s Activity Types from Acceleration Patterns: an Application of the Multi-Process Kalman Filter.” *Appl. Animal Behav. Sci.*, Vol. 111. 2008. pp. 262-273.
- [62] M. Moreau, S. Siebert, A. Buerkert, and E. Schlecht. “Use of a Tri-axial Accelerometer for Automated Recording and Classification of Goat’s Grazing Behavior.” *App. Animal Behav. Sci.*, In Press, corrected proof. DOI 10.1016/j.applanim.2009.04.008. 2009.
<http://www.sciencedirect.com/science/article/B6T48-4WBB6TP-1/2/af33be442803a939109fbd72dfdb505c>.
- [63] Navsys Corporation: GPS Receivers. “TrackTag: Ultra Low-Power GPS Data Recorder.” <http://www.navsys.com/Products/tracktag.htm>. 3 Oct., 2008.
- [64] Osmografo. “Supporting Canine Rescue Teams.” GMV Innovating Solutions. Flier. 17 Sept., 2008.
- [65] Garmin. “Astro 220: Dog Tracking System.” <http://www8.garmin.com/astro/>. 3 Oct., 2008.
- [66] NAECON. “NAECON 2010 Grand Challenge—The Smart Vest for Detector Dogs.” http://www.naecon.org/2010/NAECON_2010_Grand_Challenge.pdf. 21 May, 2010.
- [67] R. Stengel. *Optimal Control and Estimation*. Dover Publications, Mineola, NY. 1994. ISBN: 0-486-68200-5.

- [68] G. Franklin, D. Powell, and M. Workman. *Digital Control of Dynamic Systems*, 3rd Edition, Addison Wesley Longman Inc., Menlo Park, CA. 1998.
- [69] J. Miller and D.M. Bevly. "Position and Orientation Determination for a Guided K-9." *Proceedings of the ION GNSS*, Ft. Worth, TX. 2007.
- [70] J. Miller and D.M. Bevly. "Guided K-9 Tracking Improvements Using GPS, INS, and Magnetometers." *Proceedings of the ION ITM*, Anaheim, CA. 2009.
- [71] J. Miller and D.M. Bevly. "Determination of Pitch Effects in Guided K-9 Tracking." *Proceedings of the JSDE/ION JNC*, Orlando, FL. 2009.
- [72] O. Ladetto, V. Gabaglio, and B. Merminod. "Combining Gyroscopes, Magnetic Compass and GPS for Pedestrian Navigation." *Proc. Int. Symposium on Kinematic Systems in Geodesy, Geomatics and Navigation (KIS 2001)*. 2001. pp. 205-213.
- [73] D. Gebre-Egziabher, G.H. Elkaim, J.D. Powell, and B.W. Parkinson. "Calibration of Strapdown Magnetometers in Magnetic Field Domain." *Journal of Aerospace Engineering*. Apr., 2006. pp. 87-102.
- [74] M.J. Caruso. "Applications of Magnetoresistive Sensors in Navigation Systems." *Sensors and Actuators 1997*, SAE SP-1220. Feb., 1997. pp. 15-21.
- [75] D. Gebre-Egziabher. "Design and Performance Analysis of a Low-Cost Aided Dead Reckoning Navigator." *Ph.D. Dissertation*, Stanford University. Feb., 2004.
- [76] Xsens Technologies B.V. "MTi and MTx User Manual and Technical Documentation." Document MT0100P, Revision N. 27 May, 2009. p. 5.

- [77] “Canine Skeleton.” Anatomical Adaptation for Cursorial Locomotion. Univ. of Minnesota Col. of Veterinary Medicine. <http://vanat.cvm.umn.edu/run/plate3.html>. 1 June, 2010.
- [78] Q. Ladetto, V. Gabaglio, and B. Merminod. “Two Different Approaches for Augmented GPS Pedestrian Navigation.” *Intl. Symposium on Location Based Services for Cellular Users*. 2001. pp. 1-7.
- [79] J. W. Kim, H. J. Jang, D. Hwang, and C. Park. “A Step, Stride and Heading Determination for the Pedestrian Navigation System.” *Journal of Global Positioning Systems*, Vol. 3, No. 1-2. 2004. pp. 273-279.
- [80] S. Moafipoor, D.A. Grejner-Brzezinska, and C.K. Toth. “Multi-sensor Personal Navigator Supported by Adaptive Knowledge Based System: Performance Assessment.” *Position, Location and Navigation Symposium*. 2008. pp. 129-140.
- [81] S. Godha, G. Lachapelle, and M.E. Cannon. “Integrated GPS/INS System for Pedestrian Navigation in a Signal Degraded Environment.” *Proc. of the ION GNSS*, Ft. Worth, TX. 2006.
- [82] G. Lachapelle, S. Godha, and M.E. Cannon. “Performance of Integrated HSGPS-IMU Technology for Pedestrian Navigation under Signal Masking.” *Proc. of the European Navigation Conference*, Manchester. 2006.
- [83] J.Z. Sasiadek. “Sensor Fusion.” *Annual Reviews in Control*, Vol. 26. 2002. pp. 203-228.
- [84] V. Subramanian, T.F. Burks, and W.E. Dixon. “Sensor Fusion Using Fuzzy Logic Enhanced Kalman Filter for Autonomous Vehicle Guidance in Citrus Groves.” *Transactions of the ASABE*, Vol. 52(5). 2009. pp. 1411-1422.

- [85] K.M. Passino and S. Yurkovich. *Fuzzy Control*. Menlo Park, Cal.: Addison Wesley Longman. 1998.
- [86] L.A. Zadeh. "Fuzzy Sets." *Information and Control*, Vol. 8. 1965. pp. 338-353.
- [87] L. Reznik. *Fuzzy Controllers*. Oxford: Newnes. 1997.
- [88] A.P. Engelbrecht. *Computational Intelligence: An Introduction*. 2nd edition. West Sussex, England: John Wiley and Sons, Ltd. 2007.
- [89] T. Terano, K. Asai, and M. Sugeno. *Applied Fuzzy Systems*. English edition. Academic Press. 1994.
- [90] J. Favilla, A. Machion, and F. Gomide. "Fuzzy Traffic Control: Adaptive Strategies." *Proc. of the IEEE Symposium on Fuzzy Systems*. 1993.
- [91] L. Wang and J. Mendel. "Generating Fuzzy Rules by Learning from Examples." *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 22(6). 1992. pp. 1413-1426.
- [92] W. Travis. "Path Duplication Using GPS Carrier Based Relative Position for Automated Ground Vehicle Convoys." *Ph.D. Dissertation*, Auburn University. Spring, 2010.
- [93] P. Bonissone. "Control Theory." *Fuzzy Sets and Expert Systems Course Notes*, Rensselaer Polytechnic Institute, Troy, NY. 1998. <http://www.rpi.edu/~bonisp/fuzzy-course/99/FIc/control-intro.pdf>. 26 July, 2010.
- [94] S. Lenhart. "Bang-bang and Singular Controls." *Lecture Notes*, University of Tennessee, Knoxville. 2003. <http://www.math.utk.edu/~lenhart/docs/SL-lecture4.pdf>. 26 July, 2010.

- [95] N. Tien. “C.11 Bang-bang Control.” *Introduction to Control Theory Including Optimal Control Lecture Notes*, Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam. 5 Sept., 2006.
http://www4.hcmut.edu.vn/~nttien/Lectures/Optimal_Control/C.11_Bang-bang_Control.pdf. 26 July, 2010.
- [96] J. Valvano. *Embedded Microcomputer Systems: Real Time Interfacing*. 2nd edition. Thomson-Engineering Publishers. 2006.
- [97] “Chapter 1—Introduction to Control: Thermal Systems.” *Introduction to Modeling and Control Course Reading Material*. Franklin W. Olin College of Engineering, Needham, MA. 2003.
http://ece.olin.edu/icb/fall_03/mc/reading/PID.pdf.
- [98] “Hands-Free, Radio/Audio Silent Navigation System (HFSNS) for Dismounted Soldiers.” Integrated Solutions for Systems SBIR Proposal A102-073-0989. June, 2010.

Appendix A

Canine Training Procedures

This appendix describes the training protocol for the canines used in this research as articulated by the CDRI. The protocols are reproduced here verbatim with the author's approval.

A.1 Equipment

A custom made harness (Blackthorn K9 Equipment, Inc.) was employed in training canine Major. The harness was equipped with a prototype *K9 Remote Sensor System* engineered by Joe Reiter of WYLE Laboratories, Inc. for an ongoing project which is sponsored by the Transportation Security Laboratory of the U. S. Department of Homeland Security Science and Technology Directorate. The Wyle Remote Sensor System was designed to provide first responders with a remotely controlled explosive detection capability that could also serve as a platform for a variety of other sensors (e.g., video, audio, radiological, air quality, and physiological monitoring of the dog), the output of which can be transmitted to responders in real-time. For the purpose of the present experiment, the important function of the Wyle Remote Sensor System was in delivering radio controlled auditory and vibration signals to canine Major. The canine

command functions were controlled by a Wyle developed tone generator worn by the dog and actuated by a pair of Motorola EX600 radios, one worn by the dog and the other controlled by the handler of the dog. The key pad of the EX600 radios served as the handler-dog interface and pressing different keys of the radio resulted in the presentation or different tones from a speaker on the harness driven by the Wyle tone generator or actuation of one of two vibrators mounted on the underside of harness. The commands used included: “Back” (i.e., leave the handler and move in a straight line using the initial physical orientation), which was a 3,500 Hz tone issued from the speaker; “Stop” (i.e., the dog halts and sits), which was a 9,500 Hz tone; “Over Left” (i.e., the dog turns left in relation to its present orientation), which was a vibrational cue located on the left rear of the harness; “Over Right” (i.e., the dog turns right in relation to its present orientation), which was a vibrational cue located on the right front shoulder of the harness and; “Recall” (i.e., the dog returns to the starting point of the trial or current position of handler), which was a 1,500 Hz tone. An electronic collar (Dogtra, Inc.) was used in initial training to establish and apply negative reinforcement in the form of brief (<2s) mild (30-80 mA) electrical stimulation and vibration (such that the vibration was established as a conditioned negative reinforce that could intermittently function effectively without presentation of the electrical stimulation) for the emission of correct responses signaled by voice, whistle, visual, and, later, tone and vibration commands.

A.2 Training

A.2.1 Odor Detection Training

Canine Major was initially trained to “sit” in the presence of odor from C-4 explosive material by prompting him to place his nose near a box containing C-4 explosive (target) and issuing the verbal command “sit”, which he already reliably emitted. Sitting at the target box resulted in delivery and opportunity to chase and play with a tennis ball or other toy. This activity was repeated until Major placed his nose near the opening of the box and sat in the absence of the verbal command. Next, a second identical box was added to the training scenario that did not contain C-4 (i.e., blank) and Major was prompted to sample from both boxes, the order of which was randomly alternated. With two boxes, one blank and one with the target, sitting at the target containing box was reinforced and Major was encouraged to **not** sit at the blank box by the verbal command “No”, which was previously established as a conditioned punisher by association with mild leash corrections and interruption of opportunities for reinforcement in obedience type training, and mild leash pressure to not allow him to complete the sit response. Once Major reliably discriminated between the target and blank box (i.e., approximately 20 contiguous errorless trials), a 3rd and subsequently 4th blank box was added to the scenario. Once Major was reliably sitting at the target box and **not** sitting at any blank boxes in this, so called, “4-hole variable” in typical detector dog training, training to search for and alert to the target odor in varied environments was initiated.

Initially, the target material was placed in relatively easy to detect locations taking relatively a short time to encounter from the point at which Major was prompted to begin searching for the target odor. Again, detection of the target, was reinforced with delivery and opportunity to play with a ball or other toy and sitting at other locations was either

interrupted by the verbal command “No” or resulted in the command “No” and discontinuation of the particular trials and thus, opportunity to receive the ball for responding correctly. The locations of the target “hides” were gradually made deeper, higher, or otherwise more difficult for detection of the target and the distance, and thus amount of time searching required, was extended.

Finally, target odor detection training was incorporated in the context of Major’s pre-existing field trial training. Initially, trials would be alternated between blind retrieves in which Major would be directed to an area where there was a bumper to retrieve and trials in which he was directed to an area where the target material was hidden. When Major encountered the odor from the target material, he would, as is typical, show a noticeable interest and begin to interrogate the area with his nose and when he was near the target material hide would sit, or if he did not sit, the handler would verbally command Major to sit, then issue intermittent whistle commands for Major to return and along the way be verbally praised and then thrown a ball when in range of the handler. This training progressed by hiding the target material in more challenging and unexpected places including intermittent locations along the usual routes of a field trial run and in and around objects, buildings, and vehicles to which major would be directed or come near during a trial. This phase of training was considered complete when Major’s following of directional commands was reliably interrupted by encountering a target odor and he searched for and sat (i.e, alerted) as close to the source of the target odor as he was capable.

A.2.2 Remote Tone and Vibration Control

In the initial field trial training the dogs had received prior to inception of this project, voice, whistle, and visual commands were used to direct movement; the following describes the commands and resulting trained response:

- ⇒ “back” - from sitting or healing position oriented to handlers posture, dog runs away from handler maintaining general heading of original orientation

- ⇒ solid whistle – dog ceases ongoing movement and orients toward handler

- ⇒ intermittent whistle – dog recalls (returns) to handler

- ⇒ “over” and handler moving slightly and casting arm high with palm open facing toward dog in desired left or right direction – stopped and/or sitting dog at some distance from handler runs in direction casted.

Several steps were involved in transferring control from voice, whistle, and visual commands to the remotely issued tone and vibration commands delivered by equipment on the k9 harness. Additionally, modifications of the trained responses were made to accommodate remote and autonomous guidance.

Transfer of the back and recall commands were accomplished by issuing the tone command to be associated with the trained response and nearly simultaneously issuing the respective previously trained back or recall command. Across many repetitions of this pairing of tone and previously trained commands, the intensity of the previously trained

commands (i.e., “back” and solid whistle) was faded (i.e., lessened) until the requisite response was reliably exhibited upon issuance of only the requisite tone.

To prepare the dogs for exhibiting left and right movements issued by the autonomous navigation system, the typography of their response for stop, left, and right required modification. The desired topography for autonomous guidance was for the dogs to stop and remain oriented in the direction of their previous heading, rather than turn to face the handler, and orient to the next desired heading in response to left or right tone commands rather than run in a left or right direction. Having the dog orient left or right in response to the left and right vibration signals on a generally stationary axis allows for the guidance system to sense the orientation of the dog, compare the orientation to the desired heading, and then issue the tone for the dog to run straight from that orientation or issue further left and right tones to re-orient the dog in the desired heading. For the purpose of the current project, training was conducted to have the dogs orient left/right 45° and 90° from their previous heading.

Training for stopping and remaining oriented to previous heading and orienting right and left per the requisite tones were accomplished using visual cues that would have been used in the dogs’ early field trial training. Piles of retrieval bumpers and stakes to provide some vertical relief to the position of those piles of retrieval bumpers are used in early field trail training to entice the dog to run to those piles upon being issued an “over” and right or left movement cast command. Retrieval of the bumper is a very desirable activity for these types of dogs and thus, the opportunity to run and retrieve a bumper, reinforces correct responses to the right and left commands. With a stake-identified bumper pile located in the direction a dog was running, the dogs were trained to stop and

remain oriented toward their previous heading by first pairing the stop tone with the continuous whistle and fading the whistle command. The presence of the stake and bumper pile enticed the dogs to remain oriented in their previous heading. Also, as the whistle signal was faded and control transferred to the stop tone being issued from the harness, the propensity of the dog to orient toward the source of the whistle (i.e., the handler) was reduced and they were prone to stopping and remaining in more or less the approximate orientation of their last heading. The desired response of orienting the canines in the direction of their previous heading upon issuance of the stop tone was further refined by following successive approximations of this response with the tone for running out (“back”) and accessing a bumper a retrieve and issuing the recall tone if they did not orient in their previous heading.

Training for orienting left or right upon issuance of right or left vibration signals used the stake-identified bumper pile to entice the dogs to orient toward such piles at 45⁰ and 90⁰ from their initial orientation relative to the handlers position. Because the dogs’ early training includes re-orienting based on the handlers body position, the handler could “nudge” the dog toward the desired orientation by moving their own body. Issuing the requisite vibration while prompting the dog with body movements toward the stake and bumper pile to which the dogs are inclined to orient toward in anticipation of being allowed to run and retrieve a bumper resulted in the right and left vibrations alone to begin prompting the dog to move left or right. This response was further refined by the handler fading their body position cues and progressively moving further away from the stationary dog prior to issuing right and left vibration commands. Once orienting to 45⁰ fairly consistently, at least upon multiple issuance of command, a stake and bumper pile

was positioned at 90⁰ without any stake or bumper pile at 45⁰ and the vibration command was issued for a longer duration to, together, prompt the dog to orient to 90⁰. The premise for this was that the autonomous system would issue the vibration command until it sensed the dog had reached the desired orientation.

For both the stop and right/left commands, the bumper piles were made intermittent (i.e., sometimes running to the stake resulted in bumper to retrieve sometimes not) and then the stakes were faded such that orienting toward the last heading when the stop was issued and orienting left and right from that heading upon issuing right or left vibration commands respectively was faded to being only under the control of the commands and intermittently reinforced by access to bumpers to retrieve. The bumpers were often in medium to tall grass at a distance such that the dog could not see and thus, not orient based on bumper pile position. In order to combine a stop with re-orientation of heading away from the handler as opposed to just reorientation from original position, the handler would issue the back tone from at first a short distance from handler before issuing the stop tone and then issuing left or right orientation vibration commands and then progressively increasing the distance traveled by the dog before issuing the stop tone followed by vibration commands to reorient the dog.

A.2.3 Final Training Status

Prior to the initiation of the experiments reported in this dissertation, canine Major was fully proficient at being remotely guided by tone and vibration commands and his ongoing following of such guidance being interrupted by the presence of target odor to which he would follow to its source and sit (i.e., alert). Canine Major would exhibit

this repertoire in a variety of environments including open fields, hard surfaces, around and about the exterior of buildings, and inside of large structures, such as subterranean mass transit (i.e., subway) venues.

One complicating aspect of Major's final performance for the execution of the current experiments was that he tended to take shortcuts to locations he had previously encountered target odor and/or to structure and objects similar to those where he had encountered target odor in the past. This propensity to disregard movement commands to go to such productive (i.e., areas associated with presence of target odor in past) areas was considered desirable as compared to the added control afforded by more consistent use of negative reinforcement that would be necessary for following movement commands. Such added control might interfere with the precedence established for disregarding movement commands to respond to the presence of target odor, which was the immediate operational imperative of the ongoing technology development project in which Major was participating before, during, and after the experiments reported in the present paper.

Appendix B

Preliminary Canine Position and Orientation Determination

B.1 Introduction

As presented in [69], in the preliminary phase of this project, a means of tracking the canine's position, velocity, and orientation was sought, in order to provide accurate measurements for the autonomous control problem. A slightly different navigation system than the one described in Chapter 2 was used to acquire tracking data for canine motion analysis. In order to map the location, velocity, and orientation of the canine at a given point, a low-cost, limited sensor suite composed of the consumer grade U-Blox GPS receiver mentioned above, as well as a six degree of freedom Sentra IMU containing three accelerometers and three gyroscopes was used.

B.2 Canine Testing

Although the approach being utilized to track the motion of the canine (i.e., GPS/INS integration) is not unique, the canine motion provided new challenges to motion tracking. For example, unlike what is typical when analyzing a vehicle, a canine bounces while trotting and also tends to slightly tilt back and forth while walking. Also, when coming to a stop, particularly after a sprint, excessive lateral motion can occur.

These effects have not been modeled in the algorithms from this phase and must be accounted for in the EKF tuning. Also, unlike man-made machines that respond in relatively predictable ways, the canine exhibits behaviors that are independent of the inputs. Therefore, many different possible canine motion situations must be accounted for in the design and tuning of the EKF algorithms. To account for these different scenarios, a series of controlled tests were run to demonstrate and determine the unique motion of a canine and to estimate the factors that could corrupt the navigation system.

Using a compass, estimated north and east axes were marked before running applicable tests to provide qualitative measures of how effective the EKF was in tracking the canine position (see Figure B.1). Next, the dog was walked to illustrate typical canine motions. Namely, the canine was instructed to:

- 1) Walk north; turn east; u-turn and walk west; then walk south (“L” test).
- 2) Walk east; turn and walk north, turn and walk west, then turn and walk south (“Box” test).

Using these tests, the EKF was tuned to reject some of the typical disturbances created by a dog's sporadic movements, but still maintain an acceptable level of navigational performance. Videos were also made to assist in checking for error.



Figure B.1. North and East axes.

B.3 Tracking Algorithm

For this phase of the project, the accelerometer measuring acceleration in the longitudinal direction and the gyroscope measuring yaw rate are used from the IMU. The velocity, course, north, and east positions are measured by the GPS receiver. These are all integrated by the loosely-coupled, EKF tracking algorithm described in section 3.1.2. The state estimate vector for this system is:

$$\hat{x} = [\hat{V} \quad \hat{b}_a \quad \hat{\psi} \quad \hat{b}_g \quad \hat{N} \quad \hat{E}]^T \quad (\text{B.1})$$

Where: \hat{V} = Estimated velocity

\hat{b}_a = Estimated accelerometer bias

$\hat{\psi}$ = Estimated heading

\hat{b}_g = Estimated gyroscope bias

\hat{N} = Estimated northern position

\hat{E} = Estimated eastern position

Figure B.2 visually illustrates the notations used in determining northern and eastern velocities.

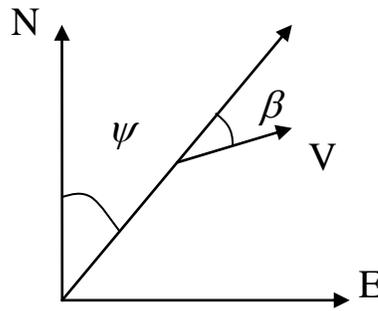


Figure B.2. Visual illustration of northern and eastern velocity determination.

The GPS receiver outputs course ($\psi + \beta$). However, the gyroscope outputs yaw rate, where the integral of yaw rate is heading. In order to integrate the yaw rate measurements with the GPS course measurements, the side slip, β , is assumed to be zero. Therefore, the estimated course ($\psi + \beta$) becomes estimated heading (ψ). Canine motion can inject errors into the EKF when the canine does not move precisely in the direction that it is facing (i.e., the canine motion contains some sideslip). However, for the initial development of the tracking algorithm, this variable is neglected.

The state equations used for GPS/INS integration are as follows:

$$\hat{V} = A_x^m - \hat{b}_a \quad (\text{B.2})$$

Where: A_x^m = Measured longitudinal acceleration from the accelerometer

\hat{b}_a = Estimated bias on the longitudinal accelerometer

\hat{V} = Estimated acceleration in the longitudinal direction

$$\hat{\psi} = g_z^m - \hat{b}_g \quad (\text{B.3})$$

Where: g_z^m = Measured yaw rate from the gyroscope measuring motion about the z-axis

\hat{b}_g = Estimated bias on the longitudinal gyroscope

The biases for this system are modeled as random walks, and the process noise is assumed to be zero mean, white noise (i.e., $\omega \sim N(0, \sigma_\omega^2)$). Therefore, the following equations represent the change in biases for the accelerometer and gyroscope:

$$\dot{b}_a = 0 + \omega_a \frac{m}{s^2} \quad (\text{B.4})$$

$$\dot{b}_g = 0 + \omega_g \frac{rad}{s}$$

Where: ω = Process noise

The estimated changes in northern and eastern directions are calculated with the following equations, according to Figure B.2.

$$\hat{N} = \hat{V} \cos \hat{\psi} \quad (\text{B.5})$$

$$\hat{E} = \hat{V} \sin \hat{\psi}$$

Note that the northern and eastern velocities are non-linear. Therefore, the Jacobian (Φ) is used for the covariance prediction in the EKF to linearize the system about the operating point. This results in the Jacobian matrix for the present system, shown below:

$$\Phi = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \cos \hat{\psi} & 0 & -\hat{V} \sin \hat{\psi} & 0 & 0 & 0 \\ \sin \hat{\psi} & 0 & \hat{V} \cos \hat{\psi} & 0 & 0 & 0 \end{bmatrix} \quad (\text{B.6})$$

There are two INS inputs to the integrated system—namely, A_x^m and g_z^m .

Therefore, the input vector is the following:

$$u = \begin{bmatrix} A_x^m \\ g_z^m \end{bmatrix} \quad (\text{B.7})$$

The above relationships and matrices are transferred into state space form.

The measured GPS velocity, V_{GPSM} , is equal to the actual velocity (state estimate number 1) plus GPS sensor noise and similarly for the other GPS measurements (see Equation (3.4)). So, the output equation can be reduced to the following:

$$y = \begin{bmatrix} V_{GPSM} \\ \psi_{GPSM} \\ N_{GPSM} \\ E_{GPSM} \end{bmatrix} \quad (B.8)$$

As mentioned above, the GPS Sensor Noise Covariance Matrix (R) and the Process Noise Covariance Matrix (Q) constitute the primary EKF tuning parameters.

$$R = \begin{bmatrix} \sigma_{GPSvel}^2 & 0 & 0 & 0 \\ 0 & \sigma_{GPShead}^2 & 0 & 0 \\ 0 & 0 & \sigma_{GPSnorth}^2 & 0 \\ 0 & 0 & 0 & \sigma_{GPSeast}^2 \end{bmatrix} \quad (B.9)$$

$$Q_k = \begin{bmatrix} \sigma_A^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{bA}^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_g^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{bg}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{north}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{east}^2 \end{bmatrix} \quad (B.10)$$

The Sensor Noise Covariance Matrix values are relatively standard and are acquired from the GPS sensor information.

$$\sigma_{GPS_{vel}} = 0.05 \frac{m}{s} \quad (B.11)$$

$$\sigma_{GPS_{north}} = 1.75 m$$

$$\sigma_{GPS_{east}} = 1.75 m$$

$$\sigma_{GPS_{head}} = \frac{0.5}{V_{GPSM}} \text{ radians}$$

The Process Noise Covariance Matrix values prove to be more variable due to the unique motion characteristics of a canine. Based on the navigation system for this phase of the project, these values were found to be the following:

$$0.01 \leq \sigma_A \leq 0.3 \frac{m}{s^2} \quad (B.12)$$

$$0.009 \leq \sigma_g \leq 0.02 \frac{rad}{s}$$

$$\sigma_{bA} \left(\frac{m}{s^2} \right) = \sigma_{bg} \left(\frac{rad}{s} \right) = 1 * e^{-10}$$

$$\sigma_{north} = \sigma_{east} = 1 * e^{-5} m$$

These noise characteristics were trial-based and varied due to different factors, especially the unique motion characteristics of canines. This will be discussed further and compensated for later in this chapter.

The state estimation covariance matrix was initialized as follows:

$$P = \begin{bmatrix} \sigma_{GPSvel}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 * e^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{GPSnorth}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{GPSeast}^2 \end{bmatrix} \quad (B.13)$$

Equations (B.1) through (B.13) were used in the standard EKF equations discussed above to attain the canine position, velocity, and orientation.

B.4 Challenges

Figures B.3-B.8 illustrate the challenge in attaining accurate tracking information when trying to utilize GPS/INS integration to accurately track canine motion. Figures B.3-B.5 show raw accelerator output for automobile, pedestrian, and canine motions, respectively. Figures B.6-B.8 illustrate raw yaw gyroscope data for automobile, pedestrian, and canine motions, respectively. A comparison of these figures illustrates that the pedestrian gyroscope and accelerometer data have higher frequency and magnitude oscillations than those for automotive applications and that the raw accelerometer and gyroscope sensor data for canines has even higher frequency and magnitude oscillations than the pedestrian data due to the motions of the canine. Whereas in automotive applications the magnitude of the typical range in acceleration is only about 0.6 m/s^2 (Figure B.3), in pedestrian applications it is common for there to be $1.5\text{-}2 \text{ m/s}^2$ magnitude oscillations in the accelerometer output (Figure B.4). Canine accelerometer output commonly sees magnitude fluctuations of $10\text{-}20 \text{ m/s}^2$ (Figure B.5), and when the canine is running, $40\text{-}50 \text{ m/s}^2$ changes in acceleration are common. Such

motion saturates the sensors and creates error which is lumped into the estimation of accelerometer bias. Whereas in automobile applications the magnitude of the typical range in yaw rate is only about 2 deg/s (Figure B.6), in pedestrian applications it is common for there to be oscillations between 20 and -20 deg/s in the gyroscope output (Figure B.7). Canine gyroscope output commonly fluctuates between 70 and -70 deg/s (Figure B.8).

Recall that one of the critical estimated states of interest for this study is the orientation of the canine (i.e., specifically heading). However, much of the high frequency motion illustrated in Figure B.8 is due to the unique motion characteristics of the canine, rather than the actual orientation change of the canine. The motions of the dog captured by the yaw gyroscope, for instance, are at a relatively higher frequency than the actual change of in the dog's orientation. This illustrates why tracking the biases on the inertial sensors is a difficult task. Standard, motion modeling equations for an EKF do not account for some of the motion characteristics of the canine. Therefore, more error is lumped into the bias estimates, which in turn, makes tracking the inertial sensor biases more difficult, which in turn, makes tracking the canine motion difficult, especially during GPS outages.

Consider also that almost any moving object is going to have accelerations inherent in its movements. However, in most scenarios, it is known that the average acceleration during a given period of time is going to be zero [76]. The measurements of said accelerations will, over a sufficient amount of time, yield an average acceleration of 0 m/s^2 plus a very small bias due to the nature of low cost, inertial sensors. However, in canine motion applications, the average acceleration does not turn out to be zero. In fact,

the average accelerometer data from a trial can routinely be above 1 m/s^2 , even though the actual canine acceleration does not actually reflect such data.

There are different factors that are coming into play that add excess “noise” to the accelerometer measurements and in turn corrupt the bias estimates. For instance, the sensor suite is attached to a vest, which is attached to the back of the canine. However, the vest is not fixed rigidly there. There is relative motion between the vest and the canine. The vest essentially moves with the canine’s skin, which is not always moving precisely in sync with the rest of the dog. In the same way that you can move the skin on your arm relative to the bones beneath, a dog’s skin moves also relative to its back, and even more so than our skin relative to our bones, especially considering the weight of its skin and the vest. This causes pack jarring to occur as the canine moves. If the canine pushes off the ground with its front paws at the same instance that the pack is moving forward, there will be a subsequent collision and jarring of the XSens unit. This happens often and results in very high accelerometer measurement spikes—sometimes over 20 m/s^2 . As noted above, such jarring saturates the accelerometers and causes a lumping of error in the accelerometer bias estimate.

The effect of pitch in canine motion can also be significant and will be discussed later in this chapter. When pitch is present in the canine’s motion, the longitudinal accelerometer measurement will be significantly affected and cause further error in the bias estimates.

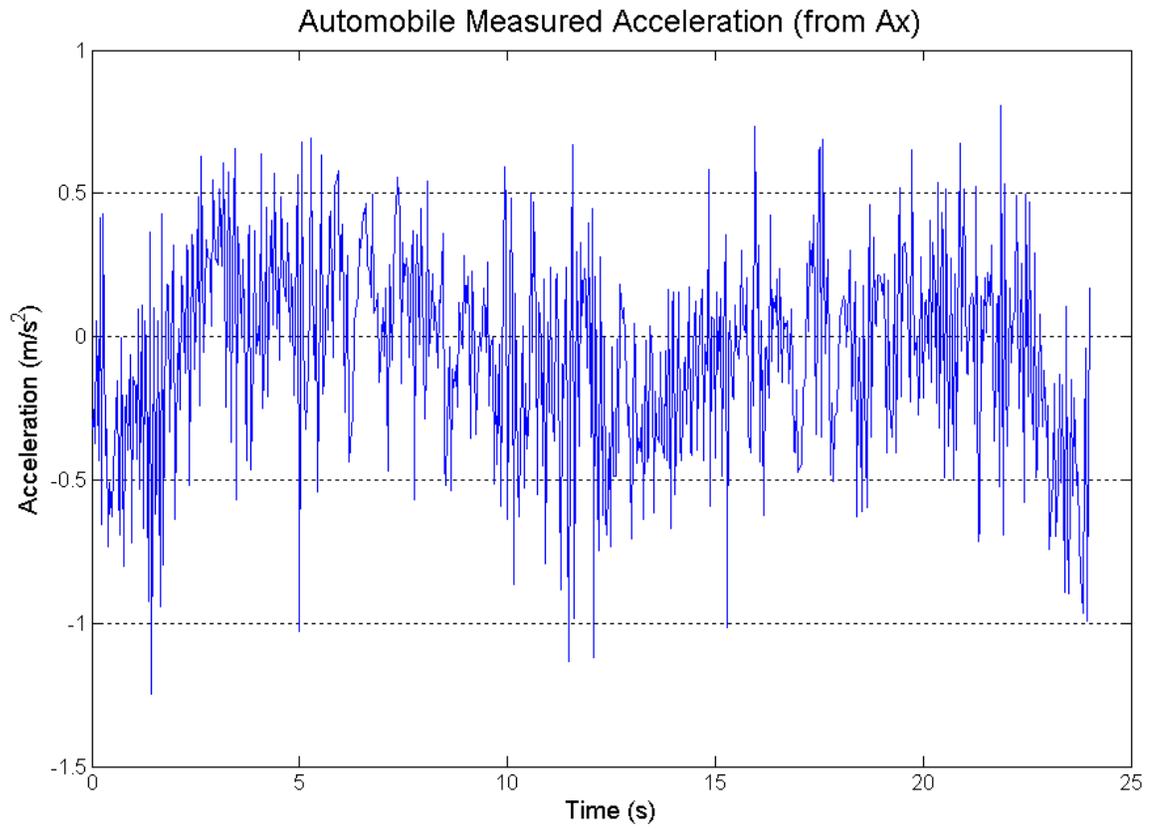


Figure B.3. Raw accelerometer measurements in automotive applications.

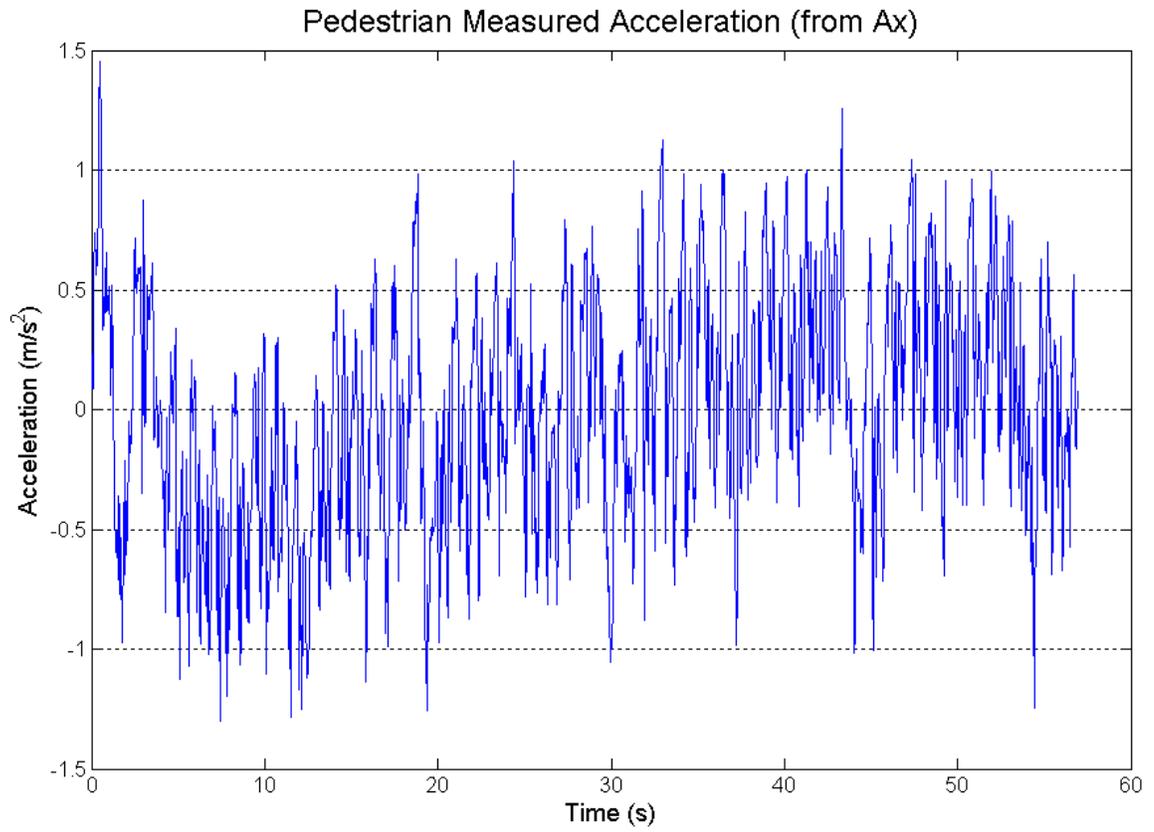


Figure B.4. Raw accelerometer measurements in pedestrian applications.

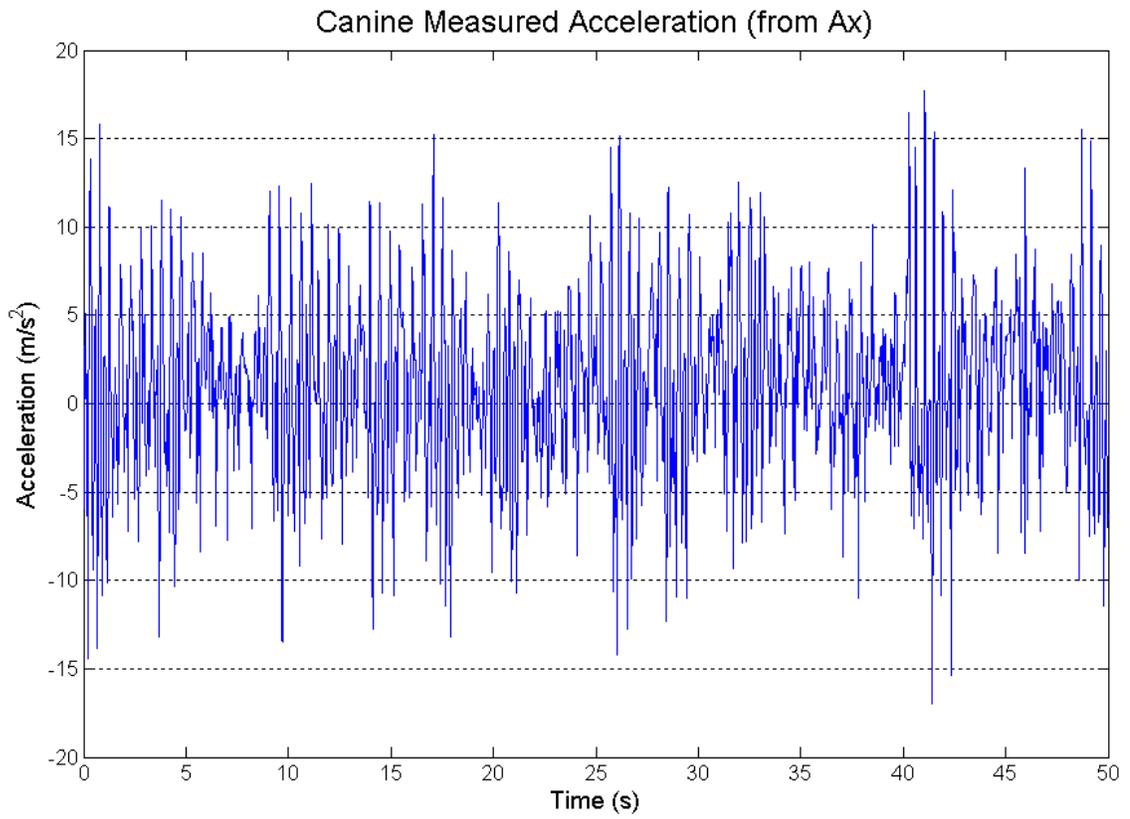


Figure B.5. Raw accelerometer measurements in canine applications.

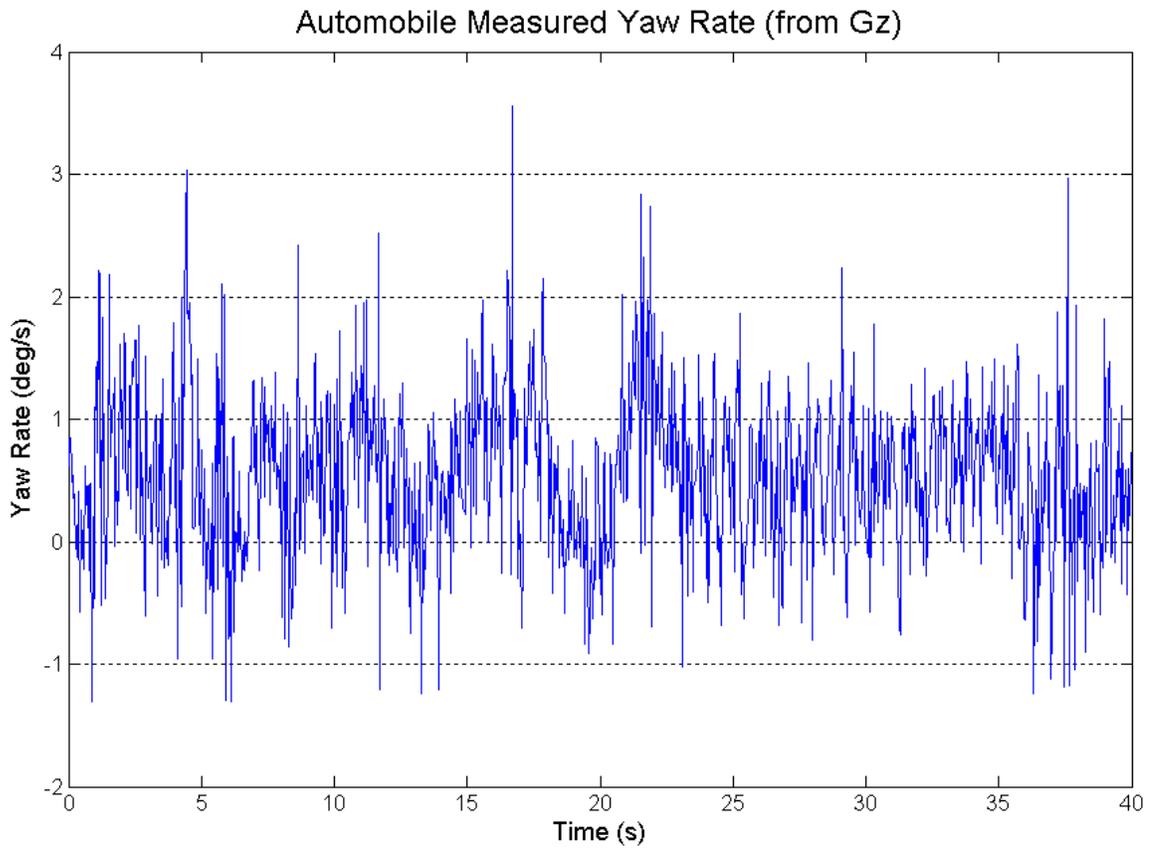


Figure B.6. Raw yaw rate measured from the IMU gyroscope in automotive applications.

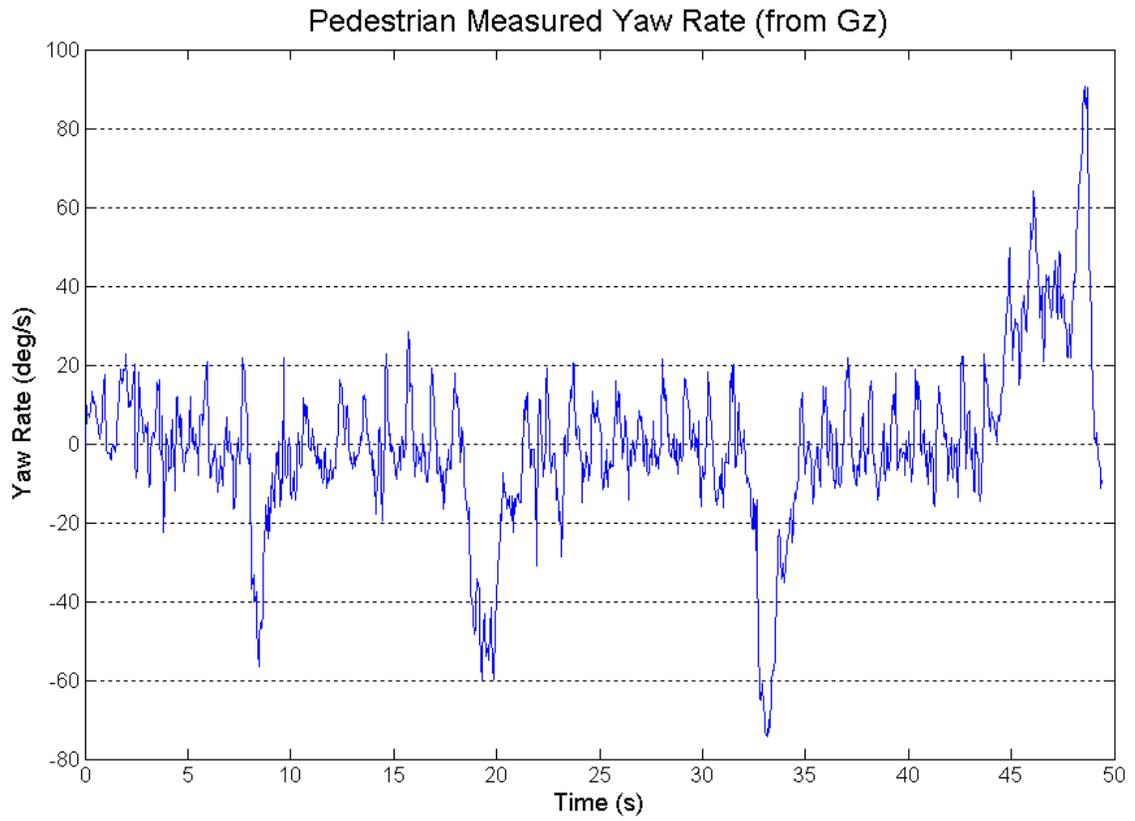


Figure B.7. Raw yaw rate measured from the IMU gyroscope in pedestrian applications.

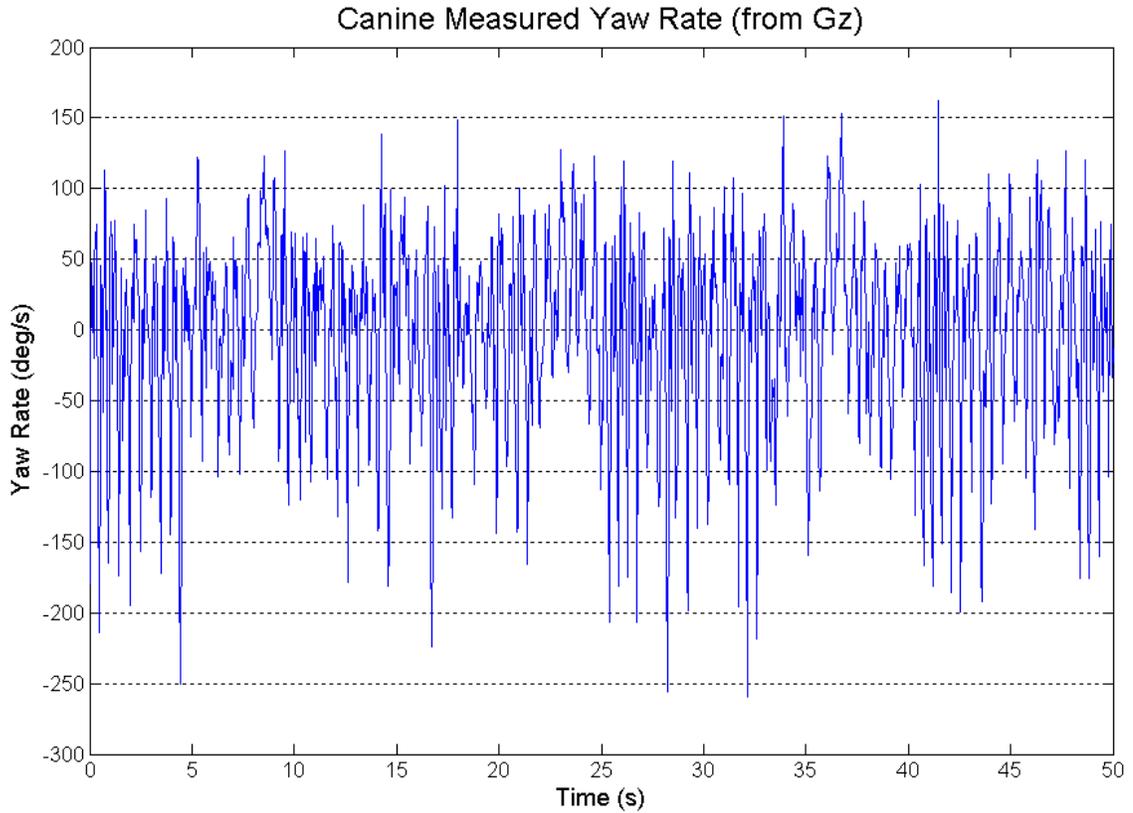


Figure B.8. Raw yaw rate measured from the IMU gyroscope in canine applications.

B.5 Experimental Results

Figures B.9-B.11 illustrate the results acquired when the canine is instructed to travel north, turn and travel east, U-turn and travel west, and then turn and travel south. Figure B.9 illustrates measured GPS velocity, EKF estimated velocity, and the velocity acquired from pure dead reckoning of the IMU without sensor biases taken into account. The pure dead reckoning velocity profile (green) drifts away from the GPS measurement and EKF estimate immediately and significantly. During the first 15-20 seconds of the trial, a “saw tooth” appearance is present in the EKF estimate while the filter is

approximating the inertial sensor biases. After approximately 20 seconds, the EKF estimate tracks the velocity more smoothly.

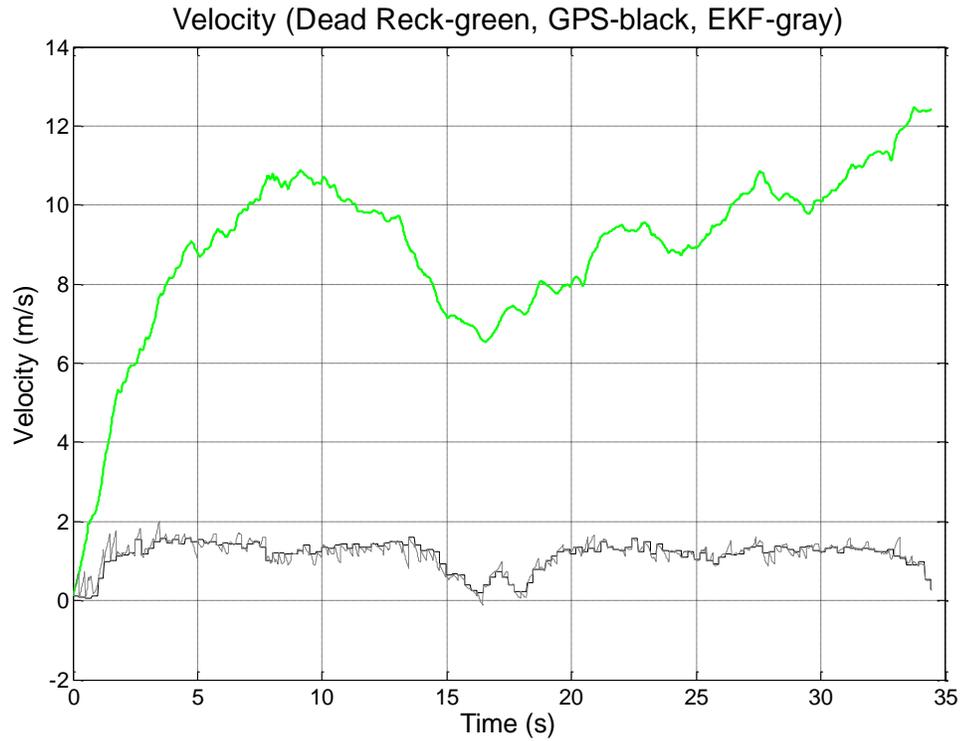


Figure B.9. Velocity profile for GPS measurements versus EKF estimate versus dead reckoning when the canine travels north, turns and travels east, U-turns and travels west, and then turns and travels south.

The EKF position estimate for the same trial (see Figure B.10) adheres to the GPS measurements, then diverges when the dog velocity approaches zero meters per second at the U-turn around 16-18 seconds. When approaching zero meters per second, the GPS course measurements become inaccurate, thus corrupting the bias estimates. Also, the filter has not had sufficient time to accurately estimate the inertial sensor biases by that

time. Then, when the canine velocity increases, the positions merge again towards the end of the test.

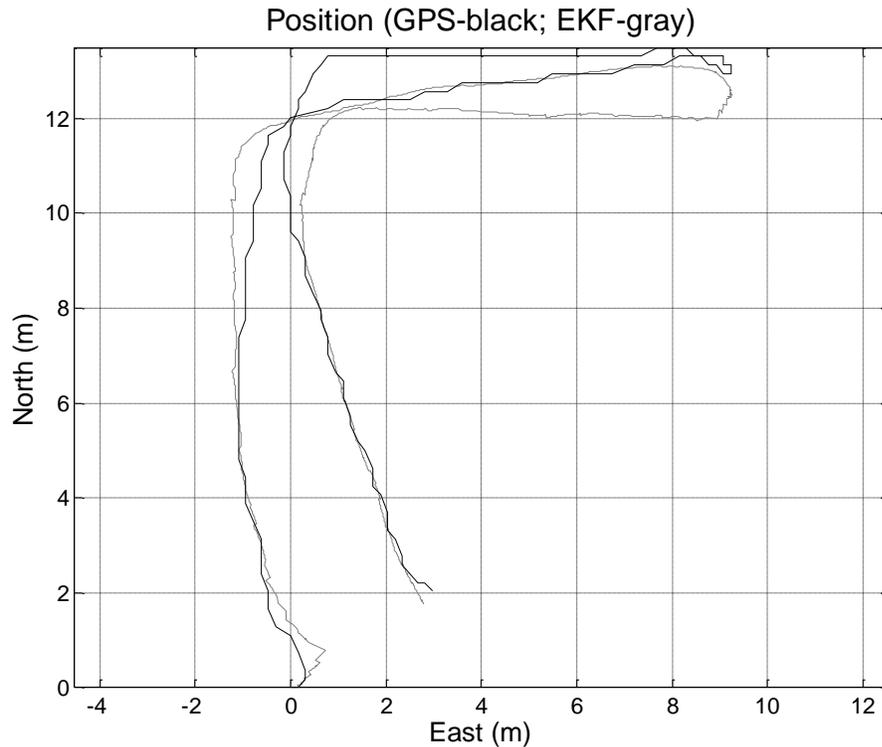


Figure B.10. GPS measurement versus EKF estimated position results when the canine travels north, turns and travels east, U-turns and travels west, and then turns and travels south.

As seen in Figure B.11, it is evident that the inertial sensor bias estimates level out over time—especially the gyroscope bias estimate—and the velocity and heading track the GPS very well. Again, the heading drifts slightly when the canine velocity approaches zero meters per second at the U-turn (see Figure B.10).

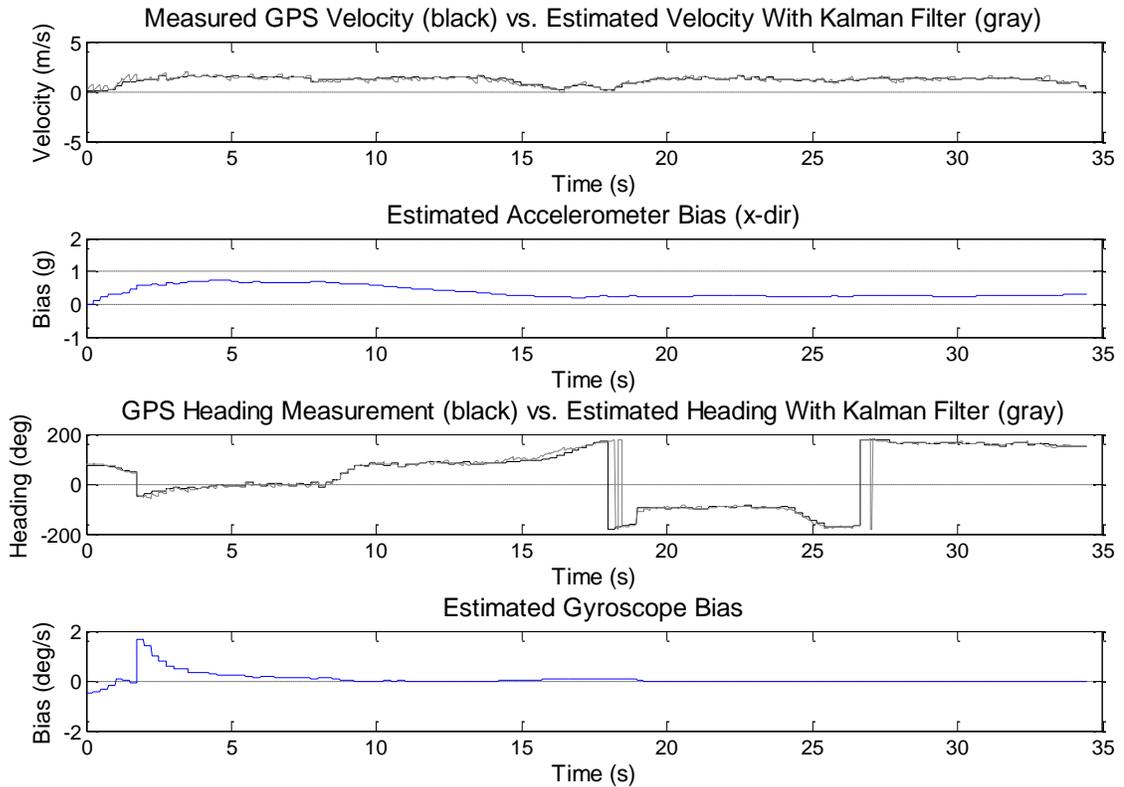


Figure B.11. Velocity, accelerometer bias, heading, and gyroscope bias EKF estimate results when the canine travels north, turns and travels east, U-turns and travels west, and then turns and travels south.

Figures B.12 and B.13 illustrate the results acquired when the dog is instructed to travel north, turn and travel east, turn and travel south, and then turn and travel west to the approximate starting position. For this illustration, GPS is artificially removed around 15 seconds to demonstrate loss of GPS and simulate pure dead reckoning. The EKF estimate at this point is based on the IMU alone, plus the last bias estimate acquired before GPS was turned off. The EKF estimated velocity profile shows the commencement of a drift away from the actual GPS measurements towards the end of the test (see Figure B.12). This is partially due to the relatively small length of time the filter

had to estimate the accelerometer bias and the number and geometry of the GPS satellites in view. However, it is believed that this is also due to unmodeled canine motion characteristics which are corrupting the accelerometer bias estimate. This will be addressed in later appendices and is discussed in Chapter 3. Notice that although the velocity drifts, the heading estimate continues to stay close to the actual GPS course measurements, since the gyroscope bias was sufficiently estimated by the EKF before the outage.

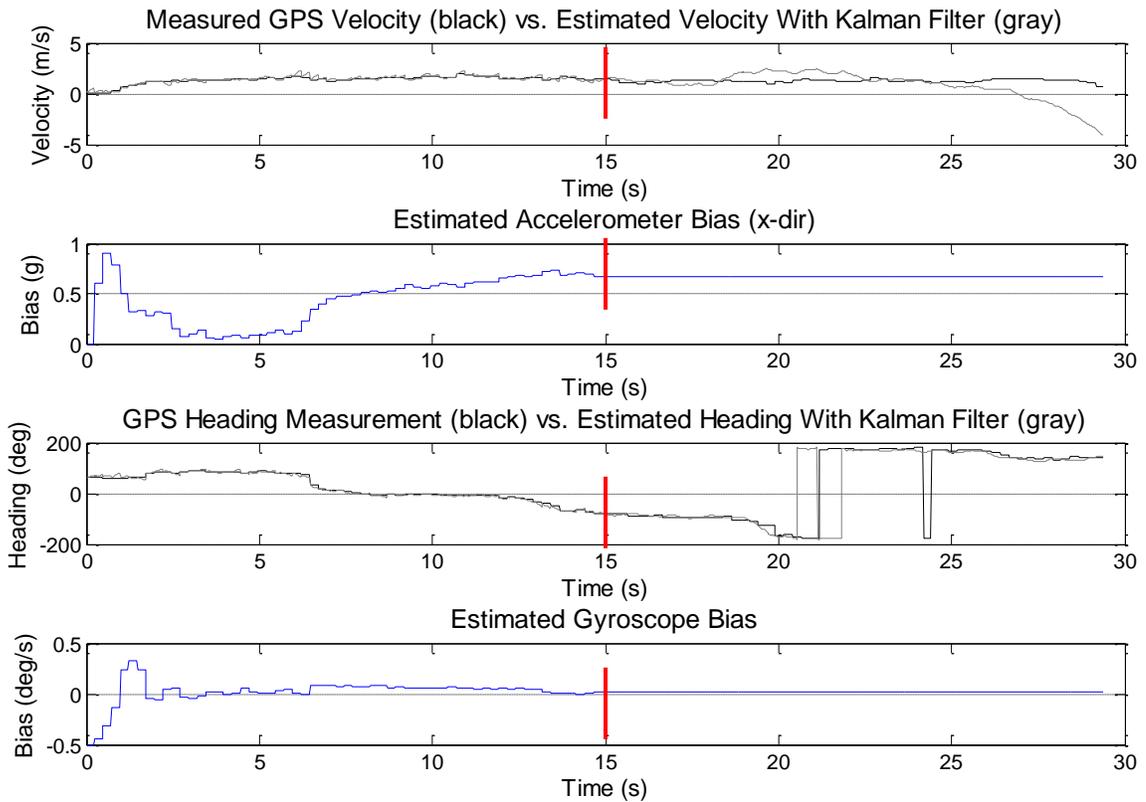


Figure B.12. Velocity, accelerometer bias, heading, and gyroscope bias EKF estimate results when the canine travels north, turns and travels east, turns and travels south, and then turns and travels west. GPS measurements are turned off in the EKF estimate at approximately 15 seconds (marked with a red vertical line).

The position plot (Figure B.13) shows the measured GPS position and EKF estimated position of the canine during this trial. The EKF estimate drifts away from the GPS measurements towards the end of the test. This would be expected during the simulated GPS outage since the accelerometer bias had not been estimated effectively by the outage. The accelerometer bias term significantly affects the velocity estimates, which in turn, according to Equation (B.5), affects the position estimate.

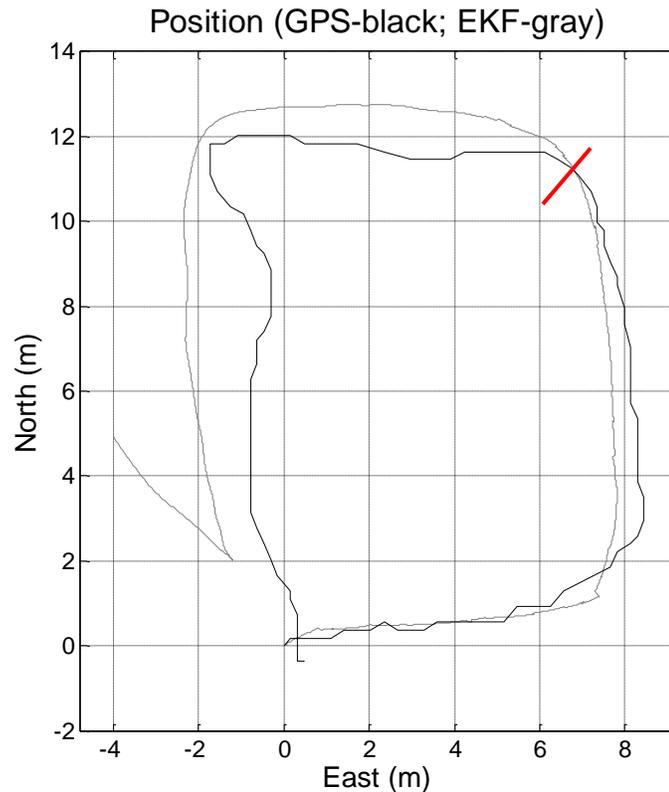


Figure B.13. GPS measurement versus EKF estimated position results when the canine travels north, turns and travels east, turns and travels south, and then turns and travels west. GPS measurements are turned off in the EKF estimate at approximately 15 seconds (marked with a red line).

B.6 Conclusions

This preliminary phase of the tracking algorithm development showed that GPS/INS integration can be utilized effectively to achieve adequate position, velocity, and orientation results with the Extended Kalman Filter, in spite of the unique motion characteristics of a canine. However, it was also clear that standard GPS/INS integration models do not adequately account for all of the significant canine behavioral characteristics. Unmodeled motion characteristics corrupted the EKF bias estimates, especially for the accelerometer, which will be addressed later in the chapter. In spite of these issues, the canine's estimated position for the above trial was still within odor detecting range for a canine after a 15 second GPS outage.

Appendix C

Canine Heading Estimation Improvements using Magnetometers

C.1 Introduction

In this phase of the navigation algorithm developments (as reported in [70]), the author sought to improve the heading estimation portion of the tracking algorithms using magnetometers. Unlike automobiles, canines stop and go frequently. When the canine stops, GPS course measurements become unstable, thus corrupting the estimate of the gyroscope bias. However, magnetometers can be used to offset this corrupting influence.

C.2 Guidance System

The guidance system for this phase is comprised of the hardware discussed in Chapter 2. Of the available measurements, the accelerometer measuring acceleration in the longitudinal direction, the gyroscope measuring yaw rate, and the magnetometers measuring the magnetic field strength in the east and north directions were used. As in the previous section, by combining the measurements from the GPS receiver,

accelerometer, gyroscope, and magnetometers using the Extended Kalman Filter, it was determined that the navigation solution could be improved upon.

C.3 Canine Testing

In order to provide qualitative measures of how effective the EKF is in tracking the canine orientation, the dog was walked in box patterns to illustrate typical canine motions. Using these tests, the EKF was tuned to reject some of the typical disturbances created by a dog's sporadic movements, but still maintain an acceptable level of navigational performance. Videos were also made to assist in checking for error. For comparison, similar tests were run using pedestrians.

C.4 Improvements

In order to improve the heading estimates during a GPS outage, improvements were made to the algorithms discussed in [69] and were utilized in acquiring the estimates. First, the sensor model was adjusted to include pre-calibration of the raw accelerometer and gyroscope data in the form of a minor scale factor, which helped to compensate for unmodeled factors in the canine's motion. Equations (C.1) and (C.2) give the new state equations for acceleration and yaw rate as used in this phase.

$$\hat{V} = SF_a A_x^m - \hat{b}_a \tag{C.1}$$

$$\hat{\psi} = SF_g g_z^m - \hat{b}_g \tag{C.2}$$

- Where: \hat{v} = Acceleration estimate
- SF_a = Accelerometer scale factor (0.95)
- A_x^m = Measured longitudinal acceleration from the accelerometer
- \hat{b}_a = Accelerometer bias estimate
- $\hat{\psi}$ = Yaw rate estimate
- SF_g = Gyroscope scale factor (0.9)
- g_z^m = Measured yaw rate from the gyroscope measuring motion about the z-axis
- \hat{b}_g = Gyroscope bias estimate

As before, the biases for this system are modeled as random walks, and the process noise is assumed to be zero mean white noise (i.e., $\omega \sim N(0, \sigma_\omega^2)$). Thus, Equation (B.4) holds.

Secondly, magnetometers were added to the sensor suite. As verified in pedestrian applications [72], magnetometers coupled with gyroscopes can provide improved heading results during GPS outages, especially since magnetometers continue to yield heading measurements during GPS outages. Magnetometers measure the strength of the magnetic field in a given direction. The XSens IMU contains magnetometers oriented in a tri-axis formation. Under the assumption that there is no change in pitch in the canine's motion (an assumption that will be discussed further in the chapter), the following holds true.

$$\psi_{MagM} = \tan^{-1} \frac{M_N^m}{M_E^m} \quad (C.3)$$

Where: ψ_{MagM} = Measured heading from longitudinal and lateral magnetometers

M_N^m = Measured strength of the magnetic field in the longitudinal direction

M_E^m = Measured strength of the magnetic field in the lateral direction

Although extensive care must be taken in magnetometer calibration in many applications due to severe biasing caused from the presence of ferrous materials in the immediate vicinity of the magnetometer location (e.g., attachment to robots and vehicles) [73-75], applying the use of magnetometers with canines proved to bypass much of this inherent biasing problem.

Finally, as discussed in the previous section, the unique motion characteristics of living biped and quadruped species, such as canines, require specialized tuning of the Process Noise Covariance (Q) Matrix in the EKF. These parameters were re-tuned for this phase after the implementation of the magnetometers to further improve the heading estimates during a GPS outage.

C.5 Tracking Algorithms

In this system, the longitudinal acceleration is measured by an accelerometer, and the yaw rate is measured by a gyroscope. The velocity, course, north, and east positions are measured by the GPS receiver. These measurements are all integrated by the tracking algorithm. As explained in the previous section, according to Equation (3.4), the new output equation can be reduced to the following:

$$y = \begin{bmatrix} V_{GPSM} \\ \psi_{GPSM} \\ N_{GPSM} \\ E_{GPSM} \\ \psi_{MagM} \end{bmatrix} \quad (C.4)$$

The state estimate vector for this system is:

$$\hat{x} = [\hat{V} \quad \hat{b}_a \quad \hat{\psi} \quad \hat{b}_g \quad \hat{N} \quad \hat{E} \quad \hat{b}_m]^T \quad (C.5)$$

Where: \hat{V} = Estimated velocity

\hat{b}_a = Estimated accelerometer bias

$\hat{\psi}$ = Estimated course

\hat{b}_g = Estimated gyroscope bias

\hat{N} = Estimated northern position

\hat{E} = Estimated eastern position

\hat{b}_m = Estimated magnetometer bias

Figure B.2 still holds as a visual illustration of the notations used in determining northern and eastern velocities. Side slip (β) is assumed to be zero as well. Therefore, GPS course is still assumed to be equal to heading (ψ) since course is equal to $(\psi + \beta)$.

After the appropriate linearizations (discussed previously), the following Jacobian matrix results for this system:

$$\Phi = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \cos \hat{\psi} & 0 & -\hat{V} \sin \hat{\psi} & 0 & 0 & 0 & 0 \\ \sin \hat{\psi} & 0 & \hat{V} \cos \hat{\psi} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (C.6)$$

Equations (B.5) and (B.6) still hold from Appendix B and will not be repeated here. As before, the GPS Sensor Noise Covariance Matrix (R) and the Process Noise Covariance Matrix (Q) constitute the primary EKF tuning parameters. The R matrix values are relatively standard and can be acquired from the sensor information.

$$R = \begin{bmatrix} \sigma_{GPSvel}^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{GPShead}^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{GPSnorth}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{GPSeast}^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{MagHead}^2 \end{bmatrix} \quad (C.7)$$

$$Q_k = \begin{bmatrix} \sigma_A^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_{bA}^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_g^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{bg}^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{north}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{east}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{bm}^2 \end{bmatrix} \quad (C.8)$$

The Sensor Noise Covariance Matrix values used for both the pedestrian as well as the canine data are as follows.

$$\sigma_{GPSvel} = 0.1 \frac{m}{s} \quad (C.9)$$

$$\sigma_{GPSnorth} = 0.01 m$$

$$\sigma_{GPSeast} = 0.01 m$$

$$\sigma_{GPShead} = \frac{0.1}{V_{GPSM}} \text{ radians}$$

$$\sigma_{MagHead} = 10 \text{ radians (Canine)}$$

$$= 0.1 \text{ radians (Pedestrian)}$$

The Process Noise Covariance Matrix (Q) values once again proved to be unique for pedestrian and canine applications (i.e., when compared to vehicle data) under the current model.

$$\sigma_{bg} = 1 * e^{-6} \frac{m}{s^2} \quad (C.10)$$

$$\sigma_g = 0.08 \frac{rad}{s}$$

$$\sigma_{north} = \sigma_{east} = 1 * e^{-5} m$$

The other values used in the Q matrix proved to be different for pedestrian versus canine data (see Table C.1). The noise on the longitudinal accelerometer for the canine is significantly higher to account for more unmodeled effects.

Table C.1. Various Process Noise Covariance Matrix (Q) values for pedestrian and canine applications.

	Pedestrian	Canine
σ_A	$175 \frac{m}{s^2}$	$550 \frac{m}{s^2}$
σ_{bA}	$25 \frac{m}{s^2}$	$8 \frac{m}{s^2}$
σ_{bm}	0.001 rad	0.1 rad

Note that the determined values of the noise characteristics on the accelerometer and the gyroscope illustrated here were based upon the particular pedestrian and canine being tested. Changing the person/canine being tested could cause different results due to unmodeled effects in the current, simple dynamic model. Tests where sprinting occurred also yielded different tuning results. A model for coping with such variation will be discussed later in this chapter.

The state estimation covariance matrix was initialized as follows for the canine.

$$P = \begin{bmatrix} \sigma_{GPSvel}^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10000 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{GPSnorth}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{GPSeast}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{bm}^2 \end{bmatrix} \quad (C.11)$$

The pedestrian state estimation covariance matrix was the same, except for the initial confidence in the accelerometer bias, which was set to 1 and the confidence in the accuracy of the heading, which was set to 1000.

C.6 Experimental Results

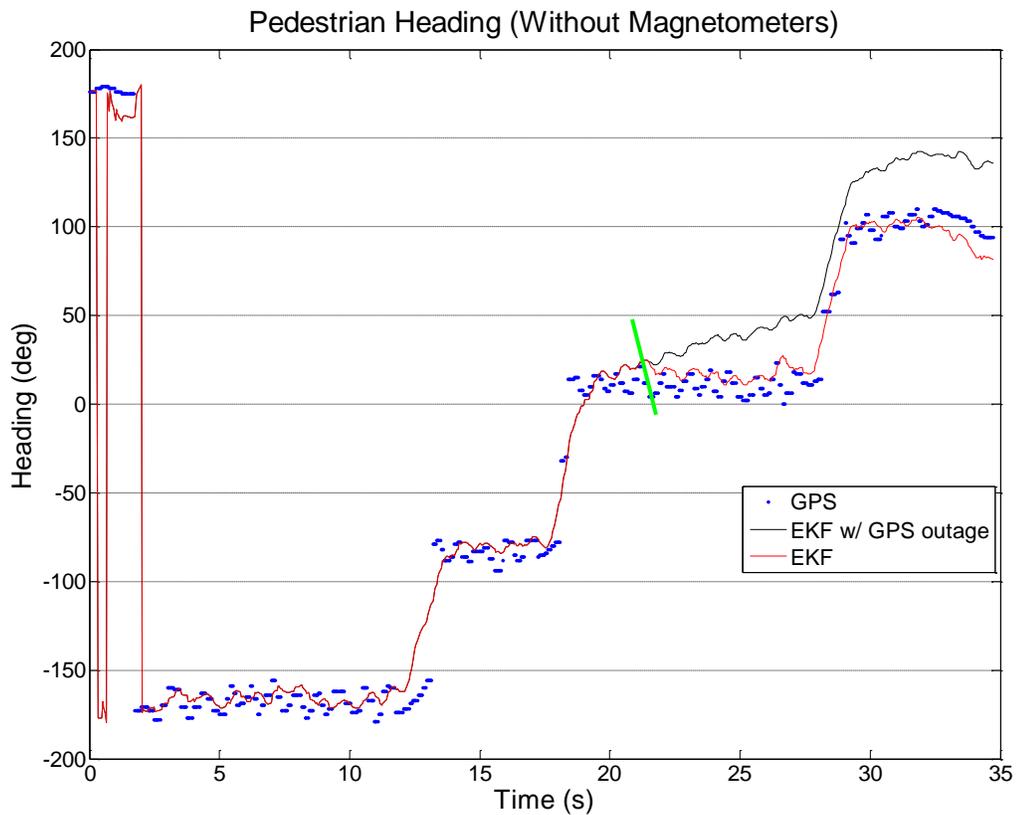


Figure C.1. EKF pedestrian heading results when magnetometers are not used.

Simulation GPS outage occurs at approximately the green line.

Figure C.1 illustrates typical heading results of the EKF for a **pedestrian** scenario when magnetometers are not present. The blue dots represent the GPS measurements.

The red line represents the EKF position estimate. The black line represents the EKF position results when a simulated GPS outage occurs approximately $\frac{1}{2}$ of the way through the test. [Note that heading wrapping occurs during the first two seconds of the trial since the heading is set to stay between ± 180 degrees.] Notice that when the GPS outage occurs, the heading estimate immediately begins to drift in comparison to the EKF and GPS course measurements.

Figure C.2 illustrates the heading results of the EKF for a pedestrian scenario when magnetometers are used. Again, a GPS outage is simulated $\frac{1}{2}$ way through the trial. For both cases (i.e., with and without magnetometers), for the sake of comparison, the tuning parameters are kept constant, and the magnetometer tuning parameters are merely added to the relevant matrices. [Note again that heading wrapping occurs during the first two seconds of the trial since the heading is set to stay between ± 180 degrees.] Notice that when magnetometers are in use, the heading estimate post-GPS outage is visibly closer to the EKF and GPS course readings. Comparing the mean differences between the EKF heading estimate (with the GPS outage in place) and GPS course measurements for this test revealed a 10.2% improvement in the pedestrian heading estimate when magnetometers are in use.

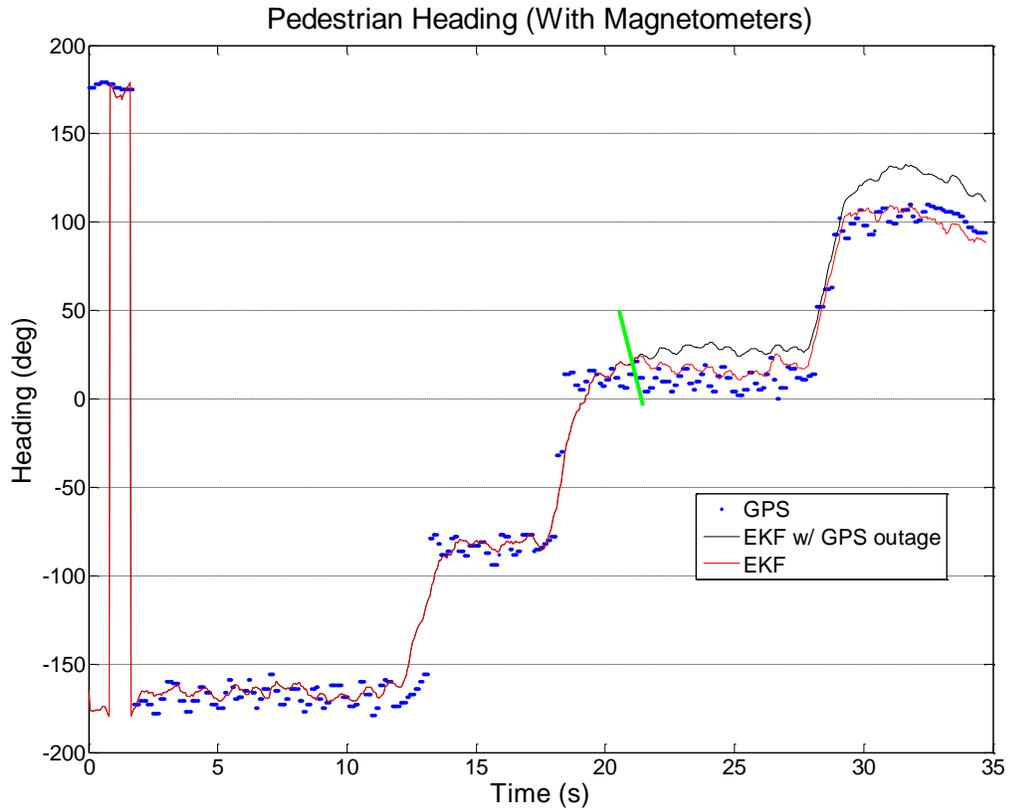


Figure C.2. EKF pedestrian heading results when magnetometers are used. Simulation GPS outage occurs at approximately the green line.

Figure C.3 illustrates the typical EKF estimate of the **canine** heading when magnetometers are not used in the algorithms. Notice that after the GPS outage is simulated, there is an immediate drifting of the heading estimate from the EKF solution and GPS course measurements.

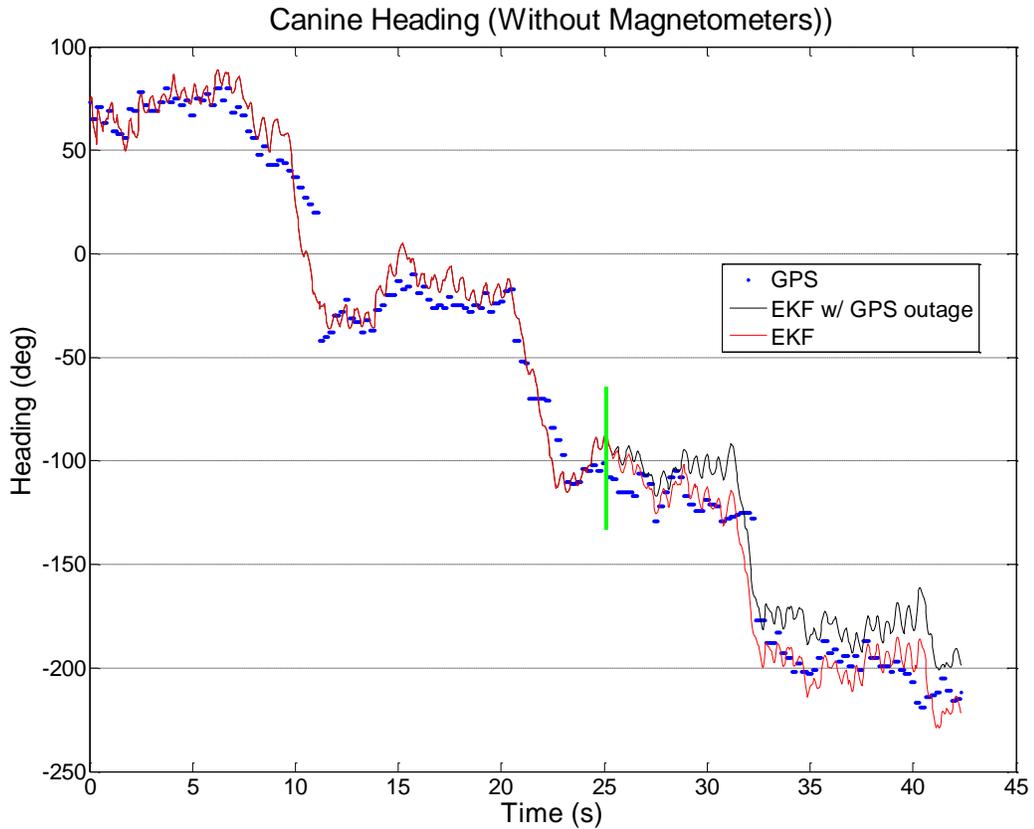


Figure C.3. EKF canine heading results when magnetometers are not used. Simulation GPS outage occurs at approximately the green line.

Figure C.4 illustrates the EKF heading estimates for a canine when magnetometers are present. Again, the corresponding tuning parameters are kept constant and the magnetometer tuning parameters are merely added to the relevant matrices. As in the previous examples, the black line represents the EKF position estimate after a simulated GPS outage $\frac{1}{2}$ of the way through the test. A drift from the EKF solution and GPS course measurements is apparent again. However, a comparison of the mean difference between the EKF heading estimate (with the GPS outage in place) and GPS

course measurements for this test revealed a 6.3% improvement in the heading estimate when magnetometers are used.

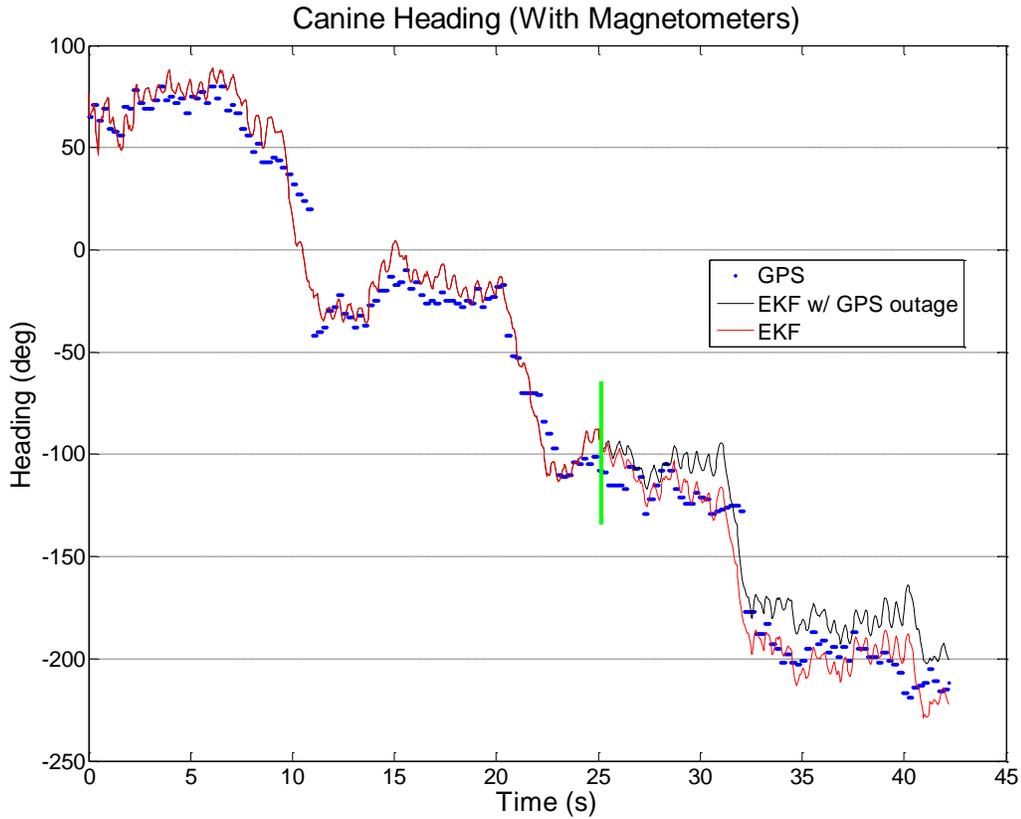


Figure C.4. EKF canine heading estimate when magnetometers are used. Simulation GPS outage occurs at approximately the green line.

C.7 Conclusions

Although extensive testing considering many different variables would be required to determine the extent to which magnetometers improve the heading estimate during GPS outages, it is clear that when magnetometers are used to bolster the heading estimate, the results are improved during a GPS outage. Magnetometers keep the heading

estimate from diverging from the EKF and GPS course measurements as quickly. This confirms the results stated in [72]. An optimized and more reliable system can be obtained by coupling gyroscopes and magnetometers in obtaining the pedestrian heading estimate during GPS outages. This applies to canine tracking scenarios as well.

This phase of the tracking algorithm development illustrated that GPS/INS/Magnetometer integration using an EKF can be utilized effectively to achieve improved orientation results over GPS/INS integration alone for some canine applications. However, the motion characteristics of a canine inject significant error into the tracking estimates. Whereas in automotive situations many unmodeled factors can be neglected due to, for instance, minimal slip and stationary sensors, canine motion contains significant slip, incline, and jostle that affects estimates. The assumption that GPS course measurements can be integrated with the yaw rate gyro (i.e., that slip can be neglected) injects error into the estimates, as well as the assumption that there is no incline in the motion of the canine (i.e., gravity is neglected). The velocity estimate diverges significantly under the current model, which results in undesirable position error during GPS outages.

Areas where there is a high ferrous content would temporarily bias the magnetometers. Thus it may be necessary in future work to compensate for the magnetometer bias that would occur when the canine enters into areas with a high ferrous content. Magnetometer calibration would be necessary in such scenarios. [72] demonstrated that the heading derived from the gyroscope “can be used to identify magnetic disturbances, while the magnetic compass can determine the bias of the gyro and the initial orientation.”

Appendix D

Canine Position and Velocity Estimate Improvements by Accounting for Pitch in a Canine's Motion

D.1 Introduction

As discussed in Appendix C and as presented in [71], the standard modeling equations that are used in an EKF for, for instance, automotive applications, are not sufficient to account for all of the motion characteristics of a canine. The result is that more error is lumped into the estimate of the inertial sensor biases by the EKF, and the Process Noise Covariance Matrix must be tuned to account for these unmodeled factors. The goal of this phase of the tracking algorithm development is to determine how significant the affect of pitch is in canine motion, and if it is found to be significant, remove its effects in the estimate of the longitudinal accelerometer's bias estimate by including pitch in the canine motion model. This will, in turn, cause less error to be lumped into the accelerometer bias estimate, which will allow for a quicker and more accurate estimate of the bias. When the accelerometer bias is estimated better, a better position and velocity estimate of the canine can be attained during GPS outages.

Also, as was mentioned briefly in Chapter 2, the small, low-cost, embedded microprocessor has limitations that a standard lap top computer does not have.

Computational expense must be considered in the development of the filtering algorithms. Thus, a second goal of this phase is to determine a filtering approach that will be computationally inexpensive enough to run onboard the microprocessor side-by-side with the controller that was discussed in Chapter 4.

D.2 Guidance System

The navigation system described in Chapter 2 is used for this phase of the tracking algorithm development. North and East positions, as well as Velocity measurements were used from the GPS receiver. The longitudinal accelerometer measurement and pitch and heading estimates from the XSens unit are used as inputs to the filtering algorithms.

D.3 Canine Testing

In order to capture the effect of pitch in canine motion, data was captured while the canine was being walked by a handler. Walking motion closely approximates canine motion while it is in odor detection mode. It is this type of motion that is most likely to be occurring during GPS outages, since the surrounding environment could limit satellite visibility in such scenarios. This observation made canine walking motion of special interest during this project.

D.4 Tracking Algorithms

As has been previously discussed, accurate estimates of canine position, velocity, and orientation are needed for the canine control problem. That tracking solution needs to

be able to hold during brief GPS outages, and it needs to be as computationally inexpensive as possible so that the algorithms will be able to run real-time on the small, low-cost microprocessor located onboard the canine alongside the control algorithms. Using an EKF in this project has proved to be effective in yielding a navigation solution that will hold during brief GPS outages and is used during this phase as well. In order to meet the second goal, a loosely-coupled, four state EKF is used for the tracking algorithm, and for this stage of the project the Attitude Heading Reference System on the XSens unit is used to provide a drift-free measurement of the Euler angles. This allows for an accurate estimate of heading and pitch, regardless of the presence of GPS satellite visibility and without adding several extra estimated states to the EKF, thus significantly lowering the computational expense of the algorithms. These choices proved to meet the goals stated above.

The four state estimate vector, initialized with zeros, is the following:

$$\hat{x} = \begin{bmatrix} \hat{V} & \hat{b}_a & \hat{N} & \hat{E} \end{bmatrix}^T \quad (\text{D.1})$$

Where: \hat{V} = Estimated velocity

\hat{b}_a = Estimated accelerometer bias

\hat{N} = Estimated northern position

\hat{E} = Estimated eastern position

The state equation for acceleration is illustrated in Equation (D.2) and will be explained in further detail later.

$$\hat{V} = A_x^m - g \sin(\theta) - \hat{b}_a \quad (\text{D.2})$$

Where: \hat{V} = Acceleration estimate

A_x^m = Measured longitudinal acceleration from the accelerometer

\hat{b}_a = Accelerometer bias estimate

θ = Pitch measurement

g = Acceleration due to gravity ($9.81 \frac{m}{s^2}$)

Equation (B.5) still holds as the state equation for northern and eastern velocities. Also as before, the bias for this system is modeled as a random walk, and the process noise is assumed to be zero mean white noise (i.e., $\omega \sim N(0, \sigma_\omega^2)$). Thus, Equation (B.4) holds.

After the appropriate linearizations discussed above, the following Jacobian matrix results for this system:

$$\Phi = \begin{bmatrix} 0 & 0 & 0 & -1 \\ \cos \hat{\psi} & 0 & 0 & 0 \\ \sin \hat{\psi} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (\text{D.3})$$

As before, the longitudinal acceleration is measured by a longitudinal accelerometer, and pitch and heading estimates are provided by the XSens unit. Thus, the input vector is as follows:

$$u = \begin{bmatrix} A_x^m \\ \theta_m \\ \psi_m \end{bmatrix} \quad (D.4)$$

Where: $\theta_m =$ Pitch measurement

$\psi_m =$ Heading measurement

Velocity (i.e., ground speed), northern position, and eastern position measurements are provided by GPS. Thus, the output vector, according to Equation (3.4), is as follows:

$$y = \begin{bmatrix} V_{GPSM} \\ N_{GPSM} \\ E_{GPSM} \end{bmatrix} \quad (D.5)$$

As before, the GPS Sensor Noise Covariance Matrix (R) and the Process Noise Covariance Matrix (Q) constitute the primary EKF tuning parameters. The R matrix values are relatively standard and can be acquired from sensor information.

$$R = \begin{bmatrix} \sigma_{GPSvel}^2 & 0 & 0 \\ 0 & \sigma_{GPSnorth}^2 & 0 \\ 0 & 0 & \sigma_{GPSeast}^2 \end{bmatrix} \quad (D.6)$$

$$Q_k = \begin{bmatrix} \sigma_A^2 * T_s & 0 & 0 & 0 \\ 0 & \sigma_{bA}^2 & 0 & 0 \\ 0 & 0 & \sigma_{north}^2 & 0 \\ 0 & 0 & 0 & \sigma_{east}^2 \end{bmatrix} \quad (D.7)$$

Where: $\sigma_{GPSvel} = 0.001 * S_{acc} \frac{m}{s}$

$$\sigma_{GPSnorth} = \sigma_{GPS east} = 0.0005 * H_{acc} m$$

$$\sigma_A = 0.1 \frac{m}{s^2}$$

$$\sigma_{bA} = 0.001 \frac{m}{s^2}$$

$$\sigma_{north} = \sigma_{east} = 1 * e^{-6} m$$

S_{acc} and H_{acc} are speed and position accuracy measurements, respectively, provided by the GPS receiver.

The state estimation covariance matrix was initialized as follows:

$$P = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (D.8)$$

D.5 Pitch Effects

Figures D.1 and D.2 illustrate typical raw pitch motion data that results from canine behavior. Figure D.1 is data from a trial where the canine is in odor detection

mode. Figure D.2 is data from a trial where the canine is sprinting. From Figure D.1, it is evident that canine odor detection motion exhibits nearly instantaneous 5-10 degree pitch fluctuations on average, and sometimes shifts in pitch of 15-18 degrees. From Figure D.2, it is evident that canine sprinting motion often exhibits magnitude changes of 40-60 degrees.

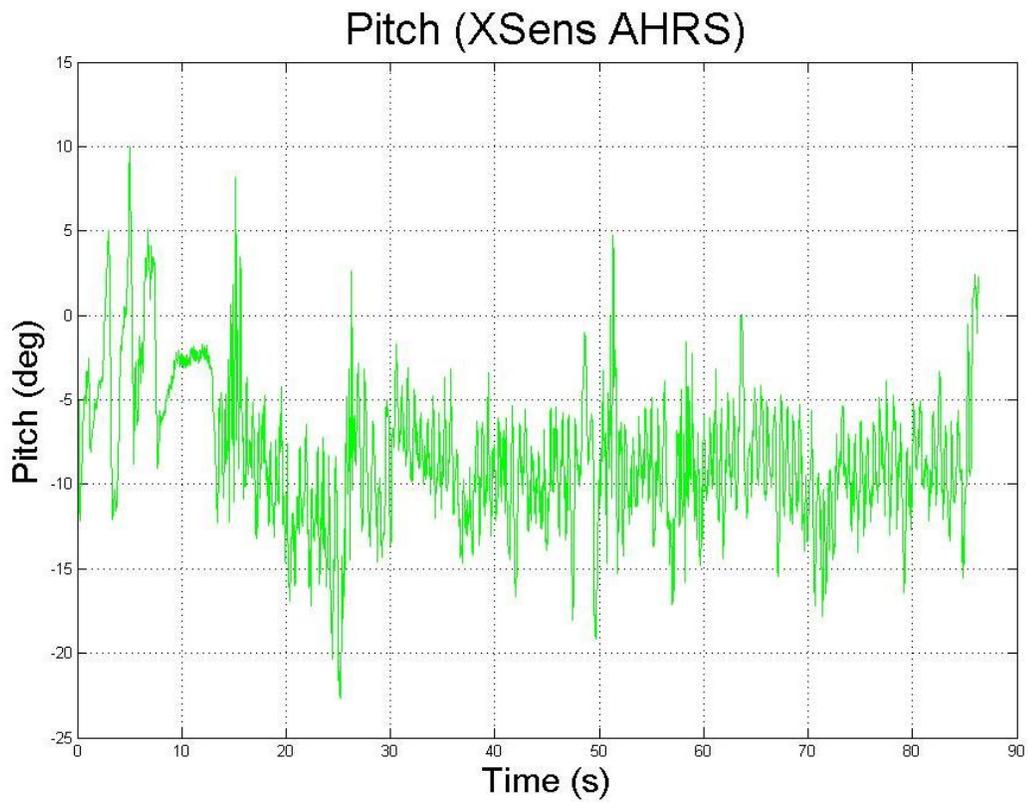


Figure D.1. Raw pitch measurement from XSens unit while canine is in odor detection mode.

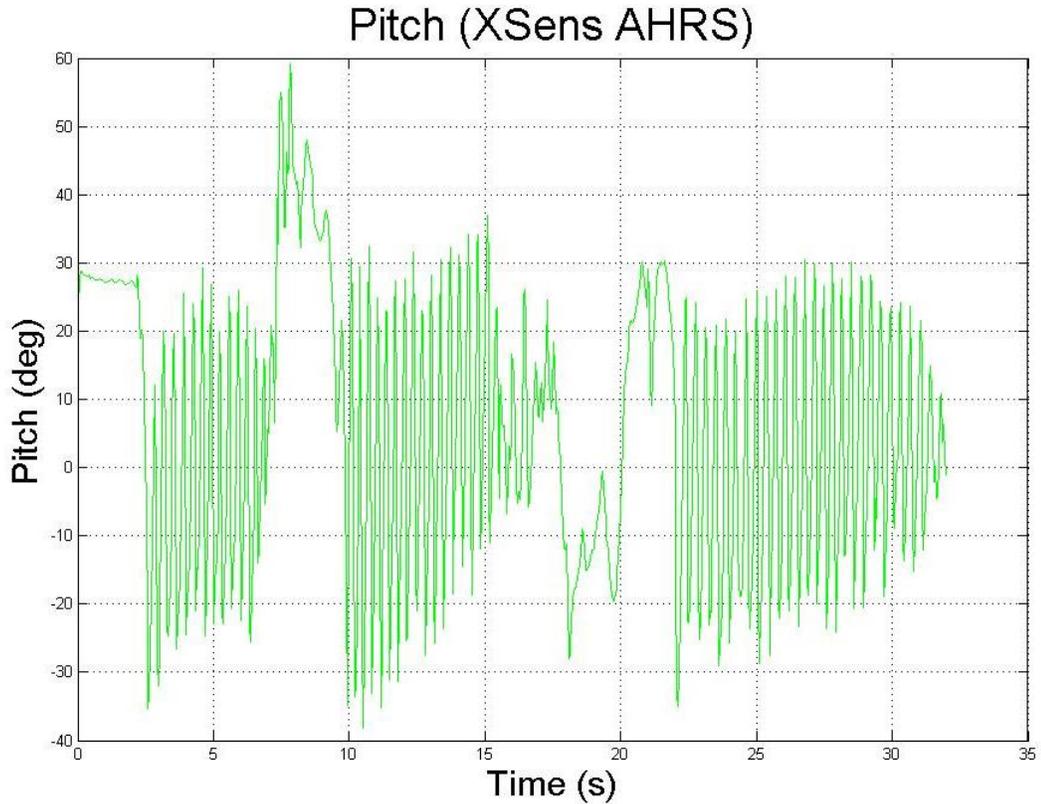


Figure D.2. Raw pitch measurement from XSens unit while canine is sprinting.

In vehicle applications, if the automobile is known to be on level ground (i.e., there is zero road grade), it would be expected that the average of the pitch measurements would be approximately zero degrees or very close to it, especially since care could be taken to ensure that the measuring devices are positioned in such a way on the automobile as to eliminate false inclination measurements. However, this is not the case in canine motion. An examination of Figure D.1 clearly shows that the average of the pitch measurements for the illustrated trial is not 0 degrees, but is rather -8 or -9 degrees.

There are several factors that can cause this phenomenon. For instance, if the canine spends a significant amount of time sitting, the XSens unit will register the presence of pitch or incline, which will affect the average pitch over time. Also, the

canine's spine is not level [77]. This results in the presence of pitch even when the canine is standing flat on level ground. Further, if the inertial sensors are not positioned precisely level on the back of the canine, a pitch can register. This scenario is common, since the vest and consequently, the sensor suite, unavoidably slide a few degrees to the side of the canine over time. Also, canine motion, in and of itself, inherently contains significant pitch behavior as well, since the back two legs do not move at the same time as the front two legs. This causes a "rocking horse"-like motion. Plus, if the canine's front or back legs are stronger than the others or push off the ground harder than the others, a pitch bias can register over time. Needless to say, in some applications, pitch can be neglected in the motion model. However, this is not the case with a canine scenario.

If perfectly level and motionless, a longitudinal accelerometer will read approximately 0 m/s^2 , plus some small bias and noise. However, the moment pitch is present, the accelerometer will begin to read the effect of gravity in its measurements and indicate that there is a longitudinal acceleration, even when the canine is sitting still. This is due to the fact that accelerometers measure all acceleration vectors, including gravity. In many typical vehicular motion models and in previous phases of this tracking algorithm evolution, the model did not take gravity into account and assumes that the integral of the accelerometer measurement in the longitudinal direction corresponds to the GPS measurement for velocity or ground speed (i.e., it did not consider the effect of incline or pitch). Thus, when incline was present, the longitudinal accelerometer would output a value due to the role of gravity, even when there was no motion present in the longitudinal direction. This significantly corrupted the accelerometer bias estimate, and in turn, the velocity estimate. This factor also deleteriously effected the canine position

estimate (which is also based on the velocity estimate—see Equation (B.5)). Recall from Figure B.5 that raw accelerometer data for canine motion often registers a significant bias. This is due, in part, to the presence of pitch in the canine motion and pack position. The effect of pitch in the level of accelerometer “noise” can be removed by accounting for pitch in the motion model, which is done for this phase of the tracking solution (see Equation (D.2)).

D.6 Canine Motion Simulator

D.6.1 Trial Profile

In order to visualize more specifically how the presence of pitch or incline affects an EKF canine tracking solution, a Canine Motion Simulator (CMS) was developed using MATLAB. An acceleration, yaw rate, and pitch profile were designed first. In the simulator, the canine starts its trial moving due north (0 degrees, yaw) on flat terrain (0 degrees, pitch) from a velocity of 0 m/s. It accelerates at 1 m/s² for three seconds, then changes heading for one second at a yaw rate of 90 deg/s. During the change in heading, a deceleration of -1 m/s² occurs, as would be expected to occur in a real world turning scenario. The canine then moves due east at a steady velocity of 3 m/s for 15 seconds. However, a 10 degree incline is present during this eastward movement. From this trial profile, the canine’s position, velocity, and yaw were determined at discrete times using the appropriate calculus equations, illustrated below.

$$V = \int A dt + V_o \tag{D.9}$$

Where: V = Velocity
 A = Acceleration
 V_o = Initial velocity

$$\psi = \int r dt + \psi_o \quad (D.10)$$

Where: ψ = Heading (yaw)
 r = Yaw rate
 ψ_o = Initial heading

From Equations (B.5), (D.9), and (D.10), the northern and eastern velocities were calculated. Then, as Equation (D.11) indicates, northern and eastern positions can be calculated by integrating northern and eastern velocities with respect to time.

$$N = \int \dot{N} dt + N_o \quad (D.11)$$

$$E = \int \dot{E} dt + E_o$$

D.6.2 Noise Characteristics

Using the trial profile information, the GPS measurements and inertial sensor measurements were developed. In order to mimic these sensor measurements, white noise was added to each of the measurements developed above using MATLAB's random

number generator function, $randn(\cdot)$. Biases were also added to the inertial sensors, as would be the case with actual inertial sensor measurements.

The GPS noise characteristics that were chosen are given in Equation (D.12).

$$\sigma_{GPS_{pos}} = 0.1m \quad (D.12)$$

$$\sigma_{GPS_{vel}} = 0.05 \frac{m}{s}$$

$$\sigma_{GPS_{\psi}}(i) = \frac{\sigma_{GPS_{vel}}}{V(i)} \text{ deg -or- } \sigma_{GPS_{\psi}}(i) = \sqrt{1e^6} \text{ deg, if velocity is 0 m/s}$$

The yaw gyroscope noise and bias values that were chosen are given in Equation (D.13), respectively.

$$\sigma_{Gz} = 0.1 \frac{\text{deg}}{s} \quad (D.13)$$

$$b_g = 0.1 \frac{\text{deg}}{s}$$

The longitudinal accelerometer noise and bias values that were chosen are given in Equation (D.14), respectively.

$$\sigma_{Ax} = 1 \frac{m}{s^2} \quad (D.14)$$

$$b_a = 0.1 \frac{m}{s^2}$$

D.6.3 Measurement Equations

The equations that were used to generate the “noisy” measurements are as follows. The GPS measurement equations for each discrete time, i , are given in Equation (D.15).

$$N_{GPS}(i) = N(i) + \sigma_{GPS_{pos}} * randn(1) \quad (D.15)$$

$$E_{GPS}(i) = E(i) + \sigma_{GPS_{pos}} * randn(1)$$

$$V_{GPS}(i) = V(i) + \sigma_{GPS_{vel}} * randn(1)$$

$$\psi_{GPS}(i) = \psi(i) + \sigma_{GPS_{\psi}} * randn(1)$$

The gyroscope measurement equation is shown in Equation (D.16).

$$g_z^m(i) = r(i) + b_g + \sigma_{G_z} * randn(1) \quad (D.16)$$

The accelerometer measurement equation is given in Equation (D.17). Note that the effect of incline or pitch is included in the accelerometer measurement.

$$A_x^m(i) = [A(i) + \sin(-\theta(i)) * (-g)] + b_a + \sigma_{A_x} * randn(1) \quad (D.17)$$

From these equations, a text file of the trial was created. Then, a six-state EKF like the one described in section B.3 was used to examine the adverse effect of pitch in the EKF

tracking solution. The ground speed or velocity, longitudinal accelerometer bias, yaw, yaw rate gyroscope bias, northern, and eastern positions were calculated from the integration of an accelerometer and gyroscope measurement with GPS velocity, course, northern position, and eastern position measurements. As is the case in section B.3, pitch is not accounted for in this motion model so that the error in the position and velocity solution can be examined.

D.6.4 Results and Discussion

When the trial profile does not include the road grade, the EKF position and velocity estimates track excellently. This is evident upon examination of the first portion of the trial, before the simulated canine begins movement on the inclined surface. However, as Figures D.3 and D.4 illustrate, as soon as the canine enters the inclined region, the EKF estimate is corrupted. In the position plot (Figure D.3), an offset from the GPS measurements ensues along with a “saw tooth”-like behavior. In between GPS measurements, only inertial sensor measurements are available. Since position estimates are based on the estimate of velocity (see Equation (B.5)), which are in turn based on the accelerometer measurement and the estimated bias on that accelerometer when the EKF is in between GPS measurements, and since the accelerometer is registering the effect of inclination and the EKF is not accounting for it, the position estimate begins to diverge immediately after a GPS measurement. In the EKF the GPS measurements are weighted with higher confidence due to their excellent accuracy. Therefore, as GPS measurements become available, the estimated position snaps back towards the GPS measurement. This results in a “saw tooth”-like appearance in the position.

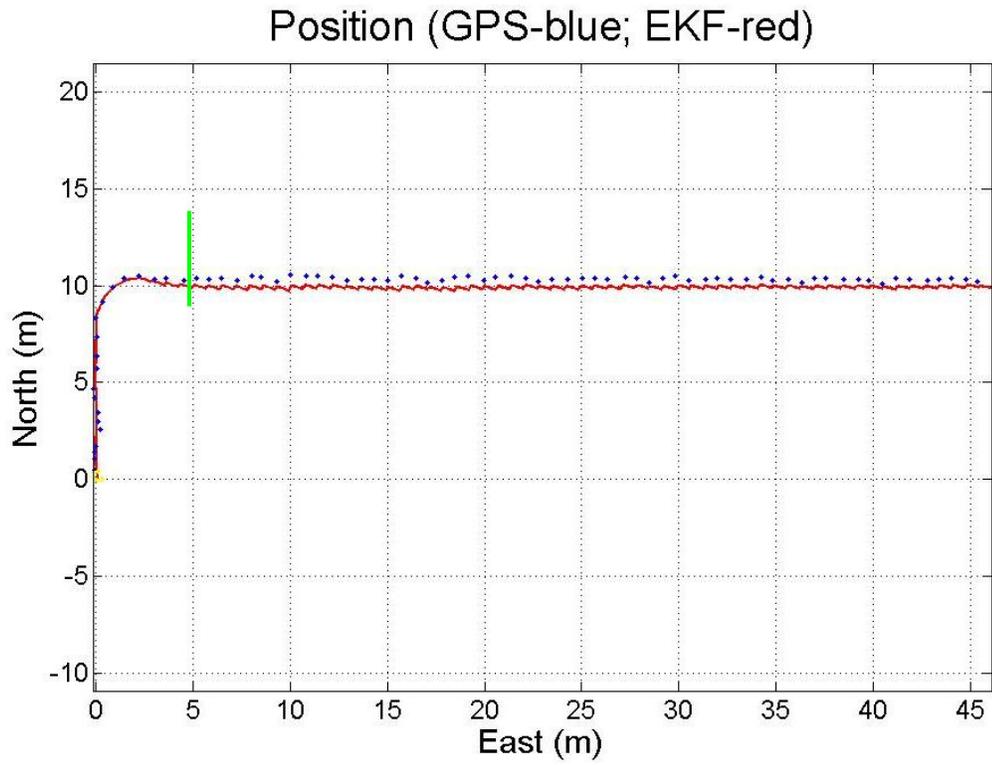


Figure D.3. Canine motion simulation EKF position estimate results when incline is present after the turn. Approximate initiation of incline is marked with a green line.

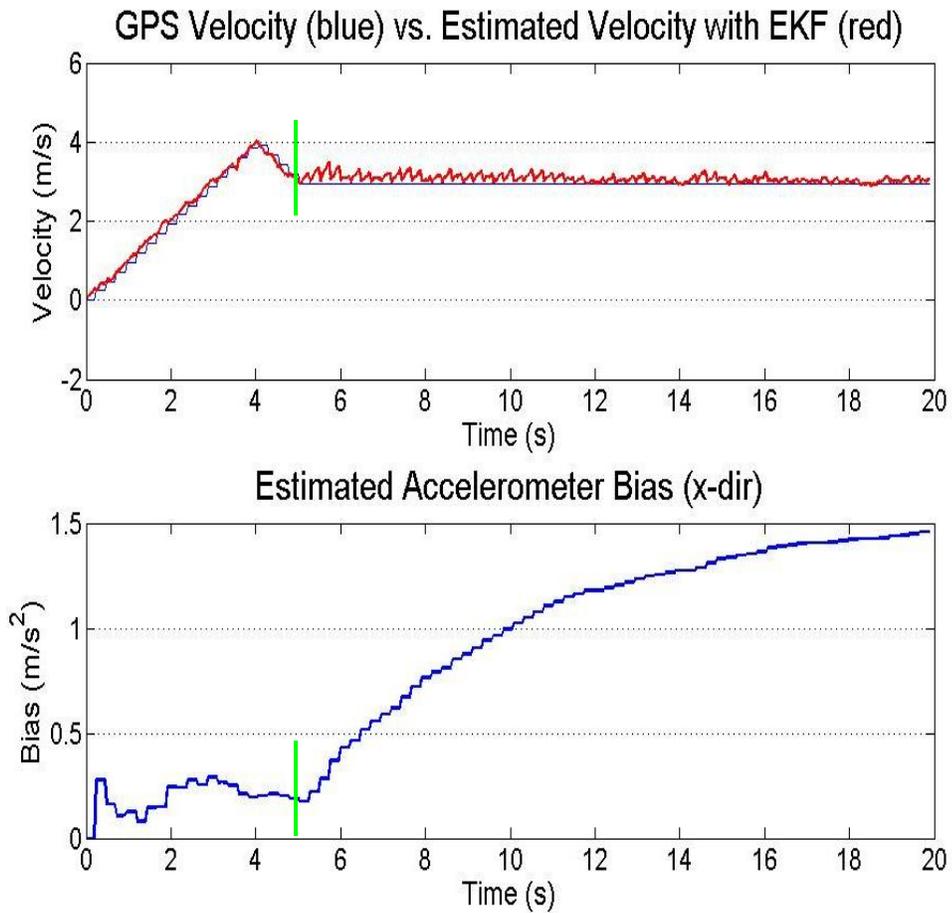


Figure D.4. Canine motion simulation EKF velocity and accelerometer bias estimate results when incline is present after approximately 5 seconds. Approximate initiation of incline is marked with a green line.

This “saw tooth”-like appearance is also present in the EKF velocity estimate (see Figure D.4) for the same reason. The GPS velocity measurement is weighted with higher confidence in the EKF. However, notable is the fact that the “saw tooth” projections get smaller and smaller and nearly disappear towards the end of the trial, at which time the EKF tracks the velocity well. This is due to the fact that the EKF lumped the effect of incline into the estimate of the accelerometer bias and consequently removed it.

However, the accelerometer bias which the EKF estimated is not the true bias of 0.1 m/s^2 (see Equation (D.14)). Thus, when the canine moves into an area with a different road grade or a zero road grade, the bias will be off. A GPS outage at such a time would result in a very quick divergence in position and velocity estimates.

D.6.5 Conclusions

It is clear that the presence of pitch or incline in a trial will significantly corrupt the standard EKF tracking estimates. It is also clear from Section D.5 that pitch is a major component of canine motion. Therefore, it can be inferred that pitch and inclination must be accounted for in the EKF canine motion model.

D.7 Experimental Results

Figures D.5-D.8 illustrate sample motion results when a canine is in odor detection mode. The blue dots represent straight GPS measurements and are used as “truth” measurements for the sake of comparison to the EKF results. The GPS position measurements are known to be within 3.5 meters of the actual canine position, based on the real-time horizontal accuracy measurements provided by the GPS receiver. The red lines represent the four state EKF results when pitch is or is not accounted for in the motion model. The green lines represent the EKF results when a simulated GPS outage is put in place $\frac{3}{4}$ of the way into the trial.

Figures D.5 and D.6 illustrate results when pitch is not accounted for in the motion model for the EKF. The velocity motion results in Figure D.5 exhibit a swift deviation of the EKF estimate from the GPS measurements. This is due to an inaccurate

estimate of the accelerometer bias by the EKF because of the unmodeled motion characteristics of the canine. After a GPS outage of about 20 seconds, the velocity estimate is off by about 4 m/s.

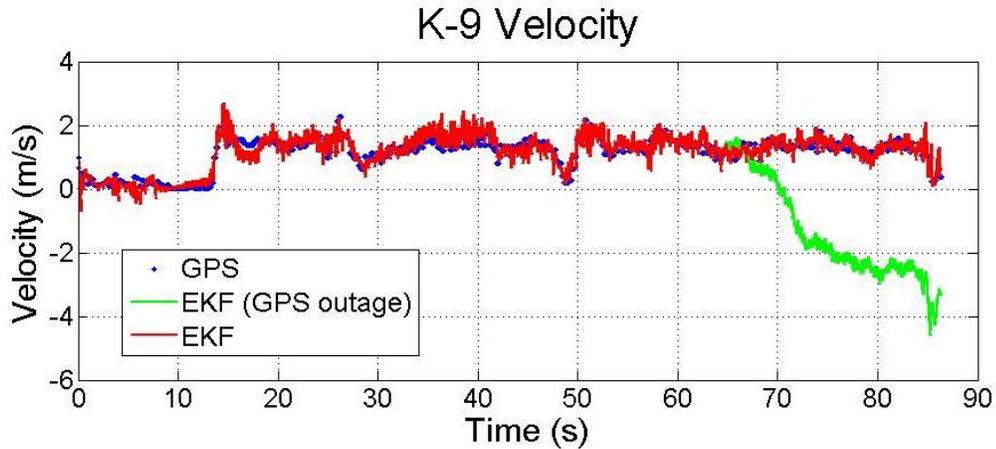


Figure D.5. EKF canine velocity estimate when pitch is not accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

The position results (Figure D.6) also exhibit significant error during the GPS outage. After the 20 second outage, the position results are off by about 50 meters. This is due, again, to the inaccurate estimate of the accelerometer bias, since the position results are based on the velocity estimate, according to Equation (B.5). Notice that when the estimated velocity incorrectly drops below 0 m/s (Figure D.5), the canine is observed to do an “about face” in the position results, which in turn results in significant deviation in the position estimate in a short amount of time. This shows the importance of accurately estimating the accelerometer bias as soon as possible before a GPS outage occurs.

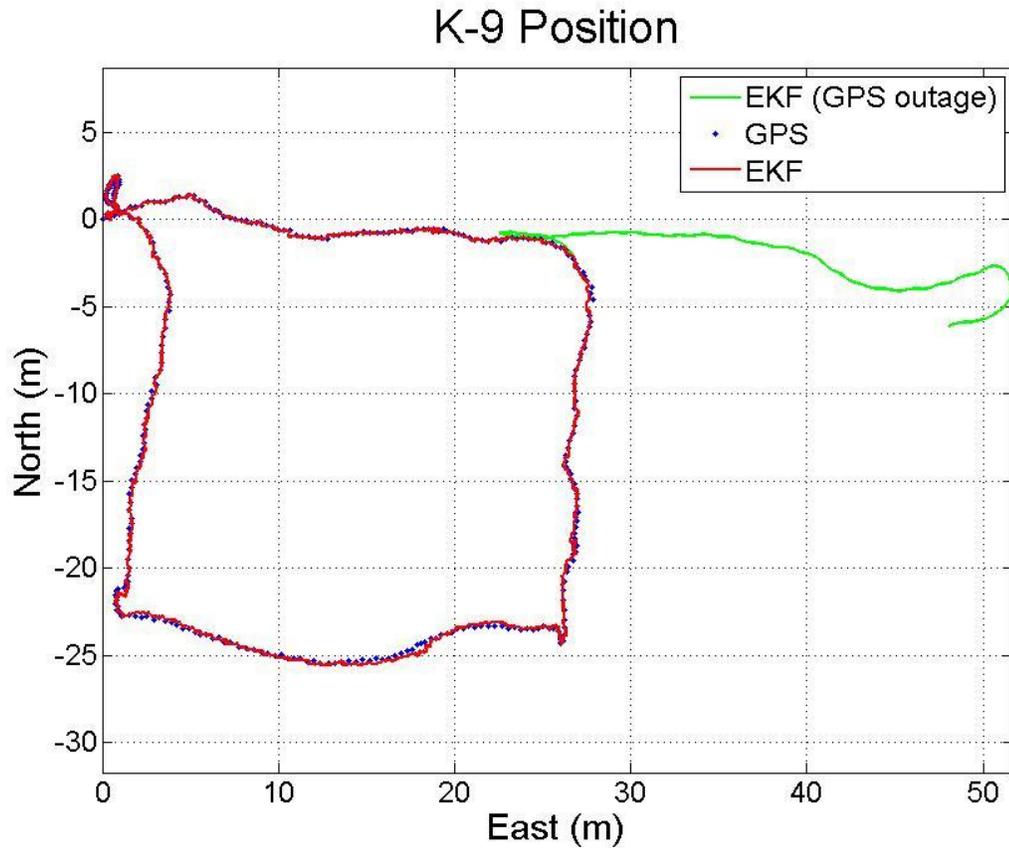


Figure D.6. EKF canine position estimate when pitch is not accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

Figures D.7 and D.8 illustrate canine motion results when pitch is accounted for in the motion model for the EKF. The velocity results (Figure D.7) show that the EKF velocity solution remains close to the GPS measurements throughout the simulated outage. After the approximate 20 second GPS outage, the velocity deviation is about 0.5 m/s.

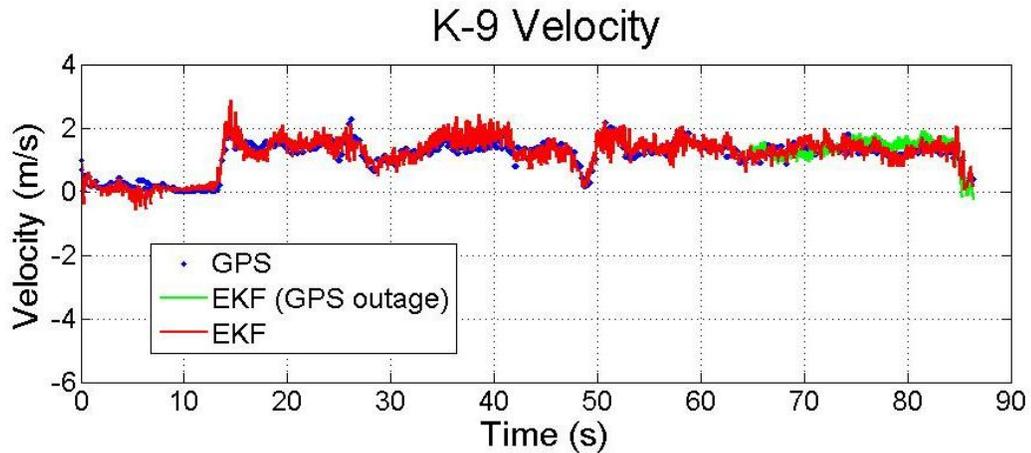


Figure D.7. EKF canine velocity estimate when pitch is accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

The position results (Figure D.8) illustrate that the EKF solution stays within 1 meter of the GPS measurements. By the end of the simulated GPS outage, the EKF position estimate is about 1 meter away from the GPS solution.

Figures D.9-D.10 illustrate sample motion results of a canine when it is sprinting. Notable is the fact that canine running motion clearly exhibits “noisier” results than the results from when the canine is in odor detection mode. The EKF solutions with and without pitch being accounted for exhibit “saw tooth”-like behavior at times. This indicates that canine running motion contains other unmodeled noise characteristics which should be addressed in the future. The solution that was chosen was discussed in Chapter 3.

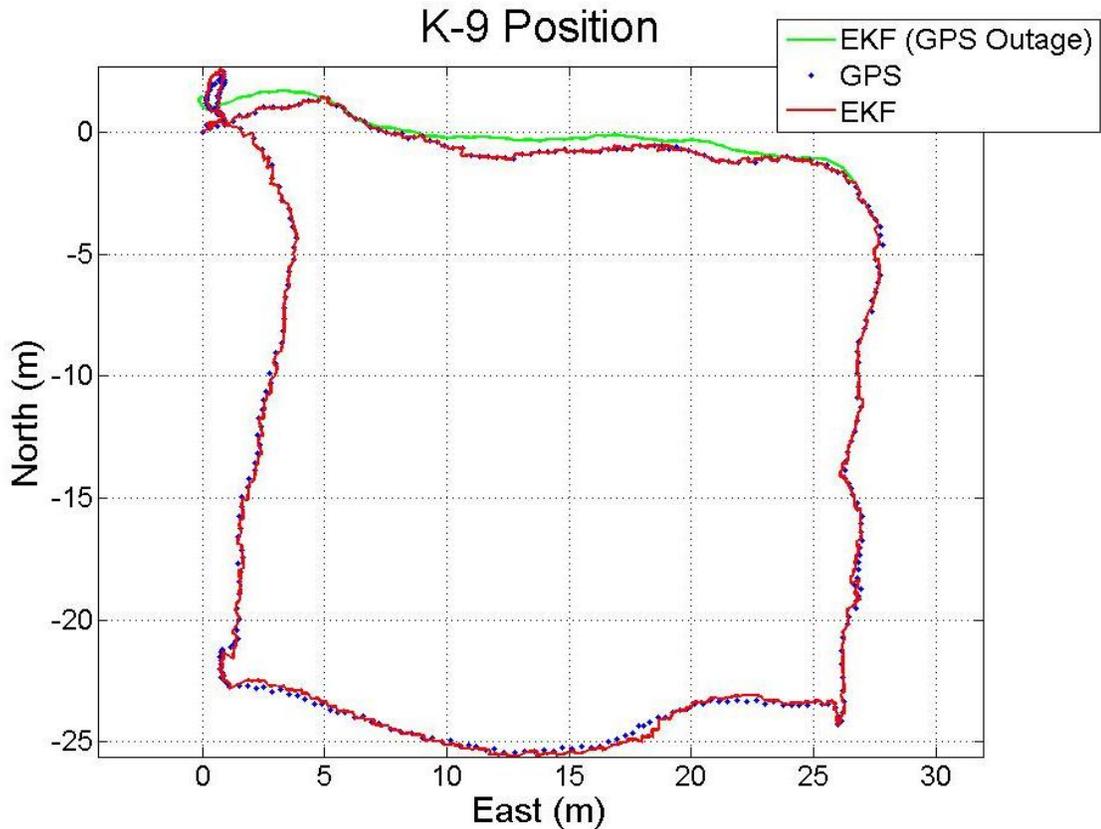


Figure D.8. EKF canine position estimate when pitch is accounted for in the motion model. The canine is in odor detection mode. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

The results illustrated in Figures D.9 and D.10 show results when pitch is not accounted for in the motion model of the EKF. As in the previous examples, the EKF position and velocity solutions deviate swiftly and significantly from the GPS solutions. After a simulated GPS outage of only about 7 or 8 seconds, the velocity estimate is off by some 18 m/s, and the position estimate is off by approximately 55 meters. As in the above example, since the accelerometer bias has not been estimated effectively by the

outage, the velocity solution is becoming more and more erroneous by the moment. Thus, as is implied by Equation (B.5), the position solution error increases even faster.

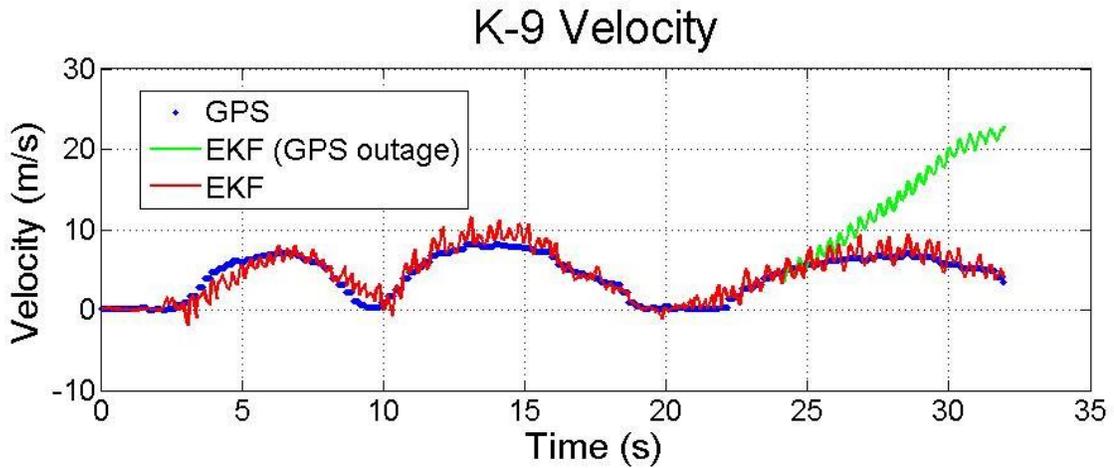


Figure D.9. EKF canine velocity estimate when pitch is not accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

Figures D.11 and D.12 illustrate the canine motion results when pitch is accounted for in the motion model. Although the sprinting results are noisy and clearly contain unmodeled motion characteristics, it is evident that accounting for pitch in the motion model significantly improves the solution during the GPS outage. The velocity solution remains close to the GPS velocity measurements throughout the outage, and the position solution is off by only 5 meters by the end of the outage.

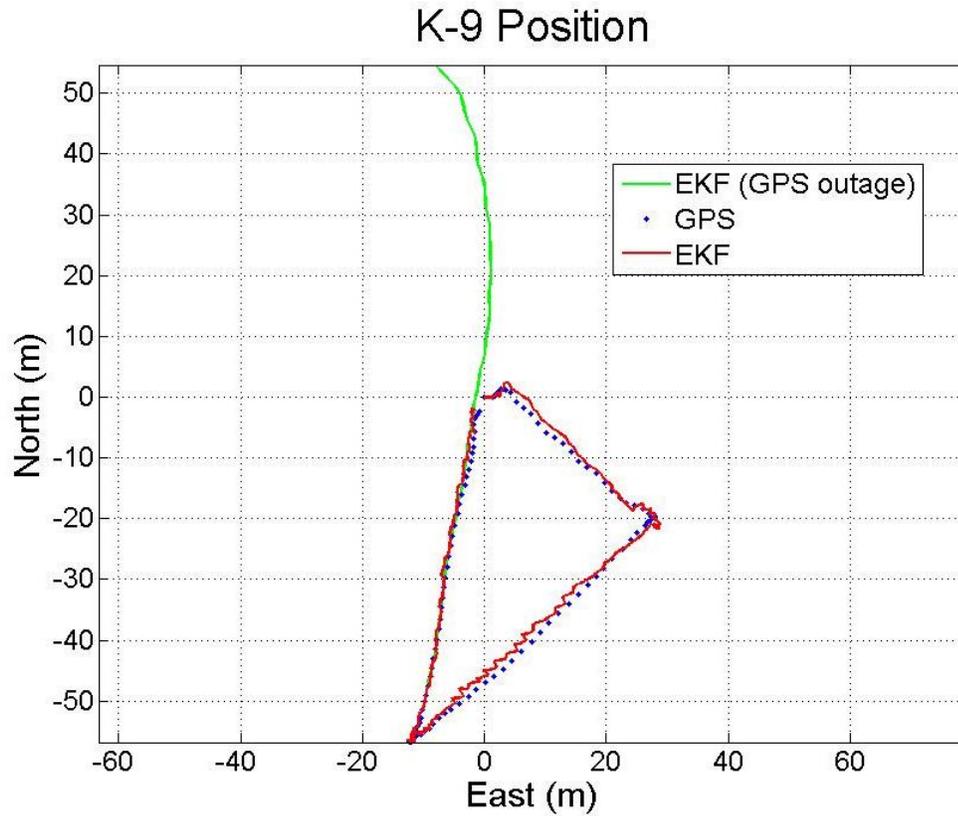


Figure D.10. EKF canine position estimate when pitch is not accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

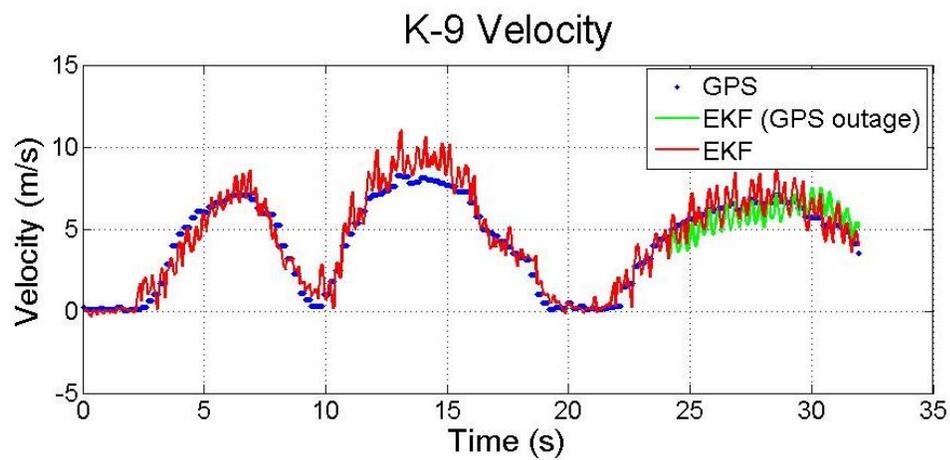


Figure D.11. EKF canine velocity estimate when pitch is accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

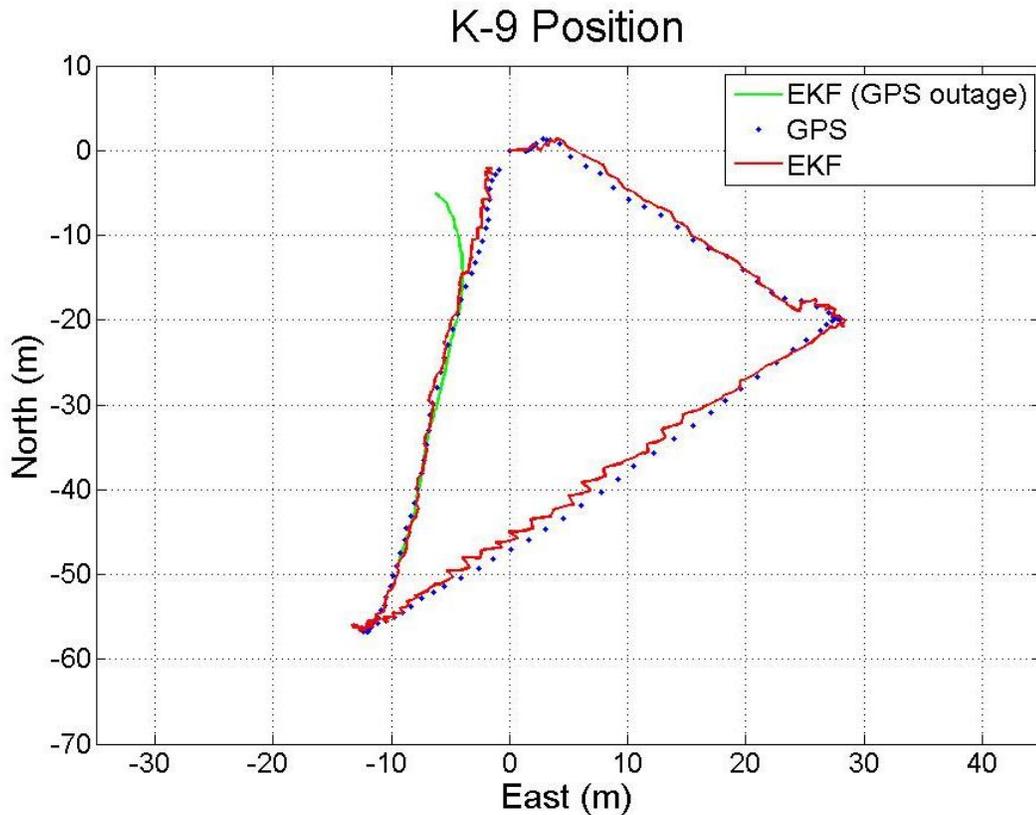


Figure D.12. EKF canine position estimate when pitch is accounted for in the motion model. The canine is sprinting. A simulated GPS outage occurs after $\frac{3}{4}$ of the trial.

As a final visual comparison, Figure D.13 illustrates another example of the position solution for a canine in odor detection mode. The GPS solution (dark blue dots) is nearly covered up by the two EKF solutions. The cyan colored line is the EKF solution without pitch in the motion model, and the red colored line represents the EKF solution when pitch is considered in the motion model. Notable in this plot is the non-GPS solution (i.e., the green line). When the inertial sensors and magnetometers are used alone in determining the canine position (i.e., without corrections provided from GPS

measurements), the final position error is about 300 meters. This illustrates the value of integrating the inertial sensors, magnetometers, and GPS.

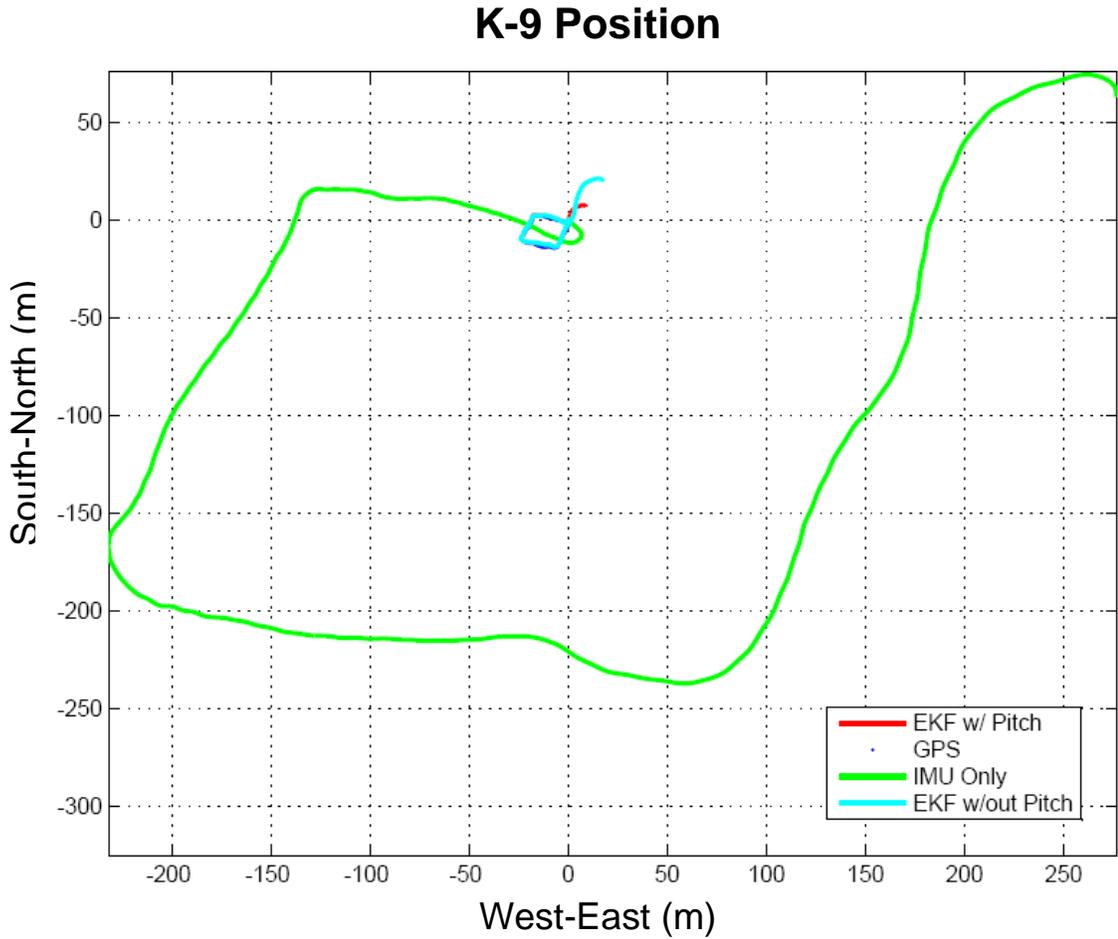


Figure D.13. Canine position results for various sensor configurations when the canine is in odor detection mode. A simulated GPS outage occurs the final 10 seconds of the trial.

Table D.1 illustrates average motion tracking results for a canine in odor detection mode after a 10 second GPS outage. The trials are approximately 40 seconds each. The

given sensor configurations are compared to GPS measurements, which are known to be accurate to within 3.5 meters, according to the horizontal accuracy measurements provided by the GPS receiver. As reported in [45], the results were found to be statistically significant using the two-tailed, paired samples T-test. Table D.2 illustrates the results of the T-testing.

As is illustrated in Figure D.13, Table D.1 compares results from different sensor configurations. Obviously, if GPS alone were used in tracking the canine, the moment GPS is lost, the position and velocity of the canine would be unknown until GPS was restored. When GPS is not used to correct the error on the longitudinal accelerometer and the accelerometer and heading measurements are alone used to determine the position and velocity of the canine, by the end of a trial the position error is 340.4 meters on average. The velocity error at the end of such a trial is 39.6 m/s on average. When a standard-type EKF is used, and the measurement from the accelerometer, the heading measurement, and GPS velocity and position measurements are used together, the position error by the end of the trial is reduced to 21.5 meters on average, and the velocity error is cut to 5.1 m/s on average. However, when the GPS measurements are coupled with the accelerometer, heading, and a pitch measurement, the errors are nearly cut in half. On average, the position error is reduced to 11.6 meters, and the velocity error is reduced to 2.8 m/s by the end of the GPS outage. Notable also is the fact that the standard deviation is lowered significantly as well, thus providing more confidence in the EKF solutions.

Table D.1. Average canine motion tracking error results for various sensor configurations after a 10 second GPS outage. The canine is in odor detection mode.

Configuration	Position Error (m)		Velocity Error (m/s)	
	Mean	Std. Dev.	Mean	Std. Dev.
Acc., Head.	340.4	355.1	39.6	25.1
GPS, Acc., Head.	21.5	17.2	5.1	4.6
GPS, Acc., Head, Pitch	11.6	7.2	2.8	1.8

Table D.2. Two-tailed, paired samples T-tests for canine position and velocity.

Configuration pairing	Position Error (m)		Velocity Error (m/s)	
	<i>t</i>	Sig.	<i>t</i>	Sig.
Acc., Head.—GPS, Acc., Head.	4.889	$p < 0.001$	7.685	$p < 0.001$
Acc., Head.—GPS, Acc., Head., Pitch	5.129	$p < 0.001$	8.212	$p < 0.001$
GPS, Acc., Head.—GPS, Acc., Head., Pitch	2.948	$p = 0.006$	2.832	$p = 0.008$

31 trials (df = 30)

D.8 Conclusions

Both intuition and the evidence indicate that pitch plays a significant role in a canine’s motion characteristics. Failing to account for it in the EKF motion model accounts for a significant source of corruption in the estimate of the accelerometer bias, which is needed for tracking in cases where GPS outages occur. Accounting for pitch effects in the motion model, however, reduces the position and velocity errors to a palatable level and reduces the standard deviation significantly, leading to more confidence in the EKF solution.

However, as articulated above, the “noise” level and “saw tooth”-like behavior, especially in running scenarios, indicates that there are other unmodeled canine behaviors still present in the EKF solution. Accounting for these could lead to a better estimate of the accelerometer bias and a better tracking solution during GPS outages. Chapter 3 presents an effective approach to compensate for such unmodeled noise characteristics.