

DGPS-BASED LOCALIZATION
FOR PATH FOLLOWING OF OUTDOOR WHEELED MOBILE
ROBOTS

by

Leslie Ssebazza

Submitted in partial fulfillment of the
requirements for the degree of
Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
July 2011

© Copyright by Leslie Ssebazza, 2011

DALHOUSIE UNIVERSITY

DEPARTMENT OF MECHANICAL ENGINEERING

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “DGPS-BASED LOCALIZATION FOR PATH FOLLOWING OF OUTDOOR WHEELED MOBILE ROBOTS” by Leslie Ssebazza in partial fulfillment of the requirements for the degree of Master of Applied Science.

Dated: July 25, 2011

Supervisor:

Readers:

DALHOUSIE UNIVERSITY

DATE: July 25, 2011

AUTHOR: Leslie Ssebazza

TITLE: DGPS-BASED LOCALIZATION FOR PATH FOLLOWING OF
OUTDOOR WHEELED MOBILE ROBOTS

DEPARTMENT OR SCHOOL: Department of Mechanical Engineering

DEGREE: M.A.Sc. CONVOCATION: October YEAR: 2011

Permission is herewith granted to Dalhousie University to circulate and to have copied for non-commercial purposes, at its discretion, the above title upon the request of individuals or institutions. I understand that my thesis will be electronically available to the public.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

The author attests that permission has been obtained for the use of any copyrighted material appearing in the thesis (other than brief excerpts requiring only proper acknowledgement in scholarly writing), and that all such use is clearly acknowledged.

Signature of Author

*To my parents, family, and friends
without your encouragement and support
this would not have been possible.*

Table of Contents

List of Tables	viii
List of Figures	ix
List of Symbols and Abbreviations Used	xiii
Acknowledgements	xiv
Abstract	xv
Chapter 1 Introduction	1
1.1 Background	1
1.1.1 Wheeled Mobile Robots and Sensors	2
1.1.2 Path Following	3
1.1.3 Localization	4
1.2 Relevant Applications	5
1.3 Motion Control Techniques	5
1.4 Path/Motion Planning	6
1.5 Thesis Contribution	7
1.6 Thesis Organization	8
Chapter 2 Literature Review of Path Following for Wheeled Mobile Robots	9
2.1 dGPS-based Motion Control for WMR in Path Following	9
2.2 Potential Field Motion Planning used for Path Following	14
2.3 Discussions	15
2.4 Chapter Summary	17
Chapter 3 Proposed dGPS-based Path Following Approach	18
3.1 Problem Formulation	18
3.2 dGPS-based Path Following	19
3.2.1 System Architecture	20

3.2.2	Modified Potential Field Path Planning (PFPP) Algorithm . . .	21
3.2.3	Numerical Methods (NM) Based Algorithm	23
3.2.4	Extended Kalman Filter (EKF)	24
3.3	Extension to Multiple Robots	29
3.4	Chapter Summary	30
Chapter 4	Simulation Results	32
4.1	Motion planning based on geometric progression (or modified) PFPP Method	33
4.1.1	Simulation Results	33
4.2	dGPS-based Motion Control NM Algorithm with EKF for Path Following	40
4.2.1	Simulation Results	41
4.3	Summary	48
Chapter 5	Experimental Studies	52
5.1	Hardware and Software Setup	52
5.1.1	Procedure	54
5.2	Preliminary Experimental Results	56
5.2.1	Case 1: Rectilinear Motion	56
5.2.2	Case 2: Curvilinear Motion	58
5.2.3	Summary	59
5.3	Experimental Results	63
5.3.1	Tuning R_{Δ} parameter	63
5.3.2	Case 1: Rectilinear Motion	65
5.3.3	Case 2: Curvilinear Motion	66
5.3.4	Summary	67
5.4	Data Analysis and Comparisons	68
5.4.1	Case 1: Rectilinear Motion dGPS without EKF	68
5.4.2	Case 2: Curvilinear Motion dGPS without EKF	70
5.4.3	Comparisons of dGPS localization with active EKF control	72
5.5	Summary	73
Chapter 6	Conclusion	78
6.1	Future Work	78
6.1.1	Model and Simulation	78
6.1.2	Experiment	79

6.1.3	Expansion	79
Appendix A	Engineering Drawings	81
Appendix B	Matlab Code	82
Bibliography	88

List of Tables

Table 2.1	Types of NMEA 0183 Data Strings [66]	10
Table 2.2	Examples of Bayesian Techniques used in WMR localization Applications	12
Table 2.3	EKF Architecture	13
Table 2.4	PFPP Pseudo Code Example (Similar to [48])	15
Table 3.1	Geometric Progression PFPP Pseudo Code	23
Table 5.1	Preliminary Position Error Comparisons	62
Table 5.2	Position Error Comparisons	75

List of Figures

Figure 1.1	Collaboration of Unmanned Vehicles (Space and Naval Systems Center, San Diego)[53]	1
Figure 1.2	General WMR Control	2
Figure 1.3	Path Following	3
Figure 1.4	Type (2,0) WMR	4
Figure 1.5	Google Test Car [28]	6
Figure 1.6	General WMR Control with Thesis Focus Highlighted	7
Figure 2.1	Wide Area Augmented System (WAAS)	11
Figure 2.2	Artificial Potential Field Example	14
Figure 2.3	Artificial Potential Field Path Planning (PFPP)	16
Figure 3.1	WMR with WAAS dGPS receiver	19
Figure 3.2	Path Following Flow Control Architecture	20
Figure 3.3	Weighted local minimum solution	21
Figure 3.4	PFPP with boundary of influence to the object of interest	22
Figure 3.5	Variance Data for $\sigma_{x_{gps}}$ variable in Measurement Covariance Matrix R_v	27
Figure 3.6	Variance Data for $\sigma_{y_{gps}}$ variable in Measurement Covariance Matrix R_v	28
Figure 3.7	Variance Data for $\sigma_{\theta_{gps}}$ variable in Measurement Covariance Matrix R_v	28
Figure 3.8	Applying Geometric Progression PFPP to Multiple WMR (number of robots = 2)	31
Figure 4.1	MobileSim Simulation (screen capture) for a Linear Path using Odometric Sensor	32
Figure 4.2	Modified PFPP Linear Path 1	34
Figure 4.3	Artificial Resulting Force (F_{tot}) for Modified PFPP Linear Path 1	35

Figure 4.4	Modified PFPP Curved Path 2	36
Figure 4.5	Artificial Resulting Force (F_{tot}) for Modified PFPP Curved Path 2	36
Figure 4.6	Modified PFPP Curved Path 3	37
Figure 4.7	Artificial Resulting Force (F_{tot}) for Modified PFPP Curved Path 3	38
Figure 4.8	Modified PFPP Linear Path 4	38
Figure 4.9	Modified PFPP Linear Path 4 (Zoomed Profile)	39
Figure 4.10	Modified PFPP Curved Path 5	40
Figure 4.11	Modified PFPP Curved Path 5 (Zoomed Profile)	41
Figure 4.12	Comparing controller gain values of k_v, k_w using Mobilesim and Equation (3.5)	42
Figure 4.13	MobileSim using EKF with $\hat{\theta}(0) \neq \theta_{ref}(0)$	43
Figure 4.14	NM Path Following Control using EKF (Path 1)	44
Figure 4.15	Control Velocities v_c, w_c (Path 1)	44
Figure 4.16	NM Path Following Control using EKF (Path 2)	45
Figure 4.17	Control Velocities v_c, w_c (Path 2)	46
Figure 4.18	NM Path Following Control using EKF (Path 3)	47
Figure 4.19	Control Velocities v_c, w_c (Path 3)	47
Figure 4.20	NM with Online EKF Control for Linear Path 4 Path Following	48
Figure 4.21	Control Velocities v_c, w_c (Path 4)	49
Figure 4.22	NM with Online EKF Control for Curved Path 5 Path Following	49
Figure 4.23	Control Velocities v_c, w_c (Path 5)	50
Figure 5.1	Hardware, Software and Testing Setup	52
Figure 5.2	Testing Environment with Stationary Point Test Data Sets . .	54
Figure 5.3	Wiimote Teleoperation	55
Figure 5.4	Testing Run Sequence Path 5	55
Figure 5.5	Comparing Controller gains experimentally (without EKF) . .	57

Figure 5.6	NM with Online EKF Control for Linear Path 4 Path Following (Zoomed Profile)	58
Figure 5.7	Control Velocities v_c, w_c (Path 4)	59
Figure 5.8	NM with Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)	60
Figure 5.9	Control Velocities v_c, w_c (Path 5)	60
Figure 5.10	NM without Online EKF Control for Linear Path 4 Path Following (Zoomed Profile)	61
Figure 5.11	NM without Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)	62
Figure 5.12	Tuning R_Δ NM with Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)	64
Figure 5.13	Tuning R_Δ NM without Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)	64
Figure 5.14	Linear Simulation Path 6 Path Following	65
Figure 5.15	NM with Online EKF Control for Linear Path 6	66
Figure 5.16	Control Velocities v_c, w_c (Path 6)	67
Figure 5.17	Curvilinear Simulation Path 7 Path Following	68
Figure 5.18	NM with Online EKF Control for Curved Path 7	69
Figure 5.19	Control Velocities v_c, w_c (Path 7)	69
Figure 5.20	NM without Online EKF control for Linear Path 6	70
Figure 5.21	Control Velocities v_c, w_c (Path 6)	71
Figure 5.22	NM without Online EKF Control for Curvilinear Path 7	71
Figure 5.23	Control Velocities v_c, w_c (Path 7)	72
Figure 5.24	Velocity v_c (Path 6)	73
Figure 5.25	Velocity w_c (Path 6)	74
Figure 5.26	Velocity v_c (Path 7)	74
Figure 5.27	Velocity w_c (Path 7)	75
Figure 5.28	Experiment Position Error (Path 6)	76
Figure 5.29	Experiment Position Error (Path 7)	77

Figure 6.1	ACM Lab Research Robotic Vehicles	80
Figure A.1	Mounting Platform Assembly	81

List of Symbols and Abbreviations Used

WMR - Wheeled Mobile Robot

PFPP - Potential Field Path Planning

GPPFPP - geometric progression Potential Field Path Planning

GPS - Global Positioning System

dGPS - differential Global Positioning System

GEO - Geostationary

GNSS - Global Navigation Satellite System

RTK - Real Time Kinematic

WAAS - Wide Area Augmented System

SBAS - Space Based Augmentation Systems

LAGPS - Local-Area Differential GPS

IMU - Inertial Measurement Unit

KF - Kalman Filter

PF - Particle Filter

UKF - Unscented Kalman Filter

EKF - Extended Kalman Filter

NM - Numerical Method

NMEA - National Marine Electronics Association

Acknowledgements

First and foremost I would like to thank my parents who have always been there to support and encourage me. By setting the example they have led me to strive for more, expand my horizon and aim for the heights. I'd like to thank my siblings and friends who have been on this journey with me, always lending an ear to my thoughts. To my family in Uganda and abroad I thank you for never doubting, always confident in me even when my own confidence in myself swayed. You gave me strength in times when I needed it most. To my Imhotep community, I thank you for allowing me to be apart of something and always encouraging my endeavours.

I would like to thank my supervisor Dr. Ya-Jun Pan for recognizing the potential in me. Without her continued help, support and direction, this work would not be possible. I would also like to thank the Dalhousie staff, especially Angus MacPherson, Jonathan MacDonald, Rob Warner, Andrew Warkentin, Rob Bauer and Rishad Irani, all of who opened doors for me that gave me the chance to develop and grow.

Finally, thanks to my graduate student colleagues, who have added value to my work in one way or another. Especially ACM research group members Dylan, Julian, Long, Balu and Trent who's help was really appreciated.

Funding for this work was provided by AUTO 21, NSERC, and CFI.

Abstract

Differential Global Positioning System (dGPS) sensors provide a way for an outdoor wheeled mobile robot to achieve better localization that results in improved navigation and control of outdoor Wheeled Mobile Robots (WMR). This thesis proposes an approach for path following of outdoor WMR. The primary focus of the approach is to use dGPS technology as the only sensing device for localization. Filter estimation techniques are also considered to account for measurement inaccuracies. The Extended Kalman Filter (EKF) is chosen and implemented as part of this approach for improved results. A secondary focus of the approach, is to incorporate a modified Potential Field Path Planning (PFPP) as an integral part of the proposed technique. The dGPS-based proposed path following approach is first simulated in the simulation environment. After which, it is applied and tested on a WMR experimental platform for a set of experimental cases.

Chapter 1

Introduction

Mobile robotic vehicle research has become an emerging research area motivated by the increase in efficient automated systems in the human work environment. The interactions between a robot, its task and the environment are primarily what mobile robots research is focused on. Examples of this can be found in the manufacturing, agriculture and tourism sectors. Research has also focused on how intelligent autonomous vehicles can be used for search, rescue and terrestrial applications [53]. Search robot teams can be used in hazardous environments to collaborate and perform dangerous tasks, such as searching for mines in a mine field (Fig. 1.1). A team of multiple robots can also be employed to explore and map an unknown area.

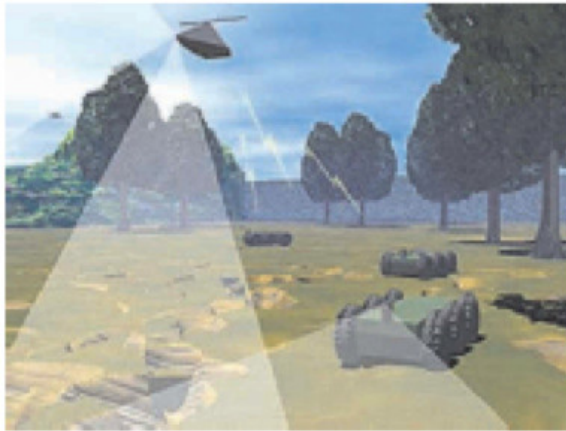


Figure 1.1: Collaboration of Unmanned Vehicles (Space and Naval Systems Center, San Diego)[53]

1.1 Background

Coordination and control of wheeled mobile robots (WMR) autonomously can be classified into two parts, Navigation and Learning [2]. Navigation of a WMR includes self localization, mapping, path planning and recovery. Learning aspects encompass

reactive behaviours such as reasoning and intelligent perception-based response. This thesis contribution is focused on self localization, recovery and the path planning aspect of navigating and controlling a WMR autonomously. A general WMR control architecture is shown in Fig. 1.2.

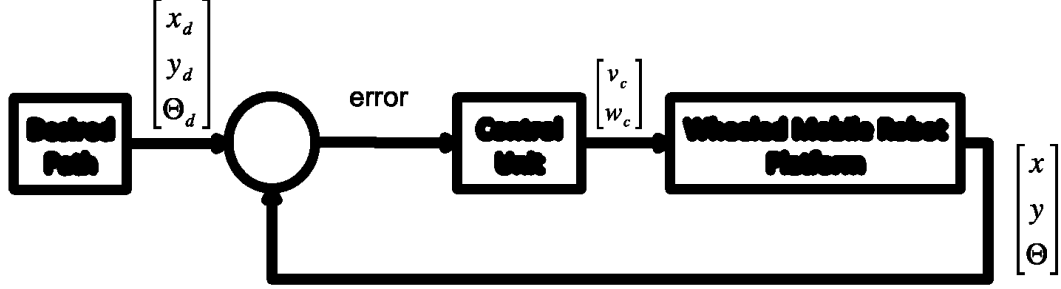


Figure 1.2: General WMR Control

1.1.1 Wheeled Mobile Robots and Sensors

In order for a WMR to autonomously navigate from one position to another, it must first have a desired path or end goal. Secondly, it must be able to obtain information of its location as it tries to follow the path within the confines of a given environment. A desired path is generated using a path planning module (examples of which can be found in [4]). Given the reference trajectory $q_{des}(r(t))$, and using the general WMR control architecture shown in Fig. 1.2, a control law can then be developed and controller designed. The controller is used to ensure that the error $e_q(t) = (q_{cur}(t) - q_{des}(r(t)))$ between the actual WMR's current pose $q_{cur}(t)$ and the desired $q_{des}(t)$ converges to a fixed value (or zero). In this case $q = [x, y, \theta]^T$, and $q_{cur}(t)$ and $q_{des}(t)$ are vectors associated with the robot pose and reference path respectively. Fig. 1.3 shows the relationship between current and desired trajectory during path following for a single robot. Sensors mounted on the robot are used to estimate and obtain information about the position of the robot. These sensors are classified as either proprioceptive (i.e. optical encoder, gyroscope) or exteroceptive (i.e. sonar, GPS) [34]. The former provides information of the robot's condition while the latter provides information of the environment condition.

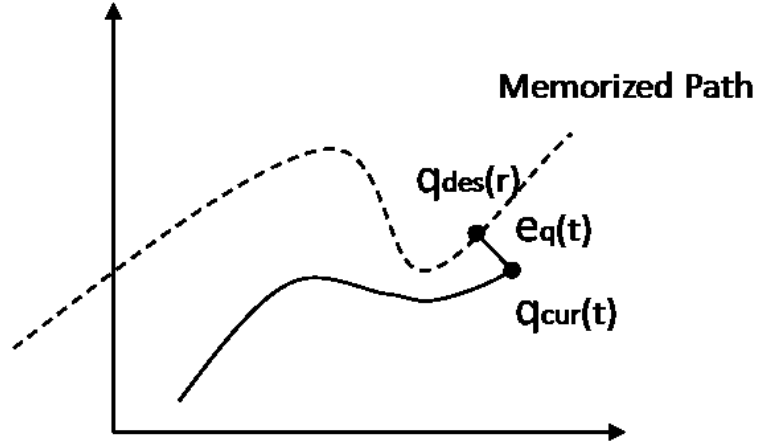


Figure 1.3: Path Following

1.1.2 Path Following

For path following control, the dynamic and kinematic models of a wheeled mobile robot must first be explored and known. Guy Champion et. al. addresses this in [5] by introducing concepts such as the degree of mobility and the degree of steerability and formulating a set of equations that take into account structural restrictions such as nonholonomic (differential) constraint. Using this formulation Champion is able to develop a generalized set of state space models that are useful in understanding the behaviors of wheeled mobile robots (WMR). The kinematics and dynamics of the WMR types considered are Type (2,0) “differential drive” mobile robots shown in Fig. 1.4. A kinematic WMR model uses a higher-level control scheme which doesn’t concern itself with the forces that generate the control velocities to achieve the desired position. The generalized ideal posture kinematic model for a WMR is written in terms of $\dot{q} = S(q)u$, where q is the global pose vector, u is the WMR velocity and steering components, and $S(q)$ is a transformation matrix that relates the two[5]. For Type (2,0) WMR this equation becomes:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \sin(\theta) & 0 \\ \cos(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix}. \quad (1.1)$$

where input u corresponds to the manipulated linear and angular control velocities $(v, w)^T$ respectively. A dynamic WMR model uses a higher-level and low-level control

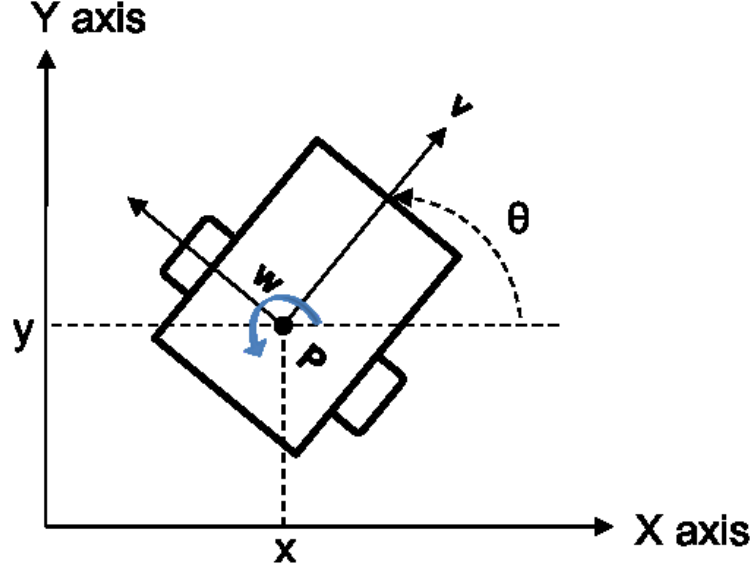


Figure 1.4: Type (2,0) WMR

scheme that concerns itself with the forces that generate the control velocities to achieve the desired position. The generalized posture dynamic model from [5] and [6] can be written as Equation (1.2) and (1.3).

$$\dot{q} = S(q)u \quad (1.2)$$

$$\bar{M}(q)\dot{u} + \bar{V}_m(q, q)u + \bar{F} + \bar{\tau}_d = \bar{B}(q)\tau \quad (1.3)$$

where q is the global pose vector, u is the WMR velocity and steering components, $S(q)$ is a transformation matrix, $\bar{M}(q)$ is a symmetric positive definite inertial matrix, $\bar{V}_m(q, q)$ is the centripetal and coriolis matrix, \bar{F} is the surface friction, $\bar{\tau}_d$ is the bounded unknown disturbances and $\bar{B}(q)$ is the input transformation matrix.

1.1.3 Localization

For WMR localization, sensor data from both exteroceptive and proprioceptive sensors [34] is used. There are some inherent issues associated with localization estimates based on sensors. The accuracy of a localization sensor in estimating the position can vary depending on sensor limitations and environmental conditions. Issues such as sensor noise, poor calibration, loss of signal, nonlinearity, corrupt data and drift all can lead to poor localization [36]. Sensor fusion is a concept that is used to decrease

the effects of inaccurate sensor data by allowing use of more than one type of sensor to estimate a WMR's position. Relying on odometric sensors alone, such as encoders used in dead reckoning estimation of position, can lead to errors in localization especially in cases where there is uneven terrain or loss of traction or wheel slippage during the WMR's motion. Similarly, accelerometer and gyroscope sensors used in Inertial Measurement units (IMU) can drift and lead to errors that grow over long periods of time, if not corrected. Global Positioning Systems (GPS) exist as a localization sensor alternative that can offer a better estimate on the position over long periods of time. For more accurate positioning specialized GPS units such as differential GPS (dGPS) are used although they are expensive and require unobstructed line of sight to the satellites.

1.2 Relevant Applications

Path following control for outdoor WMR's is relevant for autonomous vehicles in military, space and traffic control applications. For military use this includes searching an unexplored hazardous unknown environment for explosives. This can also be extended to space and interplanetary exploration such as exploring the Mars planet and collecting samples for analysis. For the case of traffic control applications, path planning and following cases have been noted in the tests conducted by Google using the Google test vehicle shown in Fig. 1.5. Other funded research, such as 'road train' research done by the European Commission [54] or DARPA challenge [29] done by various universities, has also focused on autonomous vehicles use in traffic control. The move towards autonomous vehicles on the road in the future could lead to more efficient driving which could result in reduced traffic congestion, fuel consumption and the number of road accidents.

1.3 Motion Control Techniques

There are several controller designs that have been proposed for WMR's path following. Path following control in [8] focuses on both trajectory tracking [16] and point stabilization [23]. These controllers are based on the nonlinear nature of the kinematic model in Equation (1.1). Other controller used are dynamic feedback linearization in

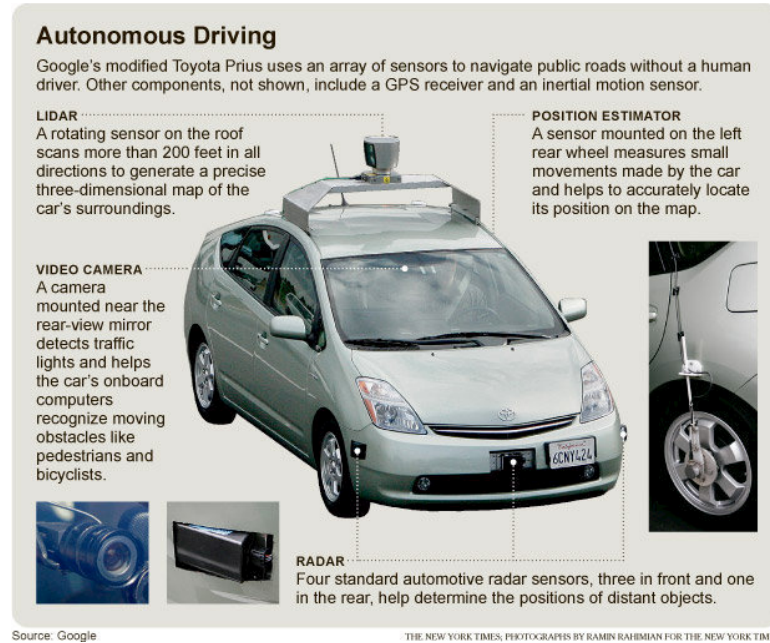


Figure 1.5: Google Test Car [28]

[7] [8], sliding mode control by Aguilar et. al. proposed in [11] and backstepping control proposed by Fierro et. al. in [6] [24], and controllers that are based on Lyapunov stability criterion [18]. Although some of these controllers are robust in nature they are based on simulations where the position of the WMR can be obtained at all times and has minimal to no inaccuracies. Due to inaccuracies associated with localization sensors as noted in Section 1.1.3, Bayesian estimation and prediction techniques, such as Extended Kalman Filter (EKF) and Unscented Kalman Filters (UKF), can be used to improve position estimate of an outdoor WMR.

1.4 Path/Motion Planning

Path planning for WMR is either global or local. Global path planning is a map-based approach that uses world environment data, such as GPS data on a world map, to determine a path [4]. Local path planning is a reactive approach where a path is determined based on the current situation. An example of local path planning is when a WMR, using sonar data, determines a collision free path when it encounters an obstacle in its vicinity. Some considerations when developing and using a path planning approach are: (1) if the environment is known or unknown and (2) if the

path planning calculations will be done online (as the robot moves) or offline (before the robot starts moving). When using local path planning, unknown environments can increase computation time during motion. In local path following the WMR requires an array of proximity sensors to be able to sense its immediate environment and plan a path accordingly. Global path planning is often used for offline known environments.

Path planning techniques [37] are classified as either general “sample-based” or specific “non-sample based alternative”. Examples of general path planning are Probabilistic Roadmap Method (PRM) and Rapidly Exploring Random Tree (RRT). Examples of specific path planning are Cell Decomposition and Potential Field Path Planning. A general technique is often more robust than the alternatives while a specific technique can often be more efficient with faster computation time, as is the case for potential field path planning (PFPP).

1.5 Thesis Contribution

The primary contribution in this thesis is to propose a dGPS-based Path Following approach. The approach uses a motion control technique based on a controller in [16] with the addition of online EKF. A secondary contribution is to incorporate a modified Potential Field Path Planning (PFPP) [4] technique specifically for dGPS localization path following. This approach will be experimentally tested on an outdoor WMR platform.

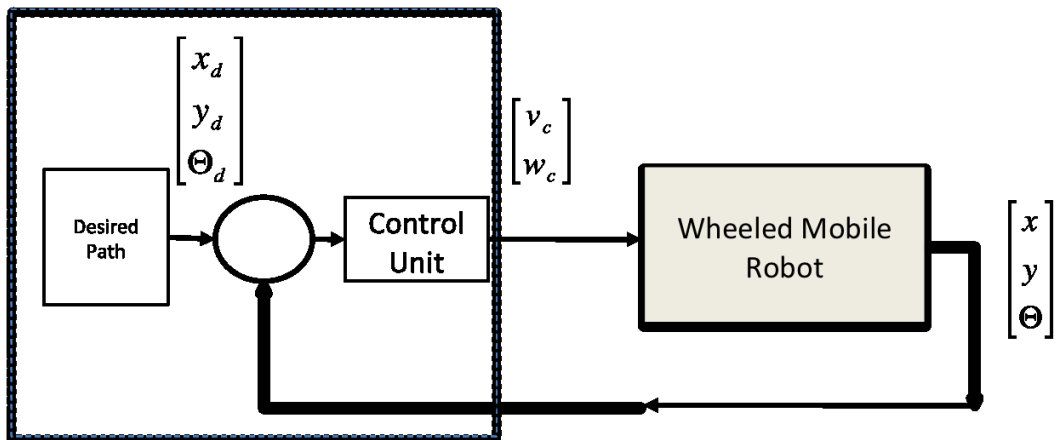


Figure 1.6: General WMR Control with Thesis Focus Highlighted

Fig. 1.6 shows the focus area of this work which includes path planning the desired path, obtaining the current pose, and calculating the wheel control velocities using the control unit. An experimental platform with this system architecture will be set-up to test and validate this approach.

1.6 Thesis Organization

The thesis paper is structured as follows. In Chapter 2 we present a literature review of the previous work done using Potential Field Path Planning(PFPP) and dGPS-based localization for Path Following of WMR. Our proposed dGPS-based Path Following approach for WMR coordination and control is then explained in detail in Chapter 3. Following the explanations, simulations of our approach are conducted in Chapter 4. In Chapter 5, using our WMR mobile platform, we test the approach and show experimental results for a set of cases. These test cases are then also analyzed and compared with simulation. Lastly we conclude in Chapter 6 and suggest some future direction for this work.

Chapter 2

Literature Review of Path Following for Wheeled Mobile Robots

Outdoor WMR research has recently focused on application areas such as agricultural farming [10], reconnaissance and rescue operations [12], and intelligent vehicles [13] [14] [15] [17]. With the emergence of affordable GPS [19] technology and slightly more accurate and expensive differential GPS (dGPS), much research has focused on combining that sensing ability into WMR's localization and path following control. The sections that follow in Chapter 2 consider literature in the area of (1) motion control using dGPS localization sensing, and (2) potential field path planning for path following purposes. In the final discussion section the limitations of both dGPS localization sensors and potential field path planning are discussed.

2.1 dGPS-based Motion Control for WMR in Path Following

Although odometric sensors such as compass, gyros and IMU's are useful in obtaining position information for outdoor and indoor WMR they tend to drift with time and can be noisy. The effect can be more significant for outdoor WMR where terrain may be uneven and, for longer distances, errors may build-up. This drift can result in erred measurements and in turn erred pose estimate. As well as environmental influences, inexpensive less accurate instruments, can also lead to significantly inaccurate position estimate. To decrease this effect and improve position estimate, the integration of GPS and or dGPS sensors as well as using sensor filters with mobile platform (i.e. WMR) is often considered.

Global Positioning system is a type of global navigation satellite systems (GNSS) which was developed by the U.S Department of Defence originally in the 1970's [55]. The GPS system works on the concept of line of sight. With at least 24 GPS satellites in orbit, GPS receivers use four or more satellite to determine their position on the earth's surface. A GPS unit receives standard GPS string data messages such as

National Marine Electronics Association (or NMEA) 0183 data from which a user can obtain position data such as latitude and longitude. The string data also includes other information such as coordinated universal time (UTC), quality of the GPS fix, the number of satellites, and altitude. Table 2.1 shows the different types of GPS NMEA 0183 data strings that can be programmed for a Hemisphere GPS receiver.

Table 2.1: Types of NMEA 0183 Data Strings [66]

Message	Description
GPGNS	Fixes data for single or combined satellite navigation systems
GPGGA	GPS fix data
GPGLL	Geographic position - latitude/longitude
GPGSA	GNSS(Global Navigation Satellite System) DOP and active satellites
GPGST	GNSS pseudo range error statistics
GPGSV	GNSS satellite in view
GPRMC	Recommended minimum specific GNSS data
GPRRE	Range residual message
GPVTG	Course over ground and ground speed
GPZDA	Time and date
GRS	Supports the Receiver Autonomous Integrity Monitoring (RAIM)
RD1	SBAS diagnostic information (proprietary NMEA 0183 message)

The accuracy of a users position is very dependent on number of satellites available and line of sight to the receiver on earth. Given ideal conditions GPS technology can have less than 10 meter accuracy. Differential GPS is an augmentation technique that reduces the GPS positioning error resulting in more accurate position. There

are several dGPS forms such as Space Based Augmentation Systems (SBAS) and Local-Area Differential GPS (LAGPS) that are used for more accurate position measurement. For these specific types of dGPS, the users receiver obtains corrections from a known reference stations along with the GPS satellite position data.

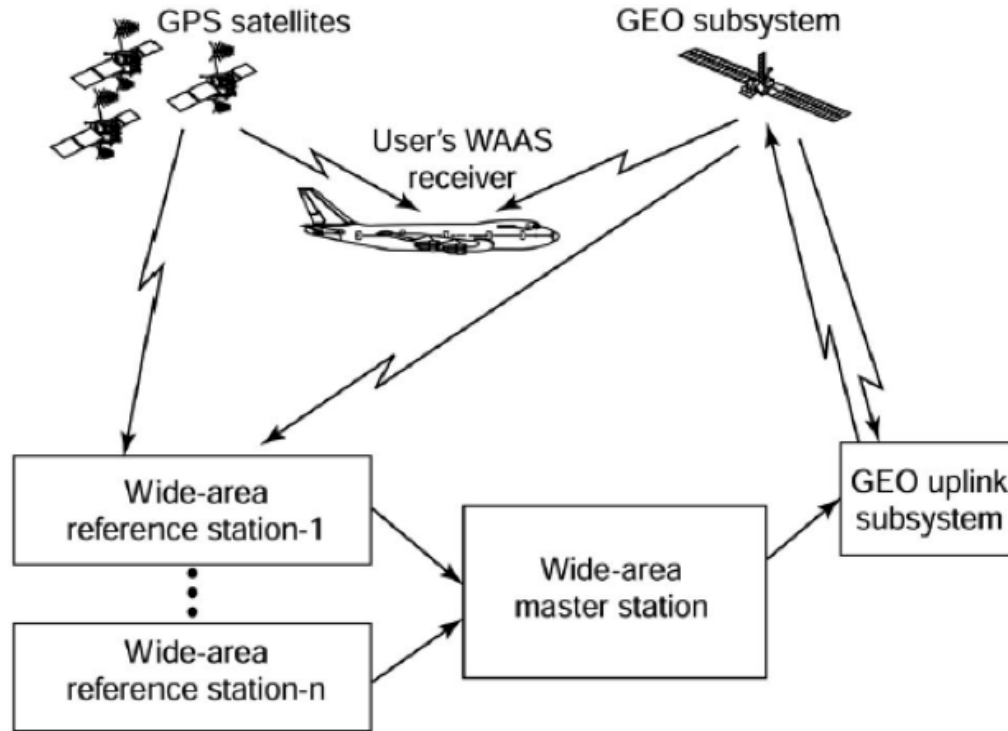


Figure 2.1: Wide Area Augmented System (WAAS)

Fig. 2.1 illustrates the use of SBAS system of differential GPS. This SBAS type is referred to as the Wide Area Augmented System (WAAS) and it is frequently used for commercial aviation purposes. The position accuracy of a WAAS differential GPS is less than 1 meter which is improvement over the normal GPS accuracy. For more accurate position data (less than 0.1 meter) a LAGPS system called 'Real Time Kinematic (RTK)' can be used. The more accurate the GPS sensor the more expensive it cost.

Given the sensor noise and varying pose inaccuracies associated with GPS technologies, estimation techniques are often used for an improved estimate of the position. The most popular approach is to use Bayesian Estimation and Prediction techniques [30] some of which are shown in Table 2.2. These techniques are often

Table 2.2: Examples of Bayesian Techniques used in WMR localization Applications

Name	Abbreviation	Noise Assumption	References
Kalman Filter	KF	Gaussian	[22]
Extended Kalman Filter	EKF	Gaussian	[25] [26] [31] [32] and[14]
Unscented Kalman Filter	UKF	none	[21] [17]
Particle Filter	PF	none	[43]

used to estimate the position based on all sensor measurements. If the WMR collects data from more than one sensor for localization it is referred to as sensor fusion. In [20] this is done using a low cost GPS; In [21] the GPS is combined with IMU measuring devices using a UKF; In [22] the second experiment uses a KF and combines odometry with indoor GPS; In [25] [26] [31] and [14] an EKF is used to integrate dGPS with other sensors; In [32] inexpensive GPS is used in sensor fusion with EKF; In [33] an adaptive KF is used that combines GPS with other sensors.

As is noted in Table 2.2 a Kalman Filter and its variants have frequently been used in research for estimation. The Kalman Filter (KF) [34] [43] was first proposed in 1960 by R.E. Kalman [35] as a filter used in the estimation of discrete linear stochastic noisy systems. Since then, extensive research has been conducted that resulted in producing variants of the Kalman filter such as the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF). The EKF assumptions make it more applicable to linearized nonlinear state space system models in [46] with Gaussian distributed noise. For nonlinear systems with non-Gaussian noise distribution the UKF was proposed in [47].

Given the nonlinear nature of the WMR kinematic model (shown in Equation 1.1) the use of EKF and UKF in localization schemes has recently garnered attention. An EKF architecture, similar to that found in [43], is shown in Table 2.3. The system improved pose estimate \hat{x}_k and covariance P_{x_k} are determined using measurements (z_k) along with predicted ($\hat{x}_k^-, P_{x_k}^-$), and prior ($\hat{x}_{k-1}, P_{x_{k-1}}$) state and covariance matrices respectively.

Table 2.3: EKF Architecture

Given			
	nonlinear stochastic Equation measurement	$x_k = f(x_{k-1}, u_{k-1}, \omega_{k-1})$ where x_k u_k are the state and input vector respectively $z_k = h(x_k, v_k)$	ω_k is random process noise where v_k is random measurement noise
	Initial Conditions /Assumptions	$\hat{x}_0, P_{x_0} = E((x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^{-1})$	State estimate and covariance matrix are known
Prediction Step			
	Compute the process model Jacobians	$F_x = \nabla_x(f(\hat{x}_{k-1}, u_k, \hat{\omega}_{k-1}))$ $G_\omega = \nabla_\omega(f(\hat{x}_{k-1}, u_k, \hat{\omega}_{k-1}))$	F_x associated with the process state variable G_ω associated with the process noise variable
	Compute the predicted state and covariance matrix	$\hat{x}_k^- = (f(\hat{x}_{k-1}, u_k, \hat{\omega}_{k-1}))$ $P_{x_k}^- = F_x P_{k-1} F_x^T + G_\omega R_\omega G_\omega^T$	
Correction Step			
	Compute observation model Jacobian	$H_x = \nabla_x h(\hat{x}_k^-, \hat{v}_k)$	
	Calculate Kalman Gain Update state estimate Update covariance estimate	$K_k = P_{x_k}^- H_x^T (H_x P_{x_k}^- H_x^T + R_v)^{-1}$ $\hat{x}_k = \hat{x}_k^- + K_k [z_k - h(\hat{x}_k^-, \hat{v})]$ $P_{x_k} = (I - K_k) P_{x_k}^-$	

2.2 Potential Field Motion Planning used for Path Following

Potential Field Path Planning is a ‘non-sample’ type of path planning algorithm that can be used for path following purposes. The concept of Potential Field Path Planning (PFPP) was originally proposed by Khatib in [38] as a collision avoidance scheme for a manipulator. When applied to WMR, a robot is treated as a particle with an artificial field that is attracted to the goal but repelled by an obstacle as shown in Fig. 2.3. As a particle, the robot faces attractive F_{att} and repulsive force

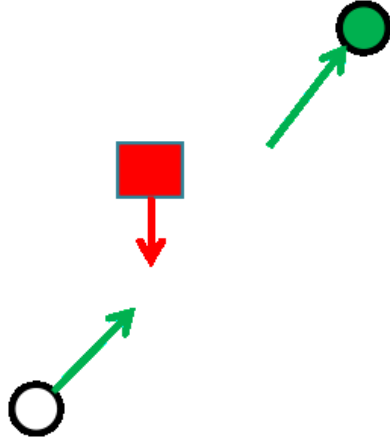


Figure 2.2: Artificial Potential Field Example

F_{rep} due to the goal and obstacle respectively. From [39] [48] the attractive potential field U_{att} and resulting attractive force F_{att} due to the goal are given by Equations (2.1) and (2.2)

$$U_{att} = \begin{cases} \frac{1}{2}\xi\rho_{goal}^2, \rho_{goal} = q - q_{goal} & \text{if } q > q_{goal} \\ 0 & \text{if } q < q_{goal} \end{cases} \quad (2.1)$$

$$F_{att} = -\nabla U_{att} = -\xi(q - q_{goal}). \quad (2.2)$$

where $q = f(x, y)$, ξ is a gain coefficient, and ρ_{goal} is the relative position between robot and goal.

Similarly, the repulsive potential field U_{rep} and resulting attractive force F_{rep} due to the obstacle are given by Equations (2.3) and (2.4)

$$U_{rep} = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho} - \frac{1}{\rho_o}\right)^2 & \text{if } \rho < \rho_o \\ 0 & \text{if } \rho > \rho_o \end{cases} \quad (2.3)$$

$$F_{rep} = -\nabla U_{rep} = -\eta \left(\frac{1}{\rho} - \frac{1}{\rho_o} \right)^2 \cdot \left(\frac{1}{\rho^2} \right) \cdot \frac{q - q_c}{\|q - q_c\|}. \quad (2.4)$$

where $q = f(x, y)$, η is a gain coefficient, ρ is the minimum position between robot and obstacle, and ρ_o is the distance of goal influence.

Summing both attractive and repulsive force vectors results in a total force vector. The total force vector F_{tot} acting on the robot particle governs the path planned for the WMR's motion. The particle moves a step size δ_i , in the direction indicated by the resultant force calculated from Equation (2.5).

$$F_{tot} = F_{att} + F_{rep} \quad (2.5)$$

Table 2.4: PFPP Pseudo Code Example (Similar to [48])

Given:	q_{start}, q_{goal}	where $q = f(x, y)$, $i = 0, q_0 = q_{init}$	
1.	If	$q_i \neq q_{goal}$	
2.	else stop	Solve for: then	\vec{F}_{tot} and $\ \vec{F}_{tot}\ $ $q_{i+1} = q_i + \delta_i \frac{\vec{F}_{tot}}{\ \vec{F}_{tot}\ }$
3.	Iterate:	$i = i + 1$	repeat step 1

This process, shown in Table 2.4, is iterative and continues until the goal is reached as shown in Fig. 2.3 or the particle gets stuck in a local minimum position. When a robot particle gets stuck in a local minimum scenario the robot has a minimum force value equal to the absolute minimum value associated with reaching goal position. Yet in terms of location the position does not coincide with the goal position. Various strategies in literature have been proposed for dealing with this issue.

2.3 Discussions

GPS and dGPS technology is very dependent on number of satellites and Line of Sight to GPS satellites. Urban environments with tall buildings can influence the data collected. dGPS improvements are also dependent on the proximity of the receiver to the ground reference station. The longer the distance the less improved the accuracy. Although dGPS receiver obtains more accurate position data when

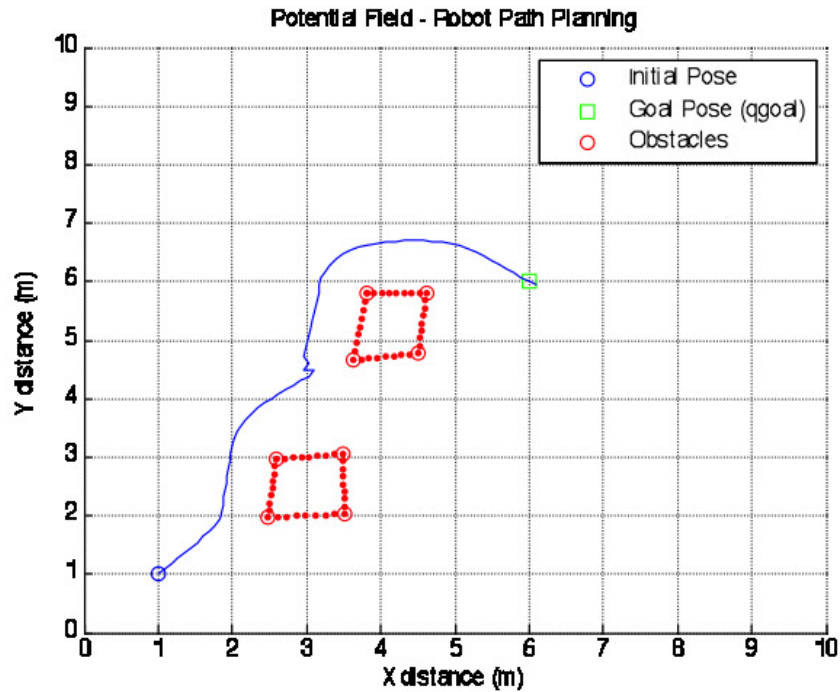


Figure 2.3: Artificial Potential Field Path Planning (PFPP)

compared to a normal GPS, it still contains some level of inaccuracy (less than $1m$). This inaccuracy in measurement can influence localization results especially in cases of small distance travel. Therefore, using an estimation technique from Table 2.3 for these cases, can still provide an improved estimate of position. Given the nonlinear nature of the WMR kinematics and the uneven terrain associated with outdoor WMR, using KF and EKF techniques which assume linearized models and Gaussian noise has limitations. Although the use of UKF and PF with no noise assumptions (noted in Table 2.2) can result in better estimate, the tradeoff is increased computation time. This time increase can then result in a slower response for the WMR.

Due to its simplicity and speed of computation PFPP is very advantageous for online and offline obstacle avoidance. An issue that arises with PFPP is dealing with local minimums. Several solutions are proposed in literature, such as using simulated annealing technique proposed by Park et. al in [49] or considering the path connecting the start and the goal as a rubber band in order to avoid local minimum scenarios proposed by Tang et. al in [50]. Sekiguchi et. al. in [51] proposes a Lyapunov candidate that covers the non-goal local minimum cases thereby erasing them and

ensuring only goal position is the minimum. Shi et. al. in [39] proposes an improved repulsive and attractive potential functions to overcome local minimums. All these techniques are useful in dealing with local minimum issues with PFPP.

2.4 Chapter Summary

In this chapter we reviewed literature that focuses on two areas: motion control using dGPS localization sensing, and potential field path planning for path following. A general overview of dGPS localization sensor is presented and the previous research work on using dGPS localization for WMR is explored. Similarly as part of Section 2.2 the PFPP algorithm is shown in detail. Finally in the last section the benefits and limitations of both dGPS localization and PFPP algorithm are discussed.

In the next chapter we propose a dGPS based approach that encompasses the two area focused on in this chapter. The aim of our work is tackle dealing with a more accurate GPS sensor as the primary (and only) localization sensor for Outdoor WMR. Although much research has been conducted with using GPS and odometric localization sensors, and similarly using Kalman Filters (KF) and its variants for estimation of WMR positions, there is not many experimental research work focused on using dGPS as a primary sensor combined with an EKF estimator. Similarly, incorporating path planning algorithm specifically to address sensor localization issues has yet to be explored as an integral part of path following.

Chapter 3

Proposed dGPS-based Path Following Approach

In this chapter, a dGPS-based approach for path following is proposed. It builds upon some of the previous literatures discussed in Chapter 1 and 2. In the problem formulation Section we consider two issues that this approach is aiming to address and contribute upon. This is followed by the dGPS-based path following Section where the system architecture of this approach is presented and each component is elaborated upon. The use and extension of this approach to other applications such as multiple robots is then briefly investigated. Finally the Chapter is concluded with a summary discussion.

The contribution focus of this work is on dealing with a more accurate GPS sensor as the primary (and only) localization sensor for Outdoor WMR. Much WMR research has been conducted with GPS and odometric localization sensors, and similarly the use of Kalman Filters (KF) and its variants for estimation of WMR positions. To our knowledge, not many experimental research works are out there that consider dGPS as a primary sensor combined with an EKF estimator. Similarly, given the minimal inaccuracies associated with the dGPS alone, path planning algorithm specific to sensor localization issues has yet to be explored as an integral part of path following.

3.1 Problem Formulation

dGPS-based localization for WMR's with KF or its variant (EKF,UKF) has been researched in the following literatures [14] [17] [25] [26] [31]. In [31] it has been shown that sensor fusion with a more accurate dGPS (vs. GPS) produces a better localization estimate. Criteria such as urban environment influences and multiple robots use of dGPS signals have also been researched in [26] and [17] respectively.

An area that this thesis aims to investigate and contribute on is how dGPS as a primary sensor performs in localization of WMR. Although dGPS has been integrated with other odometric sensors such as IMU and compass for localization ([21] [31])

not much works exist that consider dGPS only primary sensor or dGPS with dead reckoning ‘encoder only’ sensor fusion. Another secondary focus this thesis aims to incorporate is path planning scheme as part of dGPS-based Path Following. For the path following purposes, a desired path is often generated by a path planning algorithm [4]. Based on its simplicity and computation speed, the potential field path planning (PFPP) is ideal for the proposed dGPS-based path following approach.

3.2 dGPS-based Path Following

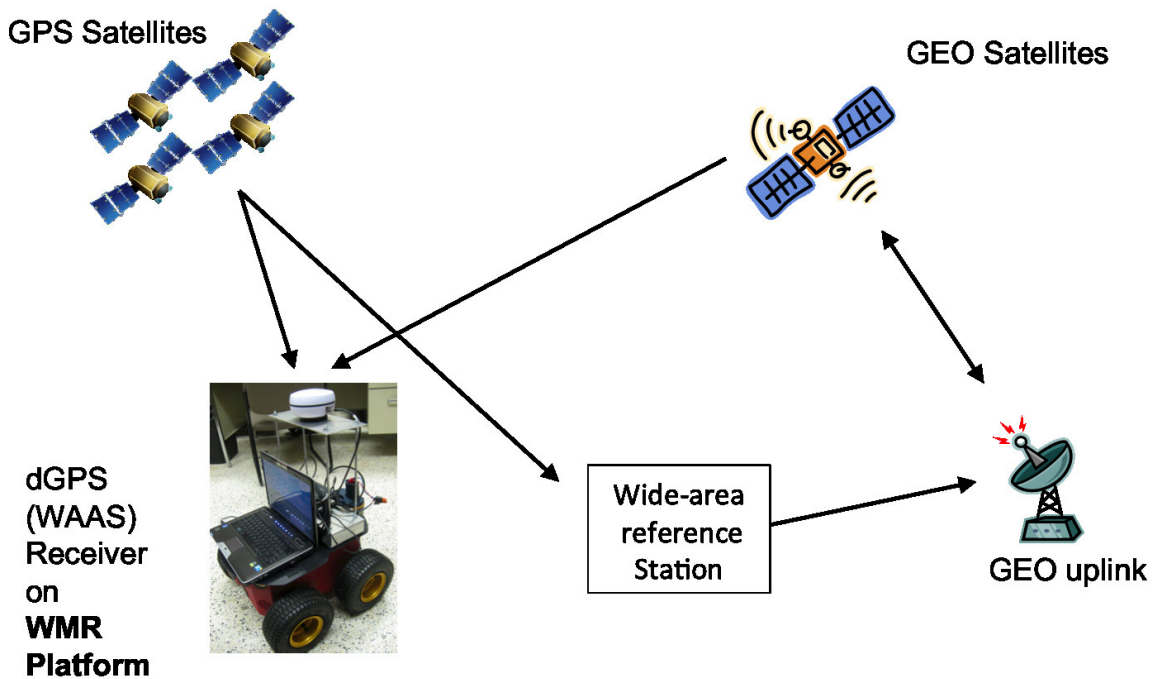


Figure 3.1: WMR with WAAS dGPS receiver

The path following approach is composed of (1) a modified PFPP and (2) a motion control based on Extended Kalman Filtered (EKF) pose estimates from the dGPS localization sensor measurements. The differential GPS technology used is a WAAS type and has an accuracy of less than $1m$. The WAAS receiver in the set-up shown in Fig. 3.1 obtains GPS string data from the GPS satellites as well as the reference ground station Geosynchronous (GEO) GPS data. The receiver, using the GEO GPS data, applies corrections to its GPS satellite data to obtain a more accurate dGPS position.

3.2.1 System Architecture

The path following control architecture is shown in Fig. 3.2. The path $(x_{des}, y_{des}, \theta_{des})$

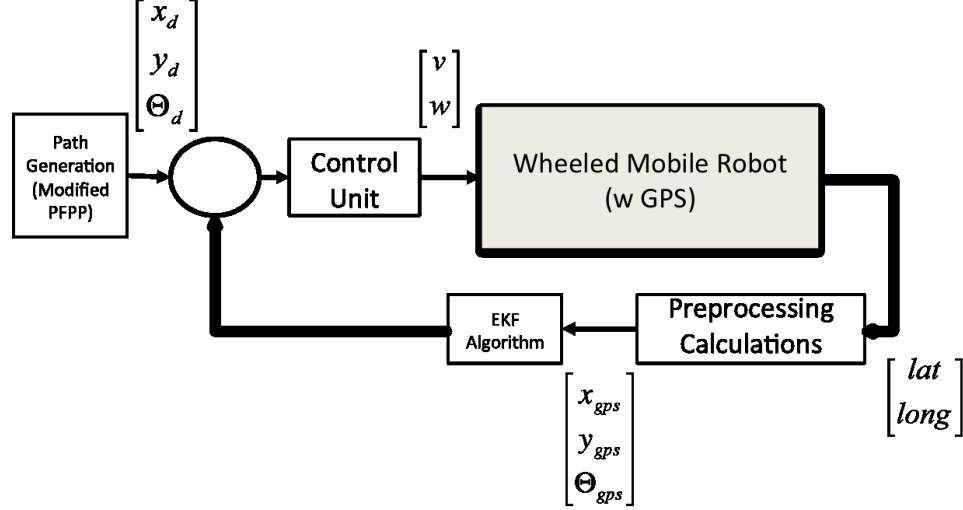


Figure 3.2: Path Following Flow Control Architecture

is generated using the modified PFPP offline. This path is then used during WMR motion as a reference for path following. From the dGPS signal we obtain $(latitude, longitude)$ pose information. Using the Haversine formula [45] and given a reference coordinate, (x_0, y_0) (corresponding to $(lat_0, long_0)$), the $(latitude, longitude)$ values are then converted to a (x_{gps}, y_{gps}) . The Haversine formula, shown in Equation (3.1), is very useful for calculating the distance between two points on a circular surface like the earth.

$$\begin{aligned}
 \delta_{lat} &= lat_2 - lat_1 \\
 \delta_{long} &= long_2 - long_1 \\
 a &= \sin^2\left(\frac{\delta_{lat}}{2}\right) + \cos(lat_1) \cos(lat_2) \sin^2\left(\frac{\delta_{long}}{2}\right) \\
 c &= 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a}) \\
 d &= R \cdot c
 \end{aligned} \tag{3.1}$$

where the earth radius $R = 6371$ km, and the angles are in radians.

Unlike the Spherical Law of Cosines, the Haversine formula works well even for small distances which are the cases considered in this thesis. As the WMR moves, angle θ_{gps} for each time instant is calculated using previous WMR position data.

With $(x_{des}, y_{des}, \theta_{des})$ and $(x_{gps}, y_{gps}, \theta_{gps})$ known for a given time instance, an EKF is used to estimate the robot's current state and calculate the wheel control velocity values (v_c, w_c) required to move it to its desired future state. These control velocities are feed through an internal low level PID controller to produce the desired torque and actuate the wheel motors to achieve the wheel velocities.

3.2.2 Modified Potential Field Path Planning (PFPP) Algorithm

The desired path is generated offline using slight variant of the PFPP proposed by Khatib et al [38]. Unlike the improved PFPP proposed by Shi et al. [39] that is more efficient in dealing with local minimum, the proposed PFPP scheme does not focus on local minimum cases for this thesis. If a local minimum is encountered the solution used is based on a less efficient 'weighted random choice' alternative as shown in Fig. 3.3. In this solution, a new artificial secondary goal (X_i, Y_g) or (X_g, Y_i) is randomly chosen to overcome the local minimum case. If it proves successful the goal is then reassigned to be the primary (X_g, Y_g) . If it proves unsuccessful the third set of artificial goals $(X_i - X_g, Y_g)$ or $(X_g, Y_i - Y_g)$ are tested to try to overcome getting stuck at the local minimum.

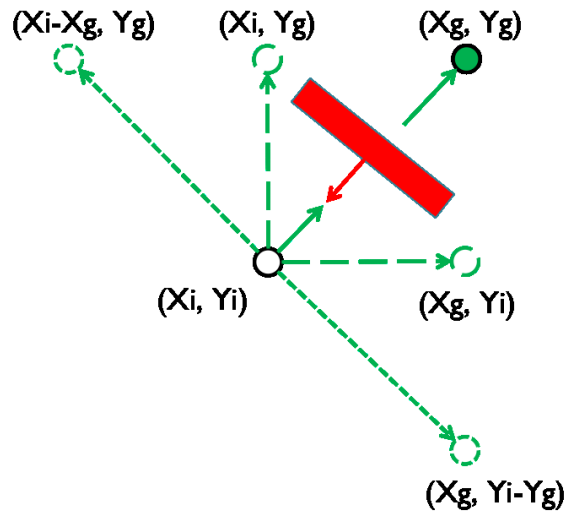


Figure 3.3: Weighted local minimum solution

The following proposed approach considers the step size δ argument in Table 2.4. Given that the areas or objects of interest during motion are q_{start} , q_{goal} , as well as q_o in the vicinity of the obstacle, this approach aims to include more points along the path

within these radius boundaries. In concentrating more points in the area of interest we ensure that, even with dGPS level inaccuracies, the robot start pose and goal pose are known well. The technique also ensures that an obstacle can be avoided. To accomplish this the step sizes are adjusted when the robot particle transits into these areas of interest. The technique uses geometric progression mathematical equation $\delta_n = \delta_{const} \cdot r^{n-1}$ where δ_n is the step size, and r is a constant ratio. The step size δ_n depends on the proximity of the robot to the boundary of influence of the start, obstacle and goal positions.

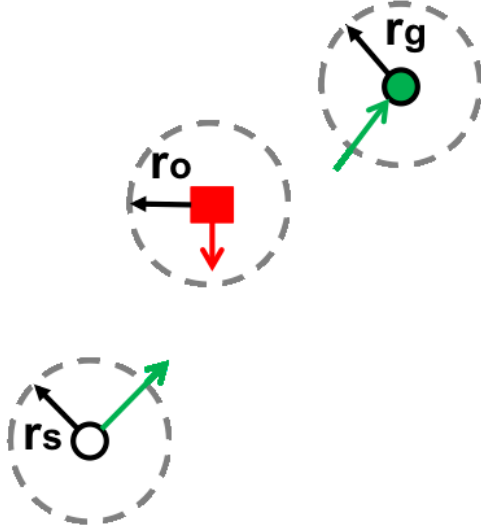


Figure 3.4: PFPP with boundary of influence to the object of interest

The artificial potential field depicted in Section 2.2 Fig. 2.3 therefore becomes Fig. 3.4. In this case the WMR is still considered a circular point particle attracted to the circular goal and repelled by the square obstacle but all the start, obstacle, and goal positions have boundaries of interest associated with them. The boundary is shown as a circular external boundary (r_s, r_o, r_g) . This boundary dictates the activation of the geometric progression PFPP shown in Table 3.1.

Table 3.1 shows the pseudo code logic for the geometric progression PFPP proposed algorithm. The result is a smaller size δ_n is used inside the radii, and then once out of the boundary, the standard larger step size δ_{const} is used.

Table 3.1: Geometric Progression PFPP Pseudo Code

Given:	q_{start}, q_{goal}	where $q = f(x, y)$, $q_0 = q_{init}$	and where $i = 0, j = 5, r = 0.5$
1.	If	$q_i \neq q_{goal}$	
		If	$\rho < \rho_{influence},$ $\delta_j = \delta_{max} \cdot r^{j-1} >= \delta_{min}$ $j = j - 1 >= 1$
		else	$\delta_j = \delta_{min} \cdot (r^{-1})^{j-1} <= \delta_{max}$ $j = j + 1 <= 5$
		end	
		Solve for: then	\vec{F}_{tot} and $\ \vec{F}_{tot}\ $ using Eqn. (2.1)-(2.5) $q_{i+1} = q_i + \delta_i \frac{\vec{F}_{tot}}{\ \vec{F}_{tot}\ }$
2.	else	stop	
3.	Iterate:	$i = i + 1,$	repeat step 1

3.2.3 Numerical Methods (NM) Based Algorithm

The controller design we use in this proposed approach is a modification of the one proposed by Gustavo et. al. in [41]. It is a numerical methods approach based on the the controller proposed by Kanayama et al. [16]. In [41], four WMR controller are proposed. Each controller depends on how much information is available regarding the desired path $(x_{des}, y_{des}, \theta_{des})$ and speed (v_{des}, w_{des}) for the WMR path. In [41], these controllers are used for WMR indoor path following cases based on odometry sensors. For our purposes we are using a modification of one of the four controllers for Outdoor WMR path following with dGPS. The outdoor environment terrain and use of dGPS measurement sensor provide a new dynamic to be considered in simulation and testing.

From Equation (1.1), we know the kinematic model for an ideal Type (2,0) WMR. If we consider process model noise due to uneven terrain and non-slipping criteria this equation can be written as:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \sin(\theta) & 0 \\ \cos(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} + \Delta. \quad (3.2)$$

where Δ is the process noise variable.

Using Euler's first order integration approximation and assuming non-zero process noise [46], the vectorial form of the kinematic model shown in Equation 3.3 can be obtained.

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \\ \theta_{n+1} \end{pmatrix} \approx \begin{pmatrix} x_n \\ y_n \\ \theta_n \end{pmatrix} + T_s \begin{pmatrix} \sin(\theta) & 0 \\ \cos(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_n \\ w_n \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{pmatrix}. \quad (3.3)$$

where T_s is the sampling time.

To solve this equation for the linear v_n and angular w_n control velocity expressions the desired trajectory should be known. The current (x_{des}, y_{des}) and future (x_{des+1}, y_{des+1}) desired pose can be calculated based on the geometric progression PFPP algorithm shown in Section 3.2.2. Using these two points and Equation (3.4), the orientation θ_{des+1} can also be calculated.

$$\theta_{des+1} = \arctan 2\left(\frac{y_{des+1} - y_{des}}{x_{des+1} - x_{des}}\right). \quad (3.4)$$

After the desired trajectory $(x_{des+1}, y_{des+1}, \theta_{des+1})$ is known, it can then be substituted into Equation (3.3) for $(x_{n+1}, y_{n+1}, \theta_{n+1})$.

Solving for v_n and w_n using the technique shown in [41] we obtain Equation (3.5)

$$\begin{pmatrix} v_n \\ w_n \end{pmatrix} = \begin{pmatrix} k_v \left(\frac{x_{des+1} - (x_n + \Delta x)}{T_s} \cos(\theta_{des+1}) + \frac{y_{des+1} - (y_n + \Delta y)}{T_s} \sin(\theta_{des+1}) \right) \\ k_w \frac{\theta_{des+1} - (\theta_n + \Delta \theta)}{T_s} \end{pmatrix}. \quad (3.5)$$

where constants k_v and k_w are positive controller gains between [0,1] that control the performance of the controller. This formulation is similar to Equation (15) in [41] and the Equation proposed by Kanayama et. al [16]. It ensures that the desired path is the target trajectory to be followed. The discrete nature of the proposed architecture and simplicity of the controller makes this controller desirable for use in this scheme. Also the platform on which this controller was tested on, Pioneer 2DX indoor WMR, is similar to the Pioneer 3AT outdoor WMR used in this thesis work.

3.2.4 Extended Kalman Filter (EKF)

The EKF has previously been used in WMR pose estimation research as noted in Chapter 2. For the discrete controller shown in Equation (3.5) the Extended Kalman

Filter pose estimate can be applied and used for smoother control. The EKF architecture steps shown in Table 2.3 are used to implement the EKF on a WMR kinematic model shown in Equation (3.4). For our application these steps include: (1) identifying the initial conditions for the pose state estimate $\hat{\mathbf{x}}_0$ and state covariance matrix P_0 given, then (2) using a known WMR model, predicting the next (or future) state estimate $\hat{\mathbf{x}}_{k+1}^-$ and covariance $P_{x_{k+1}}^-$, and finally (3) with the predicted variables, as well as the dGPS measurement and measurement covariance matrix at time T_{k+1} , solving for the improved estimates $(\hat{\mathbf{x}}_{k+1}, P_{x_{k+1}})$.

Given

The given initial state estimate and covariance are shown in Equation (3.6)

$$\hat{\mathbf{x}}_0 = \begin{pmatrix} \hat{x}_0 \\ \hat{y}_0 \\ \hat{\theta}_0 \end{pmatrix}, P_k = E((x_k - \hat{x}_k)(x_k - \hat{x}_k)^{-1}). \quad (3.6)$$

where $k = 0$.

Prediction Step

The prediction step considers the kinematic (or process) model shown in Equation (3.2) where $\Delta_k = [\Delta x, \Delta y, \Delta \theta]$ is an independent bounded process noise vector. The process jacobians calculated as part of this step are G_{Δ_k} and $F_{\hat{x}_k}$. The jacobian variable associated with the process noise G_{Δ_k} therefore becomes:

$$G_{\Delta_k} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.7)$$

While the $F_{\hat{x}_k}$ jacobian term becomes:

$$F_{\hat{x}_k} = \begin{pmatrix} 1 & 0 & -T_s \cdot v \sin(\hat{\theta}_k) \\ 0 & 1 & T_s \cdot v \cos(\hat{\theta}_k) \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.8)$$

Since the process noise Δ_k is bounded and independent a covariance matrix R_{Δ} exists for this vector. The form of this non-zero process noise covariance matrix is shown

in Equation (3.11).

$$R_{\Delta} = \begin{pmatrix} R_{\Delta x} & 0 & 0 \\ 0 & R_{\Delta y} & 0 \\ 0 & 0 & R_{\Delta \theta} \end{pmatrix}. \quad (3.9)$$

where $R_{\Delta x}$, $R_{\Delta y}$, $R_{\Delta \theta}$ are the process covariances associated with process noise variables Δx , Δy , $\Delta \theta$ respectively. As is noted in previous literature, the starting point for these values is very hard to obtain prior to testing therefore the values were all set to zero initially (i.e. $R_{\Delta x} = 0$, $R_{\Delta y} = 0$, $R_{\Delta \theta} = 0$). These zero parameter values are applied in Chapter 4 simulations and Chapter 5 preliminary experimental tests.

Lastly in the prediction step, the predicted state estimates and covariance matrix are computed using Equation (3.10) and (3.11) respectively.

$$\begin{pmatrix} \hat{x}_{k+1}^- \\ \hat{y}_{k+1}^- \\ \hat{\theta}_{k+1}^- \end{pmatrix} \approx \begin{pmatrix} \hat{x}_k \\ \hat{y}_k \\ \hat{\theta}_k \end{pmatrix} + T_s \begin{pmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} v_k \\ w_k \end{pmatrix}. \quad (3.10)$$

$$P_{x_{k+1}}^- = F_{\hat{x}_k} P_k F_{\hat{x}_k}^T + G_{\Delta_k} R_{\Delta_k} G_{\Delta_k}^T. \quad (3.11)$$

Correction Step

Initially as part of corrections step matrix parameters H_k and R_v are defined. These values along with the predicted covariance are then used to obtain the Kalman gain K_k . The Kalman gain along with dGPS measurement pose values are then used to update the estimates. Since all states are assumed to be measurable the jacobian of $h(x_k, v_k)$ is equal to an identity matrix as shown in Equation (3.12) below.

$$H_k = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (3.12)$$

R_v is non-zero measurement noise covariance matrix shown in Equation (3.13).

$$R_v = \begin{pmatrix} \sigma_{x_{gps}}^2 & 0 & 0 \\ 0 & \sigma_{y_{gps}}^2 & 0 \\ 0 & 0 & \sigma_{\theta_{gps}}^2 \end{pmatrix}. \quad (3.13)$$

where $\sigma_{x_{gps}}, \sigma_{y_{gps}}, \sigma_{\theta_{gps}}$ are the standard deviation of x, y, θ . To determine the R_v 's matrix parameters 60 stationary dGPS data points were first collected. This stationary data set was collected in the WMR test environment with a Crescent R110 Hemisphere dGPS sensor. A plot of the the R_v parameter dataset is shown in Figures 3.5 to 3.7. These plotted values correspond to $\sigma_{x_{gps}}, \sigma_{y_{gps}}, \sigma_{\theta_{gps}}$ equal to 0.0073m, 0.00621m, 0.705rad respectively. While this measurement covariance R_v data is assumed constant and used for all the simulations and tests, it should be noted that as the robot moves the measurement covariance may vary from location to location.

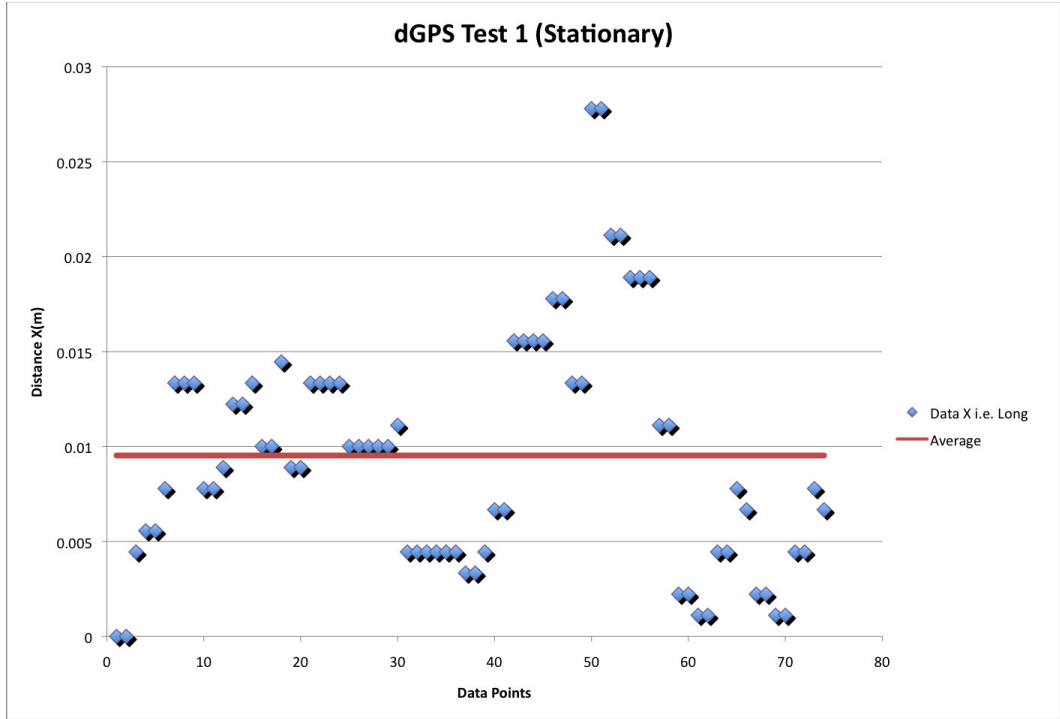


Figure 3.5: Variance Data for $\sigma_{x_{gps}}$ variable in Measurement Covariance Matrix R_v

Using the previous parameters and predicted state, the kalman gain K_k is calculated with Equation (3.14).

$$K_k = P_{x_{k+1}}^- H_k^T (H_k P_{x_k}^- H_k^T + R_v)^{-1} \quad (3.14)$$

Finally to obtain the improved estimates $(\hat{x}_{k+1}, P_{x_{k+1}})$ Equation 3.15 is used.

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= \hat{\mathbf{x}}_{k+1}^- + K_k [z_k - h(\hat{\mathbf{x}}_{k+1}^-, \hat{v})] \\ P_{x_{k+1}} &= (I - K_k) P_{x_{k+1}}^- \end{aligned} \quad (3.15)$$

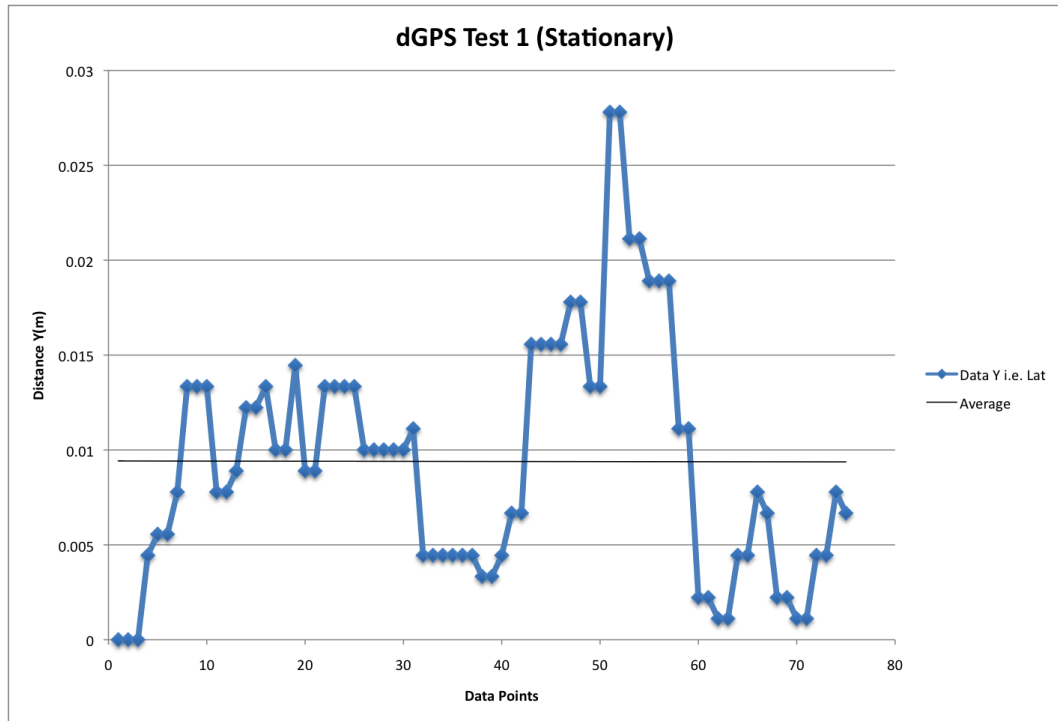


Figure 3.6: Variance Data for $\sigma_{y_{gps}}$ variable in Measurement Covariance Matrix R_v

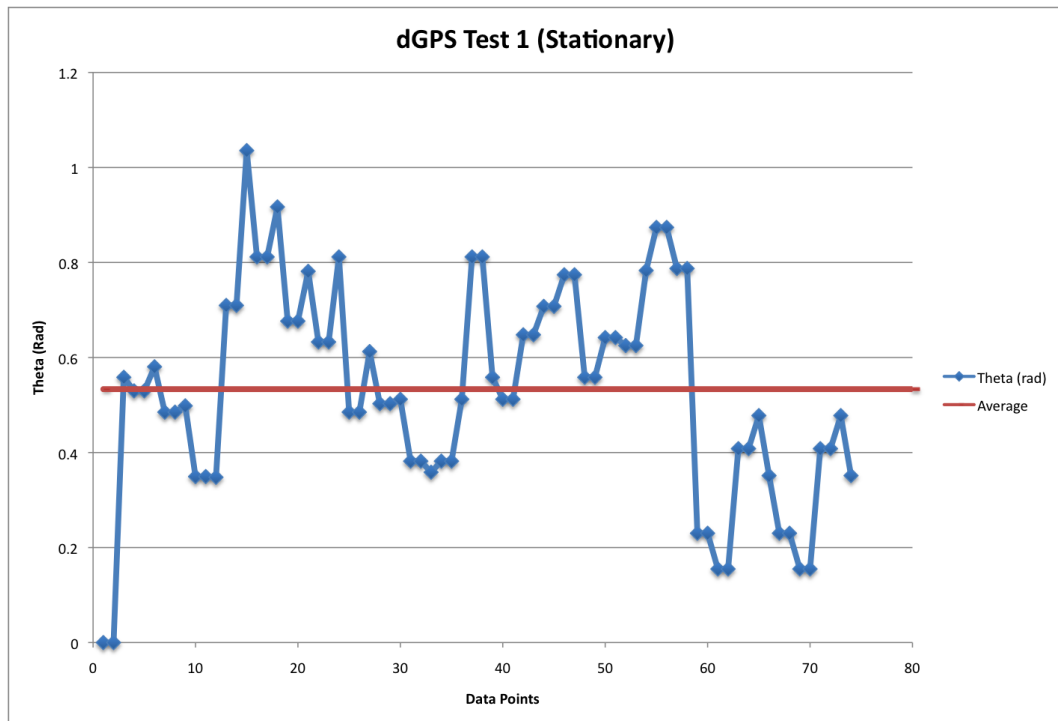


Figure 3.7: Variance Data for $\sigma_{\theta_{gps}}$ variable in Measurement Covariance Matrix R_v

where z_k is associated with the dGPS measurement $(x_{gps}, y_{gps}, \theta_{gps})$. And the θ_{gps} term in z_k is calculated using Equation (3.4) rather than using an additional heading sensor such as compass. This θ_{gps} value is obtained using the two consecutive points (x_{gps}, y_{gps}) and (x_{gps-1}, y_{gps-1}) and Equation (3.4).

Using the complete online EKF, the final form of control law from Equation (3.5) then becomes:

$$\begin{pmatrix} v_{n+1} \\ w_{n+1} \end{pmatrix} = \begin{pmatrix} k_v \left(\frac{x_{des+2} - \hat{x}_{k+1,1}}{T_s} \cos(\theta_{des+2}) + \frac{y_{des+2} - \hat{x}_{k+1,2}}{T_s} \sin(\theta_{des+2}) \right) \\ k_w \frac{\theta_{des+2} - \hat{x}_{k+1,3}}{T_s} \end{pmatrix}. \quad (3.16)$$

where $(\hat{x}_{k+1,1}, \hat{x}_{k+1,2}, \hat{x}_{k+1,3})$ are the $(\hat{x}_{n+1}, \hat{y}_{n+1}, \hat{\theta}_{n+1})$ respectively.

3.3 Extension to Multiple Robots

The previous two sections of this chapter have concentrated on defining the proposed approach for a single WMR. In this section we consider how this approach may be extended to the path following control of more than one (or multiple) WMRs. In extending the dGPS-based path following approach to multiple WMRs both path generation by the proposed modified PFPP Algorithm and motion control based on dGPS localization must be considered. The current proposed path planning component to the approach consider ‘prioritization only’ for generating a path. A path is generated for the first assigned WMR (Fig. 3.8(a)) and then a path is generated for the WMR’s that follow using the previous paths as obstacles (Fig. 3.8(b)). This is unlike the PFPP for multiple WMR particles proposed in previous literature [42] which considers both space and time as well as prioritization. Similarly based on the multiple robotic control scheme, which can either be cooperative or distributive control, the EKF can still be applied for pose estimation and motion control. An example of estimation technique being applied to multiple WMR can be found in [17] where UKF is applied.

A big concern when dealing with multiple WMR is developing online avoidance schemes and communication network between WMR to ensure collision-free path following. Some works that address these issue for multiple robots are [15], [17] and [44]. Although challenges exist when considering multiple WMR, multiple robot control is feasible and ‘doable’. Certainly this proposed approach, with some modifications, can be translated to multiple WMR use.

3.4 Chapter Summary

In the following chapter we proposed a dGPS-based path following approach that uses a modified PFPP and a NM motion control based on Extended Kalman Filtered (EKF) pose estimates. The aim of the approach is to handle the minimal inaccuracies that might be associated with using dGPS sensing as the primary (and only) localization sensor for Outdoor WMR path following control. The system architecture is outlined for the approach followed by a comprehensive explanation of the path planning and motion control components. Finally the use of this approach for Multiple WMR control is briefly explored at the end of this chapter.

In the next chapter, we simulate several cases of this path following approach for a single WMR. The first set of simulations are to test the functionality of the approach for linear and curvilinear trajectory paths. The later simulations are experimental test cases which are more specific and geared to compare with the experimental work presented in Chapter 5.

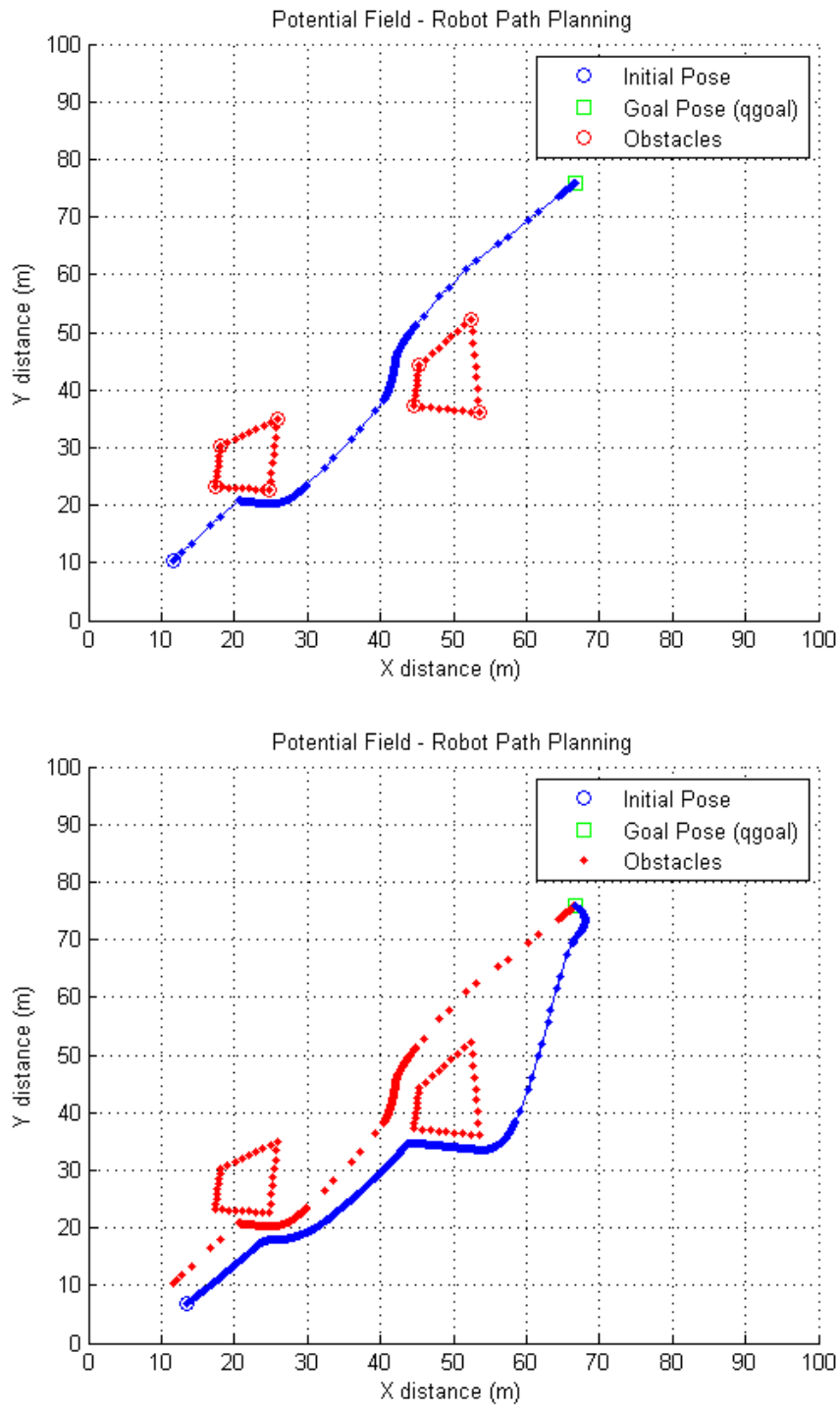


Figure 3.8: Applying Geometric Progression PFPP to Multiple WMR (number of robots = 2)

Chapter 4

Simulation Results

The simulations for the proposed algorithm presented in the previous Chapter are conducted using both Matlab 7.11 and MobileSim 0.5.0 simulation environments. The kinematic WMR's model is programmed in Matlab and simulations are run for both the motion planning and the motion control parts of the proposed algorithm. MobileSim, shown in Fig. 4.1, is a Pioneer MobileRobots Simulator software that is useful to test C++ executable program. This software package is a useful debugging tool to check aspects of program executable code on a simulated Pioneer WMR prior to applying the algorithm to the experimental Pioneer WMR platform for testing.

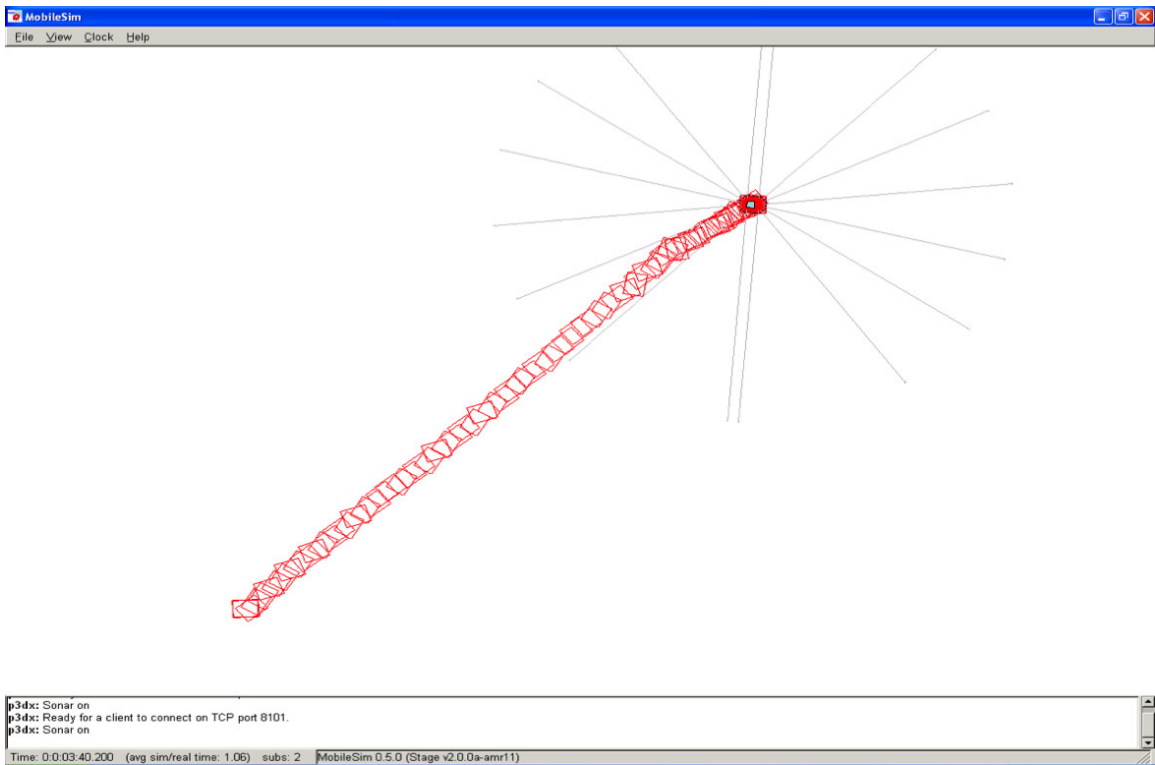


Figure 4.1: MobileSim Simulation (screen capture) for a Linear Path using Odometric Sensor

The simulations that follow focus on the two areas of the proposed Path Following

algorithm: (1) Motion planning based on geometric progression (or modified) PFPP and (2) Motion control using the Numerical Methods (NM) controller with Extended Kalman Filter (EKF).

4.1 Motion planning based on geometric progression (or modified) PFPP Method

The cases considered for the proposed geometric progression PFPP are (1) linear trajectory without obstacles, and (2) curvilinear trajectory with obstacles. Considering these cases allows for future algorithm expandability as most planned paths can be a combination of both linear and curvilinear trajectories. The path planning distance travel at each time step is dictated by maximum δ_{max} (and minimum δ_{min}) step sizes. These calculated values ensure the WMR control velocities do not exceed the maximum values (v_{max}, w_{max}) . The values also ensure the WMR can achieve motion within the sample time period T_s .

In order to better plan a path in a real world environment, a pixel map image is also incorporated in the latter simulations to aid in charting the path of the robot in its test environment. The pixel image, shown in Fig. 4.6 for example, is a Bing image [64] on which path planned simulations are overlaid. The image map's four corner pixel coordinates match the corresponding GPS coordinates for those specific locations. This map image enables a path to be planned that considers the test site location. The test environment is the Dalhousie University Sexton Campus parking lot where all the experimental test cases were conducted.

4.1.1 Simulation Results

A. Modified PFPP Linear and 'Obstacle Filled' Curved Path

Linear trajectory path planning using the proposed modified PFPP is simulated for a straight line with a distance travel of less than $20m$. The generated straight line path follows the technique from Table 3.1 in Chapter 3, where the modified PFPP gain coefficients, from Equations (2.1)-(2.5), are $\xi = 1$ and $\eta = 10$. The boundary of influences which are depicted in Chapter 3 Fig. 3.4 for this linear trajectory case are $r_s = r_g = 0.6m$ and $r_o = 0m$ (since there is no obstacle). The maximum δ_{max} and

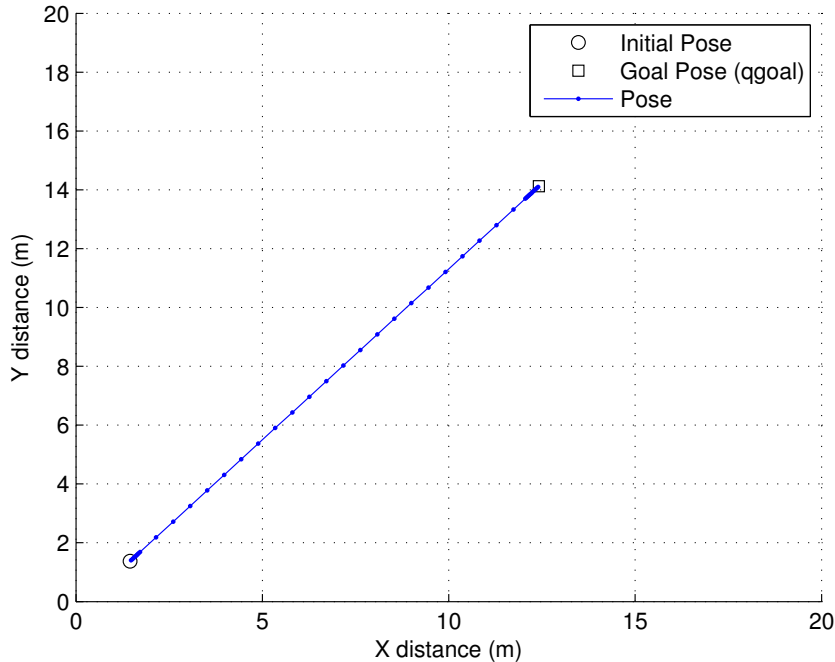


Figure 4.2: Modified PFPP Linear Path 1

minimum δ_{min} step sizes are $0.07m$ and $0.004375m$ respectively. The path planning results of the simulation are shown in Fig. 4.2. Fig. 4.3 shows the corresponding artificial resulting force (F_{tot}) responsible for modified PFPP path generation. The artificial force is initially larger and decreases to zero as the robot ‘particle’ approaches the goal.

A curvilinear trajectory path planning of less than $20m$ travel, is generated using the proposed modified PFPP. As was the case with the linear trajectory path planning, the maximum δ_{max} and minimum δ_{min} step sizes for curvilinear trajectory path planning are $0.07m$ and $0.004375m$ respectively. Similarly the modified PFPP gain coefficients are $\xi = 1$ and $\eta = 10$. For this simulation case though, the boundary of influences are changed compared to the previous linear trajectory with $r_s = 1.0m$, $r_g = 1.5m$, and $r_o = 2.5m$. The simulation resultant planned path is shown in Fig. 4.4, and is based on artificial force result shown in Fig. 4.5. The artificial force vector at each time step results in a path generated in a direction that avoids the obstacle as the particle moves from start to goal position. Similar to linear path in Fig. 4.3, the artificial force for a curved path is initially larger and decrease to zero as the robot

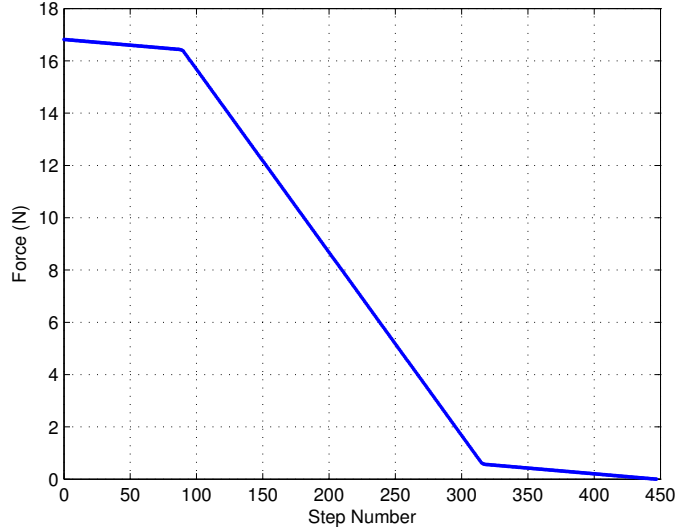


Figure 4.3: Artificial Resulting Force (F_{tot}) for Modified PFPP Linear Path 1

‘particle’ approaches the goal. As the robot particle moves close to an obstacle the artificial force experiences a relative minimum (or valley). For the case considered in the simulation, the relative minimum artificial force does not reach the absolute minimum force associated with the goal position. If however the relative minimum did approach absolute minimum force values, the local minimum strategy proposed in Section 3.2.2 and depicted in Fig. 3.3 would be invoked.

Comparing the linear and curved paths of similar distance travel, the step number is significantly greater for the curved ‘obstacle-filled’ path (Fig. 4.5) than for the linear path (Fig. 4.3) which is in part due to the obstacle.

B. Modified PFPP with Map Image Overlay

In the following simulation a Bing map image is used to aid in path planning purposes. The simulation is for a curvilinear trajectory path with a travel distance greater than $20m$. The threshold boundaries of start, obstacle and goal boundaries of interest are increased from the previous section to $r_s = 2.0m$, $r_g = 2.0m$, $r_o = 2.0m$. Since the robot particle is traveling a longer distance travel, the maximum δ_{max} and minimum δ_{min} step sizes for curvilinear trajectory path planning are increased to $0.35m$ and $0.0219m$ respectively. Similarly the modified PFPP gain coefficients for longer distances are changed to $\xi = 1$ and $\eta = 800$. The path planning simulation result is

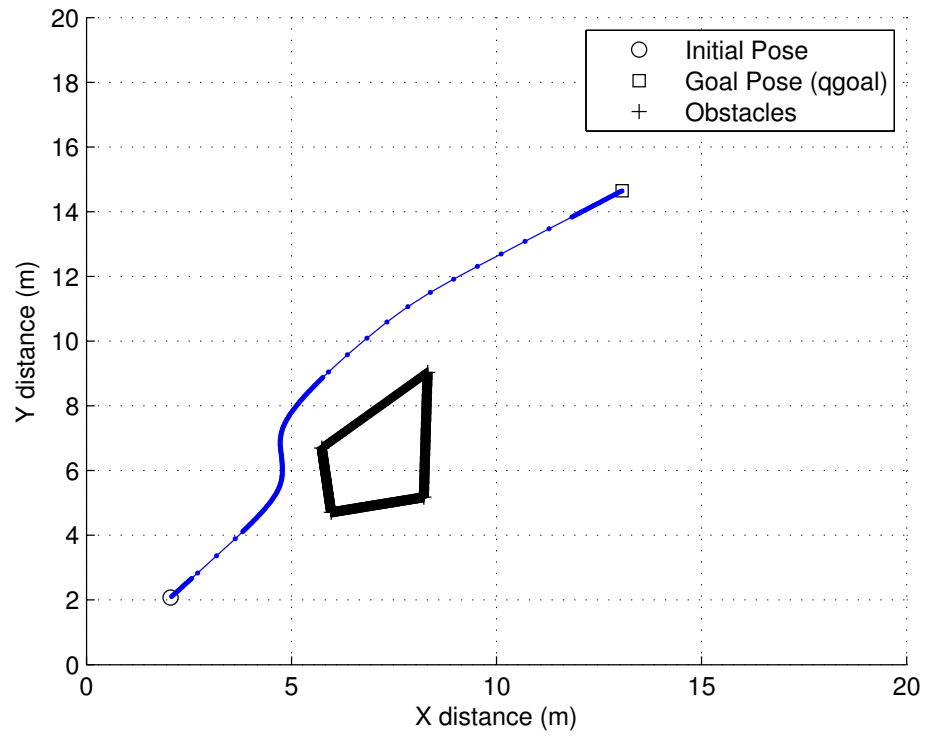
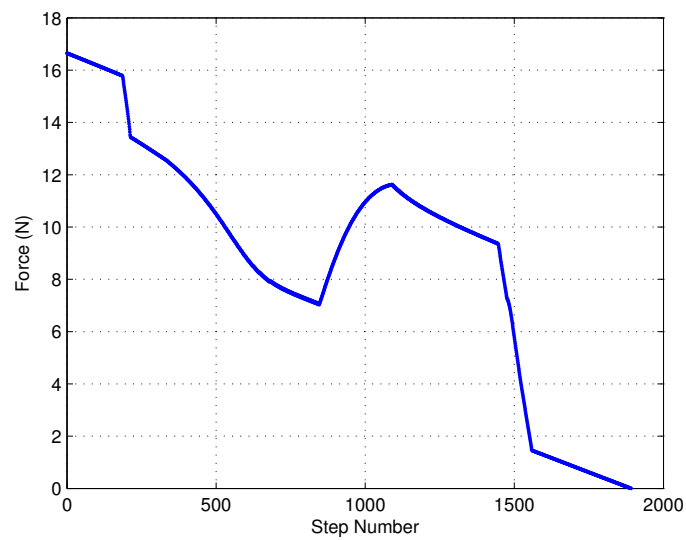


Figure 4.4: Modified PFPP Curved Path 2

Figure 4.5: Artificial Resulting Force (F_{tot}) for Modified PFPP Curved Path 2

shown in Fig. 4.6. A corresponding artificial force plot is shown in Fig. 4.7. For this case the repulsion force is higher due to the high value coefficient η . This stronger repulsive force is chosen to ensure the robot ‘particle’ does not enter the obstacle boundary of influence r_o . For a robot particle, such a large obstacle would result in numerous path planning data points that would translate to slow motion which is not desirable for a longer travel distance. Hence in the vicinity of the obstacle shown in Fig. 4.6 there is no decrease in distance travel as observed in previous simulations. The higher η also provides added safety factor in absence of boundary of influence effects. The points plotted for the path in Fig. 4.6 and the corresponding F_{tot} in Fig. 4.7 are every 10th point. This is done in order to reduce congestion and observe the trend more clearly.

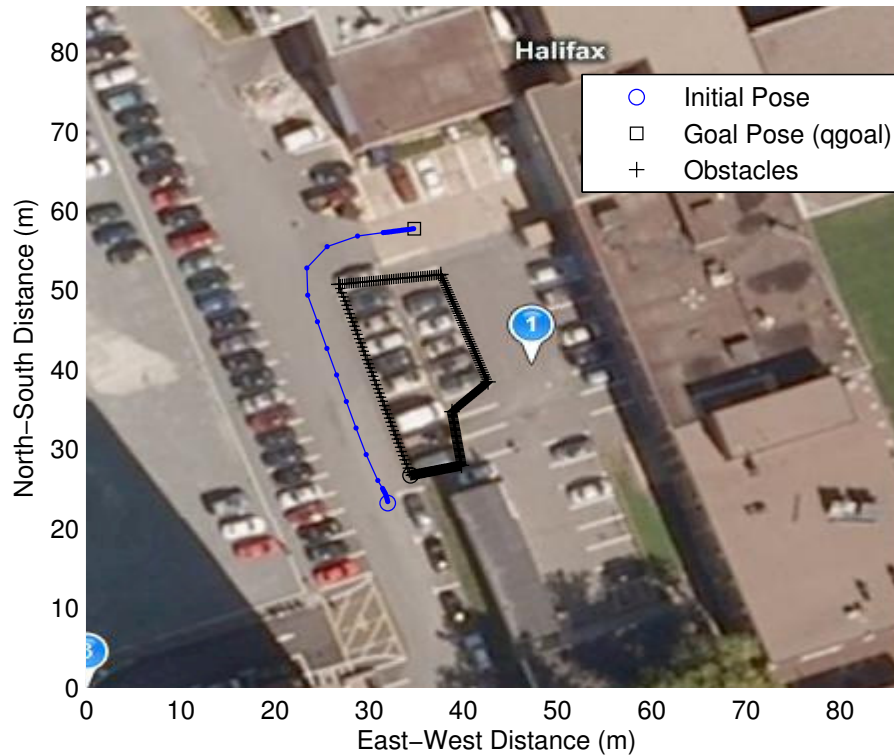


Figure 4.6: Modified PFPP Curved Path 3

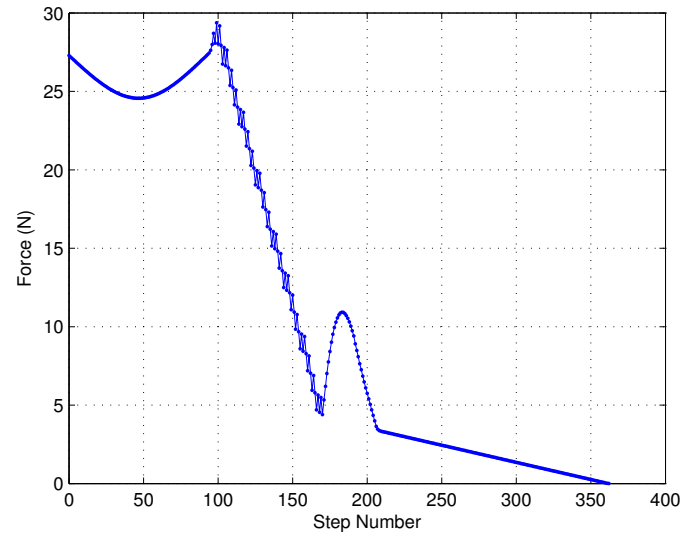


Figure 4.7: Artificial Resulting Force (F_{tot}) for Modified PFPP Curved Path 3



Figure 4.8: Modified PFPP Linear Path 4

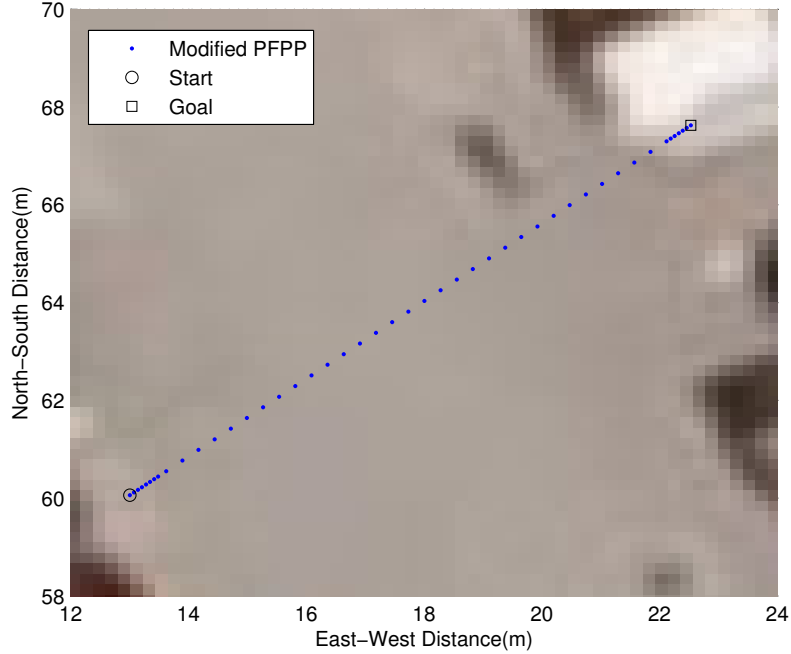


Figure 4.9: Modified PFPP Linear Path 4 (Zoomed Profile)

C. Experimental Test Case Simulations

Based on the previous path planning simulation results in part A and B of this Section two cases were simulated for eventual testing purposes. These test cases are short distance (1) linear trajectory and (2) curvilinear trajectory modified PFPP paths. Certain general parameters are set for consistency between the two test case simulations. These are step sizes, boundary of influence radii and gain coefficients.

The maximum δ_{max} and the minimum δ_{min} step sizes for both cases are equal to 0.35m and 0.0219m. Similarly for both cases, the boundary of influences values r_s , r_g , r_o are 0.5m, 0.5m, 1.5m respectively. These boundary influence radii r_i 's (start r_s and goal r_g) are chosen to allow the robot to have defined slow movements within a bound that also coincides with dGPS sensor accuracy concern area of less than 1m. As with the previous simulation in Fig. 4.6, η is chosen to be higher and ensure no obstacle boundary of interest r_o influences on the path planning. The modified PFPP gain coefficients for both test cases are $\xi = 1$ and $\eta = 300$. Linear trajectory path planning test case results are shown in Figures 4.8 and 4.9. The zoomed profile, shown in Fig. 4.8, is useful for close analysis and comparison with experiment tests.

Similarly for curvilinear trajectory path planning the results are shown in Figures 4.10 and 4.11.



Figure 4.10: Modified PFPP Curved Path 5

4.2 dGPS-based Motion Control NM Algorithm with EKF for Path Following

The cases considered for the motion control part of the path following algorithm correspond to the trajectories planned in Section 4.1. NM controller parameters such as maximum limits and gain values are constant for these simulations. The EKF process from Equation (3.6) to (3.16) is used for all NM with EKF simulations in Matlab. With the prospect of migrating to the robot platform for testing the NM with EKF is also simulated using MobileSim. The EKF constant parameters G_w , H_k , R_v , R_w are assigned as defined in Chapter 3 for these simulations.

The calculated control velocities v_n, w_n from Equation (3.5) are limited by the maximum absolute translational and angular velocities ($v_n = 700mm/s, w_n = 2.0rad/s$) which is set for the WMR safety. Starting controller gains $0 < k_v < 1, 0 < k_w < 1$ are chosen for the simulation based on Scaglia et. al values in [41]. As in [41], Fig. 4.12

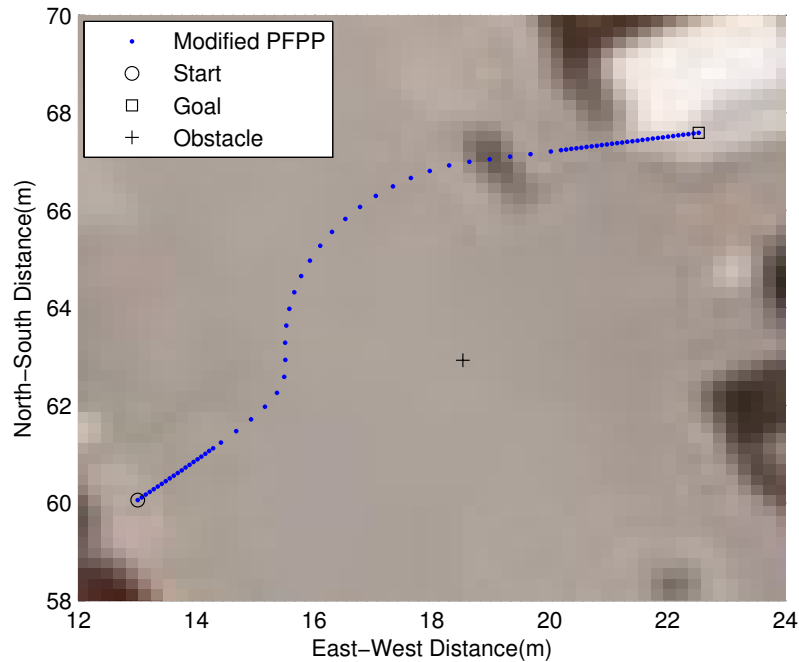


Figure 4.11: Modified PFPP Curved Path 5 (Zoomed Profile)

shows that $k_v, k_w = 0.2$ produces a good results when initial conditions $(\hat{x}_0, \hat{y}_0, \hat{\theta}_0)$ are equal to desired initial condition $(x_{des}, y_{des}, \theta_{des})$.

Similar to the Matlab simulations, MobileSim simulations can be used to test the program code prior to applying it to the WMR platform. Therefore prior to testing the NM controller with EKF shown in Equation (3.16) with dGPS measurement on the robot, the NM controller with EKF is simulated using MobileSim and simulated dead reckoning sensors. Fig. 4.13 shows the result obtained from the MobileSim straight line trajectory path following simulation using NM with online EKF. The initial conditions for the MobileSim simulations were $\hat{x}_0 = x_{ref}(0)$, $\hat{y}_0 = y_{ref}(0)$, $\hat{\theta}_0 \neq \theta_{ref}(0)$. The error observed in the results indicates that initial conditions affect the results when using this control approach.

4.2.1 Simulation Results

There are a total of five simulations run in this section corresponding to the five paths planned in the previous section. The simulations run are the following: (A) linear without obstacles and curvilinear with obstacles, (B) Curvilinear with map image

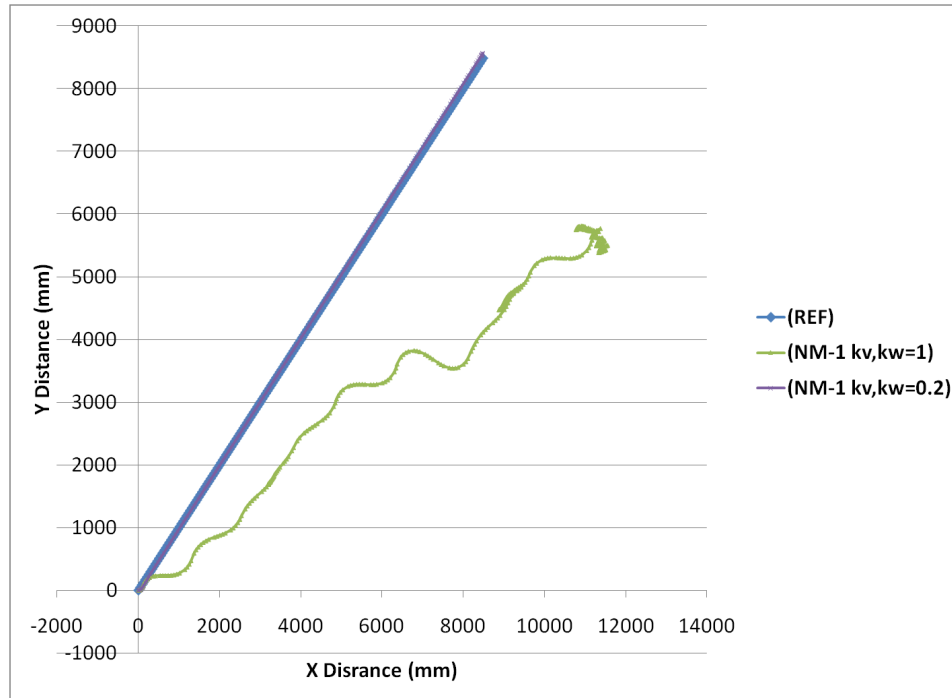


Figure 4.12: Comparing controller gain values of k_v , k_w using Mobilesim and Equation (3.5)

overlay and, (C) experimental linear and curvilinear test cases. These simulations are conducted in Matlab where the parameters are set for simulated dGPS measurement, sampling time and initial conditions.

A Matlab random noise function generator $randn(\cdot)$ is used to simulate a dGPS measurement with gaussian noise distribution bounded by the dGPS instrument accuracy ($< 0.60m$ at 95% confidence). Using the standard deviation σ calculated from the instrumentation manual, a bound value of 0.212 m on measurement noise in the north-south σ_y and east-west σ_x is obtained based on the dGPS instrument accuracy information. Propagating the error through atan function a value of $\sigma_\theta = 0.401$ rad is obtained. These values are then multiplied by the Matlab $randn(\cdot)$ function to produce a simulated noisy gaussian dGPS measurement. For each simulation the sampling time and initial condition criteria are consistent throughout. The sampling time (T_s), in the NM controller Equation (3.16), is equal to 1 second. This sampling time was selected as a base value because it would allow enough time for computation during testing. In the case of initial conditions criteria, $\hat{\mathbf{x}}(0)$ is equal to the simulated gps $\mathbf{x}_{gps}(0)$. The desired or reference pose parameters $(x_{des}, y_{des}, \theta_{des})$ are obtained

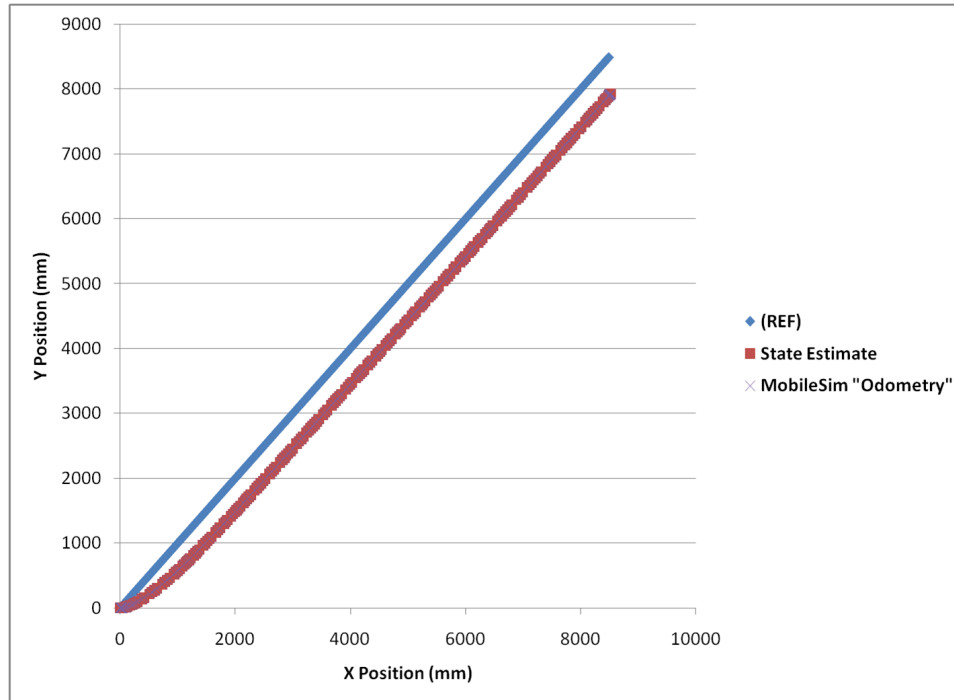


Figure 4.13: MobileSim using EKF with $\hat{\theta}(0) \neq \theta_{ref}(0)$

from Section 4.1. The initial covariance matrix P_0 is $P_0 = (\mathbf{x}_{des} - \hat{\mathbf{x}}(0))(\mathbf{x}_{des} - \hat{\mathbf{x}}(0))^{-1}$.

A. Linear and ‘Obstacle Filled’ Curved Path

The first simulation run is a straight line trajectory short distance simulation. The dGPS measurement is simulated using the Matlab random function. A reference vehicle is assumed to lie on the reference path which is planned in the previous Section as shown in Fig 4.2. The simulation Figures 4.14 and 4.15 show the result. The state \mathbf{x} and state estimate $\hat{\mathbf{x}}$ pose vectors in Fig. 4.14 are in agreement while the simulated Gaussian distributed dGPS measurement fluctuates. The control velocities in Fig. 4.15, also show agreement between state estimate and simulated robot state. The simulated robot state plot represents how a simulated robot would ideally respond with no sensor noise. For the straight line motion, linear velocity subplot in Fig. 4.15 indicates smaller translational velocities at the start and end of the simulation and larger velocities in between. The changing velocities corresponds to the changing step sizes as the robot ‘particle’ moves from start to goal positions shown in Fig. 4.14. As shown in Section 4.1, the boundary of interest r_o and r_g is set to dictate when the

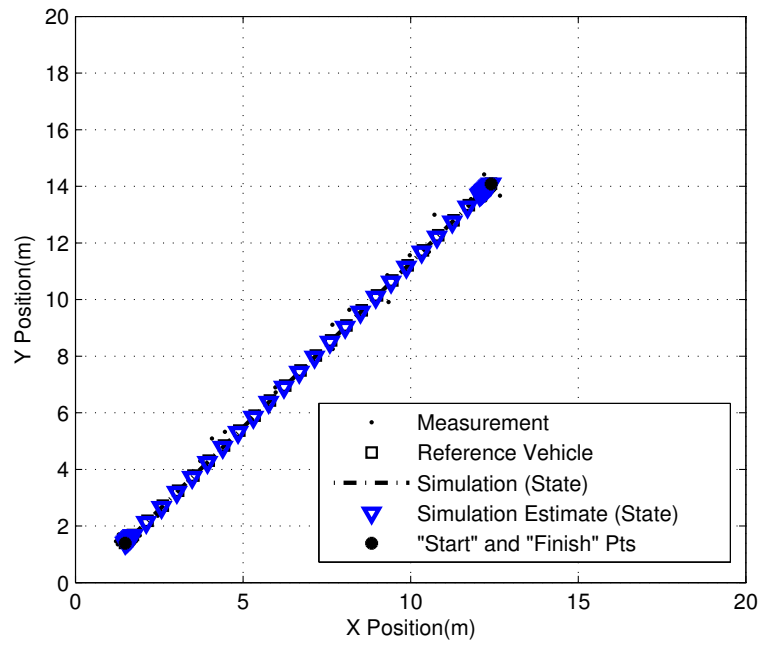
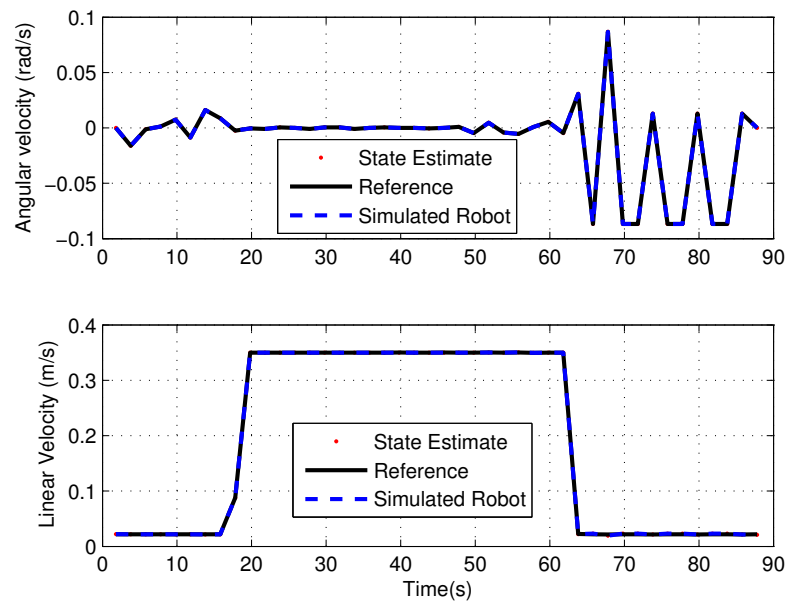


Figure 4.14: NM Path Following Control using EKF (Path 1)

Figure 4.15: Control Velocities v_c , w_c (Path 1)

change in velocities occurs.

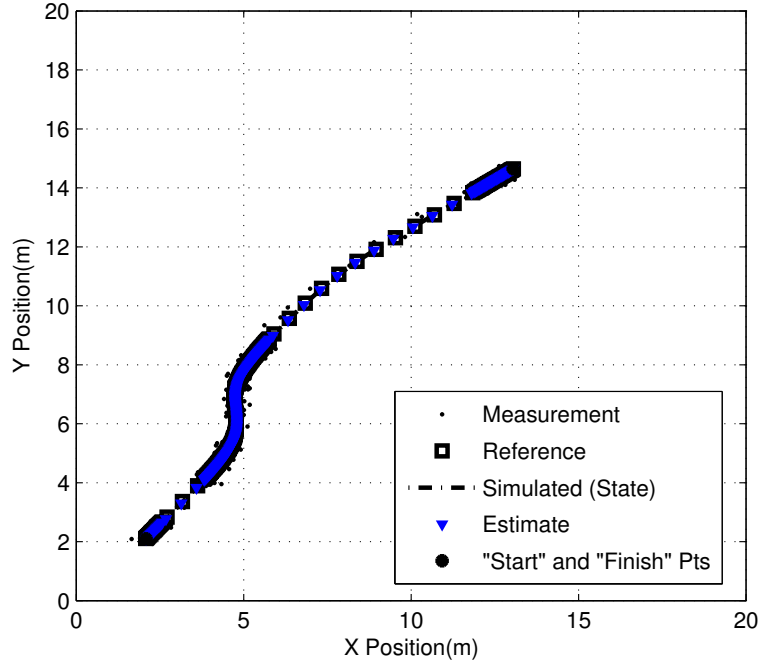


Figure 4.16: NM Path Following Control using EKF (Path 2)

The second simulation is a curved line trajectory simulation. The reference path for this simulation is planned using the results from the previous Section, shown in Fig 4.4. Similar to the first simulation, the results in Fig. 4.16, show agreement between the state estimate and the state. Likewise the simulated measurement are observed to fluctuate but within the accepted bound. From the control velocities plot in Fig. 4.17, the linear velocity subplot follows an increasing and decreasing trend which translates to changing distances between consecutive data points observed in Fig. 4.16. This trend is very much the aim of this proposed technique with decreased velocity in the vicinity of the object of interest (i.e. start, obstacle and goal).

B. Map Image Overlay

The third simulation run is a curved trajectory simulation with a travel distance greater than $20m$. By considering a longer travel distance we can determine if there are any error growth issues with the NM with EKF controller. The reference path is planned using the results from the previous map image overlay section, shown in

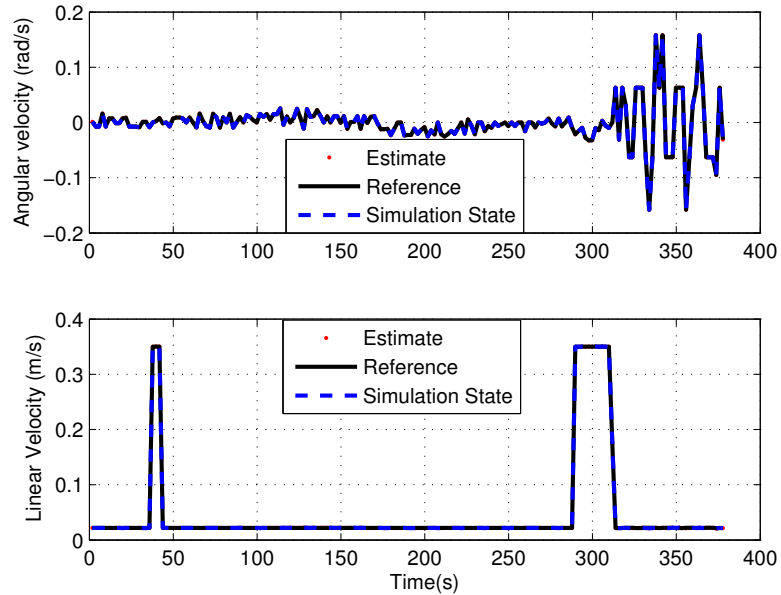


Figure 4.17: Control Velocities v_c, w_c (Path 2)

Fig 4.6. This path 3 simulation results are shown in Figures 4.18 and 4.19. For the position plot in Fig. 4.18, overall the estimate and simulated state are in agreement. There is a slight error between reference and state estimate as it nears the goal position. As for comparing control velocities between estimate and reference, Fig. 4.19 also agrees with only a slight difference in the control translational velocity.

C. Experimental Test Case Simulations

The fourth and fifth simulations are experimental linear and curvilinear test case simulations. In the simulation an EKF motion control is applied on the two experiment-ready path planned cases shown in zoomed profile Figures 4.9 and 4.11 from Section 4.1. As was the case with the previous three simulations in this section, the experimental test (1) linear and (2) curvilinear simulations include the simulated dGPS measurement, the reference vehicle on the reference trajectory, an ideal WMR state and an EKF estimate. The position and control velocities for the linear test case are shown in Figures 4.20 and 4.21. Both plots show fluctuations when using simulated dGPS Measurement for both localization and velocity control. For the curvilinear test case, the position and control velocity results are in Fig. 4.22 and Fig. 4.23

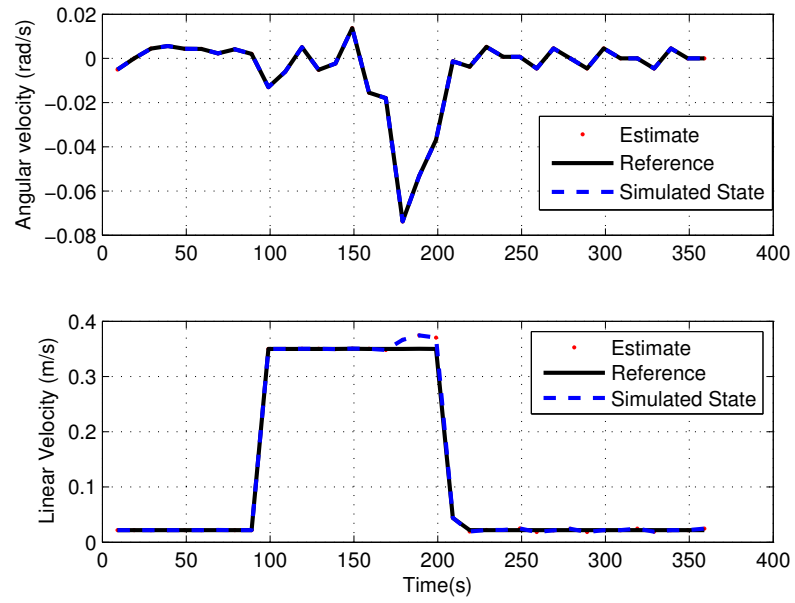


Figure 4.18: NM Path Following Control using EKF (Path 3)

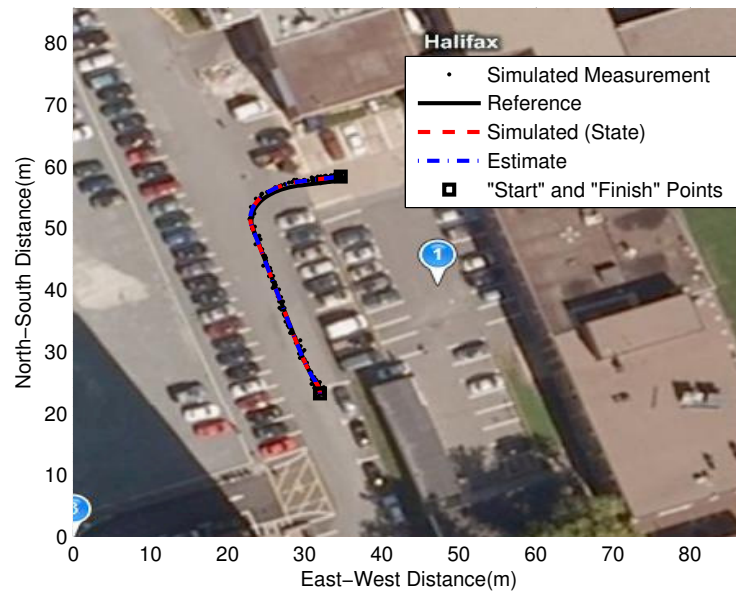


Figure 4.19: Control Velocities v_c , w_c (Path 3)

respectively. Unlike the previous simulations, the position simulation plots for both linear (Fig 4.20) and curvilinear (Fig. 4.22) show a slight error between final robot position and desired goal position. The curved path shown in Fig. 4.22 also has some error during motion along the curved section of path. Similarly the control velocities Figures 4.21 and 4.23 are more gradual than the reference vehicle velocities.

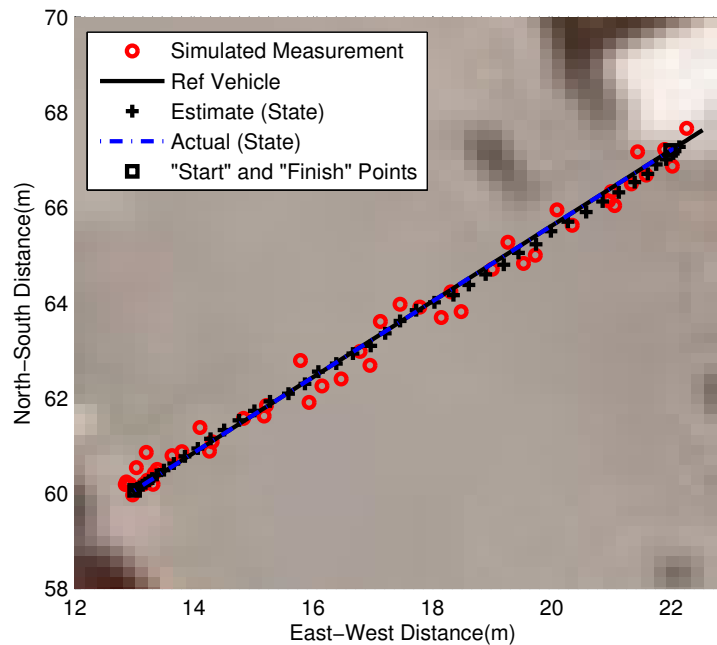


Figure 4.20: NM with Online EKF Control for Linear Path 4 Path Following

4.3 Summary

Modified (or geometric progression) PFPP and Motion control using NM with EKF are both simulated in this Chapter.

As was noted in Section 4.1 and observed in the simulation results for linear and curvilinear path planning, the modified (or geometric progression) PFPP component of the path following approach concentrates more points in the area of interest. For path planning the boundary of influences (r_s, r_g, r_o) and step sizes ($\delta_{max}, \delta_{min}$) parameters are adjustable. The boundary of influence values are set to be under 2 meters in all simulated cases. In the case of obstacle boundary variable r_o , the boundary

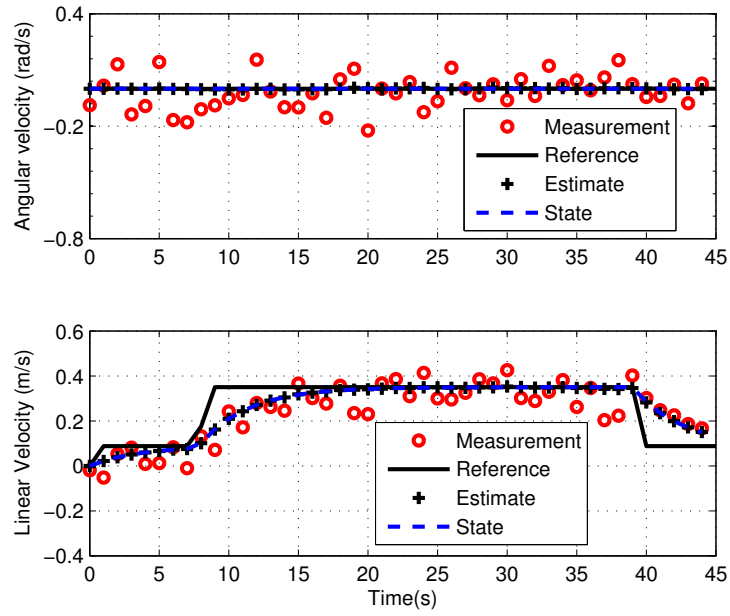


Figure 4.21: Control Velocities v_c, w_c (Path 4)

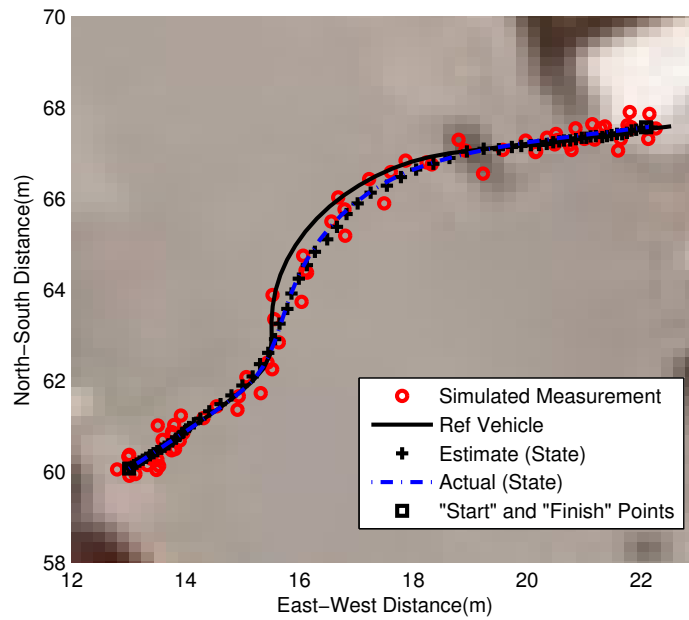


Figure 4.22: NM with Online EKF Control for Curved Path 5 Path Following

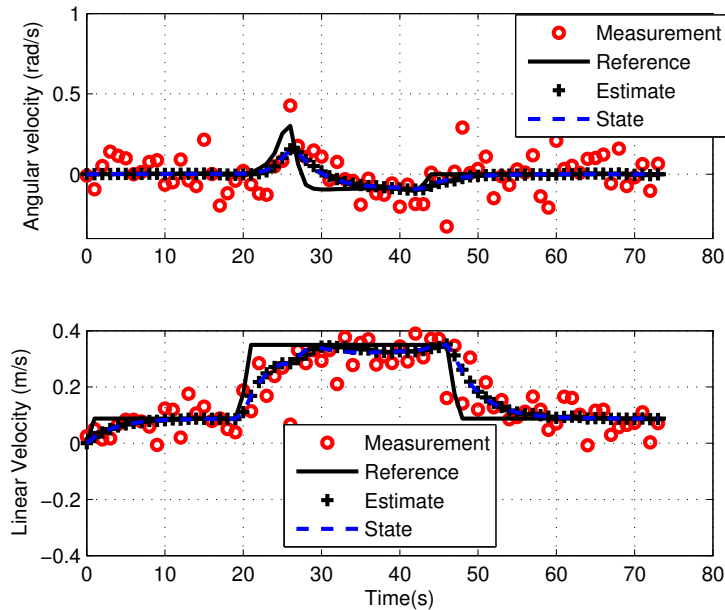


Figure 4.23: Control Velocities v_c , w_c (Path 5)

influence effect can be negated using a stronger repulsive force coefficient η as is observed in the Figures 4.6 and 4.10. For experimental testing purposes, the distances simulated were short and located in places that could be easily tested. A longer distance travel curvilinear trajectory, shown in Fig. 4.6, was also explored to determine if the NM controller with EKF still holds. For all experimental-ready test cases, the map image overlay provides a useful guide for path planning in the test environment where known stationary obstacles such as cars can be planned for and avoided.

As part of the path following approach, motion control using NM with EKF was simulated with Matlab. The simulations for the Linear and Curved Path with NM control and EKF in parts A, B, and C show good agreement between the estimated state and the simulated robot. There are some slight errors observed in all the parts A, B and C but overall the trend is good. For experimental test cases (part C), the state estimate control velocities (as shown in Figures 4.21 and 4.23) are much smoother than the equivalent simulated dGPS measurement velocities. This smoother trend tends to be better for the WMR control as it results in less fluctuating velocity commands for the WMR motors. In Section 4.2 the parameters such as step size δ_{max} and δ_{min} and sampling time T_s are set. These parameters dictate the translational

control velocities observed in all simulations. The result for the experimental test cases is $T_s = 1\text{s}$, $\delta_{max} = 0.35\text{m}$ and $\delta_{min} = 0.0219\text{m}$. Combined, these values ensure that the resulting translational control velocity v_n does not exceed the maximum set limits of $v_{max} = 700\text{mm/s}$.

In Chapter 5, we consider experimental test cases simulated in this Chapter and conduct some preliminary tests of the approach with the current parameters. The tests are conducted in the parking lot test environment using a research mobile platform. After the preliminary tests some parameters are fine-tuned and a second set of tests is conducted. Finally the Chapter 5 is concluded with performance comparison and analysis of the algorithm.

Chapter 5

Experimental Studies

In this Chapter the experimental set-up is first presented. The hardware and software section includes details on the mobile platform, its environment, and a procedure on how the WMR path following experiments are conducted. In Section 5.2 we conduct preliminary tests for the two experimental cases presented first in Section 4.1.1, then again in Section 4.2.1. The preliminary experimental results are used as a base to fine-tune the process covariance R_{Δ} parameter of the EKF. Simulations and experiments are conducted with the new parameter for two new experimental test cases in Section 5.3. The final Section in this Chapter is for analysis of the proposed approach simulation and experimental results.

5.1 Hardware and Software Setup

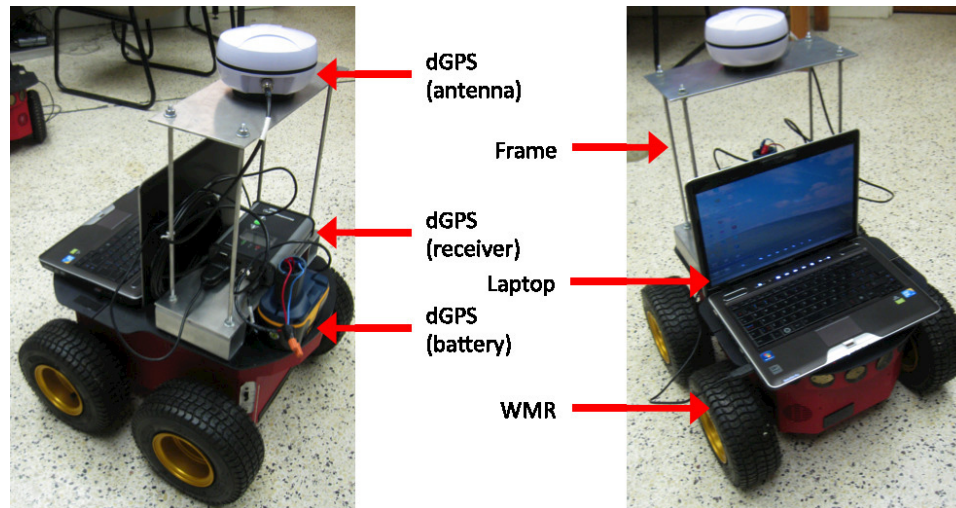


Figure 5.1: Hardware, Software and Testing Setup

The testing experiments are conducted using a Toshiba Laptop running Windows XP with Visual Studio C++ Express 2008 and Aria C++ Class Library. The Aria C++ library for the Pioneer WMR has extensive functionality for the robot and

any additional accessories such as GPS sensors. It is used to communicate with the WMR. The GPS Unit and the WMR are connected to the laptop via a serial to USB cable. The GPS unit used in these experiments is a R110 Crescent Hemisphere receiver with an antenna capable of obtaining a dGPS SBAS (WAAS) signals from the surrounding GPS and GEO satellites, as shown in Fig. 3.1. In order to set the GPS unit to receive the correct GPS string, a GPS software ‘PocketMaxPC’ is used to configure the receiver. PocketMaxPC also provides a quick and easy way to test dGPS status and ensure that a GPS string is being received through the correct serial/usb port. The closest WAAS ground reference station to the test site is located in Gander Newfoundland, Canada. As noted in [62], the reference station communicates with the Telesat (ANIK-1fR, W107.3 deg, PRN -138 (51)) GEO satellite in order to provide the user with differential WAAS GPS capabilities. The WMR is a Pioneer 3AT Outdoor research platform WMR developed by Adept MobileRobots. It is a research platform used for navigation in [44] and is similar to the Pioneer 2DX Indoor WMR used for path following in [41] by Scaglia et. al to test the NM control algorithm. The Pioneer 2DX is also used in [51] by Houshangi et. al. to test the UKF method for accurately determining the position of a WMR. Unlike the Pioneer 2DX, the Pioneer 3AT has four larger ‘outdoor-ready’ wheels and no onboard computer. A frame (Appendix A.1) is designed and built to house the GPS unit receiver and antenna. Both laptop and WMR are powered by rechargeable unit specific batteries. The GPS receiver is powered by a 18V drill battery. Fig. 5.1 shows the complete mobile platform used to test the dGPS based path following approach.

The Dalhousie Sexton Campus parking lot shown in Fig. 5.2 is the testing site for the experiment trials. The lot is paved and in fairly good condition with minimal cracks and slopes. In order to manoeuvre the WMR to a desired location for testing a Wiimote game controller was used for tele-operation. The Wiimote can communicate with the Laptop via bluetooth communication. A C++ program was created to combine the Wiimote [65] and Aria class libraries. This combination allows the user to then teleoperate the Pioneer 3AT WMR.

For testing the experimental cases visual markers, such as mini pylons, were used to mark the locations of the start, obstacle and goal positions. Fig. 5.4 shows a test sequence of the WMR Platform in motion as the robot moves from start to finish.



Figure 5.2: Testing Environment with Stationary Point Test Data Sets

Prior to proceeding with the path following algorithm the robot obtains its current or initial angle θ_{gps} within the test environment using a 1 meter forward and backward motion. Since the current dGPS can only provide information on (x_{gps}, y_{gps}) , using the previous position and current position the current heading information can then be calculated. After determining θ_{gps} , the robot then proceeds to follow a path using the proposed approach. For the EKF filter, the initial state estimate conditions are $\hat{x}_0 = x_{gps}$, $\hat{y}_0 = y_{gps}$, $\hat{\theta}_0 = \theta_{gps}$.

5.1.1 Procedure

The procedure for the testing trials follow these steps:

- First, using PocketMaxPC software ensure the receiver is receiving the correct

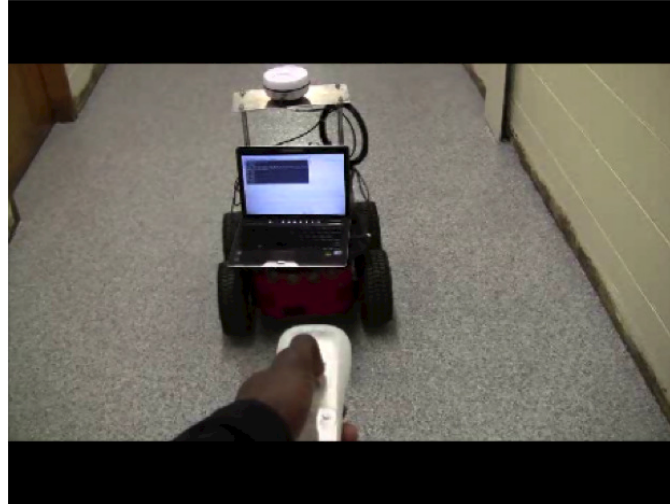


Figure 5.3: Wiimote Teleoperation



Figure 5.4: Testing Run Sequence Path 5

NMEA 0183 message strings. (GPGGA and GPRMC) and that there is SBAS differential lock.

- Then run a C++ program (gpsStationary.cpp) to acquire 60 data points for latitude and longitude of starting, obstacle and goal stationary. (Saved in a text file)
- Using the text file, run a Matlab program (LL to xy.m) to calculate the mean (\bar{x}_s, \bar{y}_s) and standard deviation (σ_x, σ_y) for start, obstacle and goal.
- Using the mean values for start, obstacle and goal positions, run a Matlab program (PF 031911a.m) to generate the modified PFPP path (Saved in a text file)
- With this generated path, run a Matlab program (NM 031911a.m) to simulate

the NM with online EKF path following

- Finally, using the generated planned path text file (.txt), run the C++ path following program (p2dxFollower06.cpp) on the Outdoor WMR platform until the whole text file is read.

As previously mentioned, the robot first obtains its bearing or heading using a short back and forth motion before it begins following the path. During autonomous motion the robot uses the generated text file as the reference. The WMR moves in a discrete manner and at each time-step ($Ts = 1s$) it calculate a new state estimate ($\hat{x}_n, \hat{y}_n, \hat{\theta}_n$). The robot number of steps are limited by the number of data points in the path text file.

5.2 Preliminary Experimental Results

For the preliminary experiment, the test cases considered are those simulated in part C of Chapter 4. These two previously described cases are linear and curvilinear short distance trajectories. The control velocities generated by the EKF estimate (as per proposed approach in Eqn (3.16)) are used to move the WMR autonomously during a test trial run. Prior to conducting these experiments a set of tests were run to determine how controller gains k_v and k_w influence the test. Figure 5.5 shows the results. Controller gain of $k_v = k_w = 0.2$ is chosen for all experiments based on this comparison.

5.2.1 Case 1: Rectilinear Motion

The straight trajectory experimental result is shown in Fig. 5.6. There are four data sets plotted in this figure. The first dataset shown in the plot legend is the dGPS measurement which is represented by a ‘circle’. This measurement data indicates the location the dGPS receiver registers as the WMR moves. The second line of interest is the ‘solid line’ which represents the reference trajectory or planned path for path following. This path is calculated from part C of the linear trajectory in the previous Chapter. The third dataset is a ‘dash-dot’ line and it represents the simulation ideal WMR response. Lastly, the final dataset, indicated by a ‘+’, represents the EKF

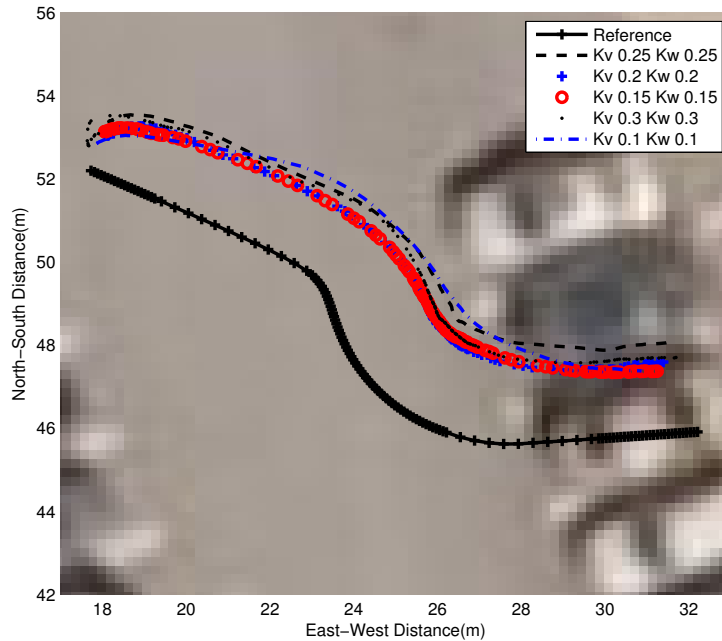


Figure 5.5: Comparing Controller gains experimentally (without EKF)

estimate of the location of the robot. This EKF position estimate is then used to generate control velocities to drive the WMR platform.

From Fig. 5.6, we note the difference between the ideal WMR simulated robot response versus the results observed in Fig. 4.20. This is primarily in part due to the difference in initial conditions between the plots. Even with the collected and averaged fixed position points using the procedure in Section 5.1.1, there is a shift in starting points for the experiment. This results in a position error between simulated robot and reference path. Fig. 5.6 shows that the measurement and state estimate also have an error associated with the reference path. They also deviate slightly from each other during initial motion but are in close agreement as the path following continues. The increased points associated with the proposed approach are observed more clearly at the start position in Fig. 5.6 rather than at the goal position. From the plot we can infer that the experiment path step max number is reached prior to reaching the goal area and slowing down.

This trend is also observed in the linear velocity subplot of Fig. 5.7. The estimate control velocity values in Fig. 5.7 are used to move the robot while the simulation and

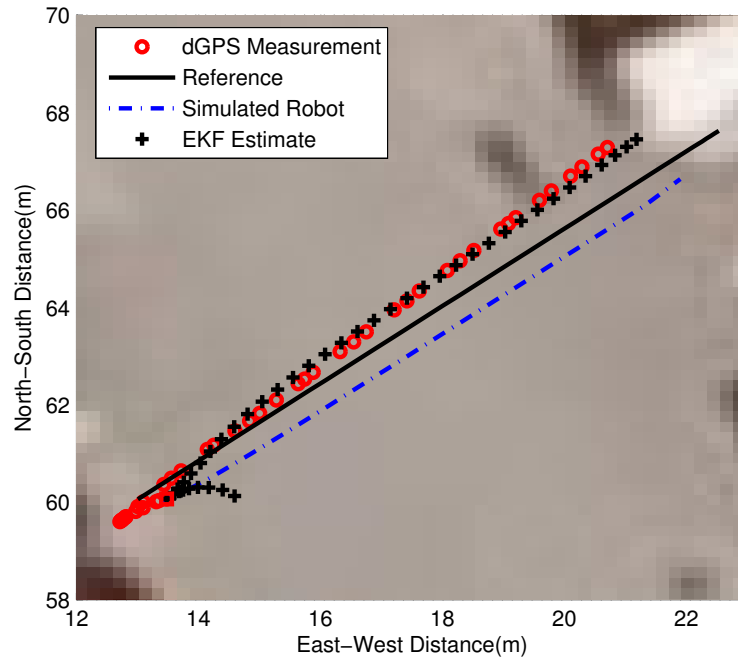


Figure 5.6: NM with Online EKF Control for Linear Path 4 Path Following (Zoomed Profile)

reference plots are for comparison purposes. Although the trend is similar between simulation and estimate, the estimate control velocities in Fig. 5.7 overshoot and undershoot the simulation robot response in a number of locations on the plot.

5.2.2 Case 2: Curvilinear Motion

For the curvilinear trajectory path, experimental results are shown in Fig. 5.8. Similar to the dataset legend in experimental case 1, the dGPS measurement is represented by a ‘circle’. Both the reference and simulated robot plots are represented by a ‘solid line’ and ‘dash-dot’ respectively. Lastly the EKF position estimate, which is also used to generate control velocities shown in Fig 5.9, is indicated by a ‘+’ symbol.

Since the simulated robot initial conditions are not equal to the reference initial conditions, the simulated robot results in an error at the start of motion. Fig. 5.8 shows that the measurement and state estimate also have errors with respect to the reference path and to each other. The error observed between state and dGPS measurement is more significant in a curved path than in the previous linear path

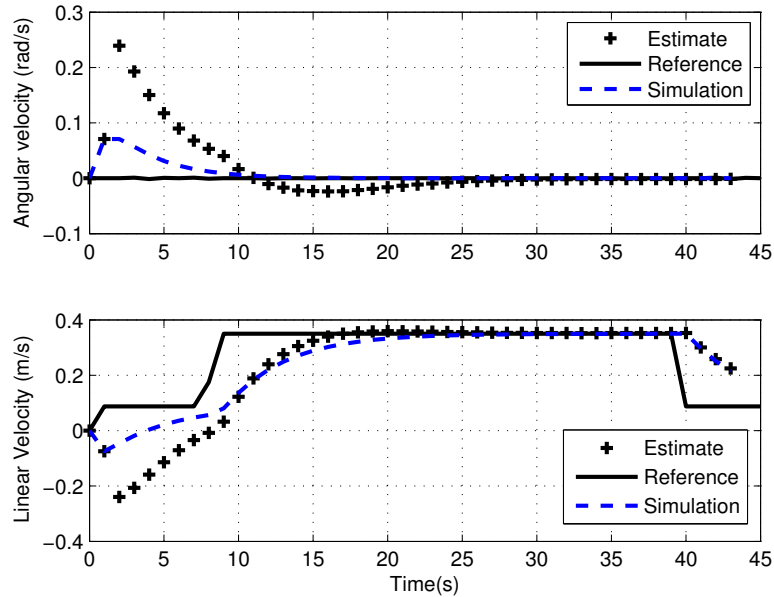


Figure 5.7: Control Velocities v_c , w_c (Path 4)

results. The increased points associated with the proposed approach are observed at the start and goal positions in Fig. 5.8.

In Fig. 5.9, the trend of the estimate calculated velocities is similar to the simulated robot velocities shown in ‘dashed’ lines. Errors between the simulation and estimate velocities are more pronounced at the start position with overshoots and undershoots but become minimal as the robot moves.

5.2.3 Summary

The results for a straight line path shown in Fig. 5.6 show some agreement of the position between state estimate and dGPS data. For the curved path in Fig. 5.8, the results are less agreeable with state estimate overestimating its location compared to where the dGPS sensor knows the robot position to be. This could be attributed to simplifications in the process model section of the EKF by setting process noise covariance $R_{\Delta} = 0$. The ideal kinematic model with EKF does not take into account experimental factors such as non-flat terrain test site and minimal skidding of Pioneer 3AT WMR.

Comparing the simulation and experiment measurements, the fluctuations of the

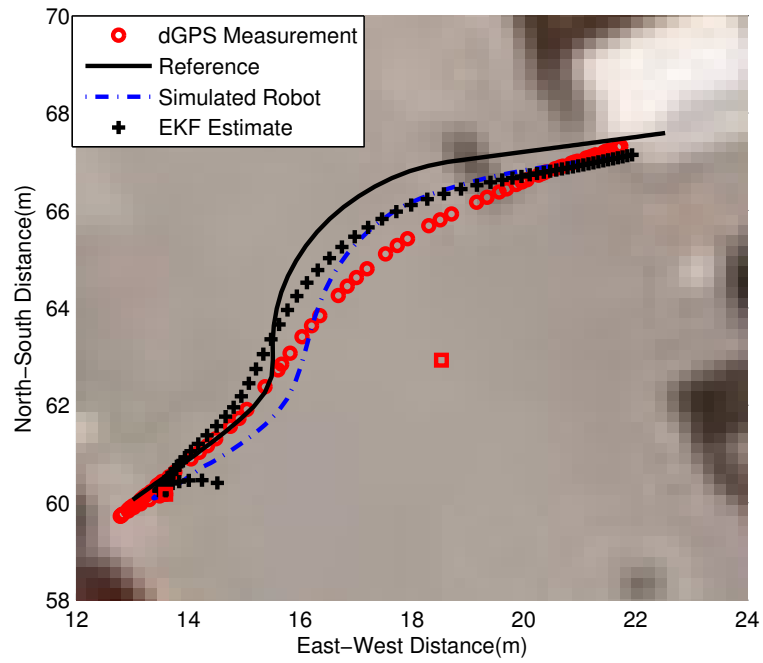


Figure 5.8: NM with Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)

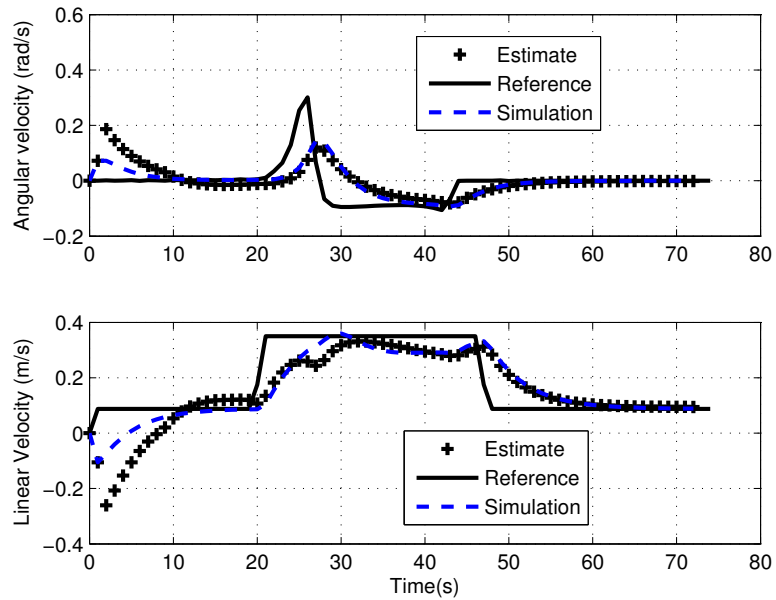


Figure 5.9: Control Velocities v_c , w_c (Path 5)

actual dGPS measurement are less aggressive and bound appears (visually) to be smaller in Fig. 5.6 and Fig. 5.8 than the simulated gaussian dGPS measurement in Fig. 4.20 and Fig. 4.22. This indicates that the dGPS measurement can be more accurate than the maximum instrument accuracy of less than $0.6m$ used in simulation. The fluctuations while small also do not clearly exhibit a gaussian trend which is used in simulation.

In order to compare with a norm, a set of experiments were conducted using direct dGPS measurement $(x_{gps}, y_{gps}, \theta_{gps})$ for linear and curvilinear trajectory. These experiments bypass the EKF component shown in Fig. 3.2 which corresponds to using Equation (3.5) instead of (3.16). The linear and curvilinear path results are plotted in Figures 5.10 and 5.11.

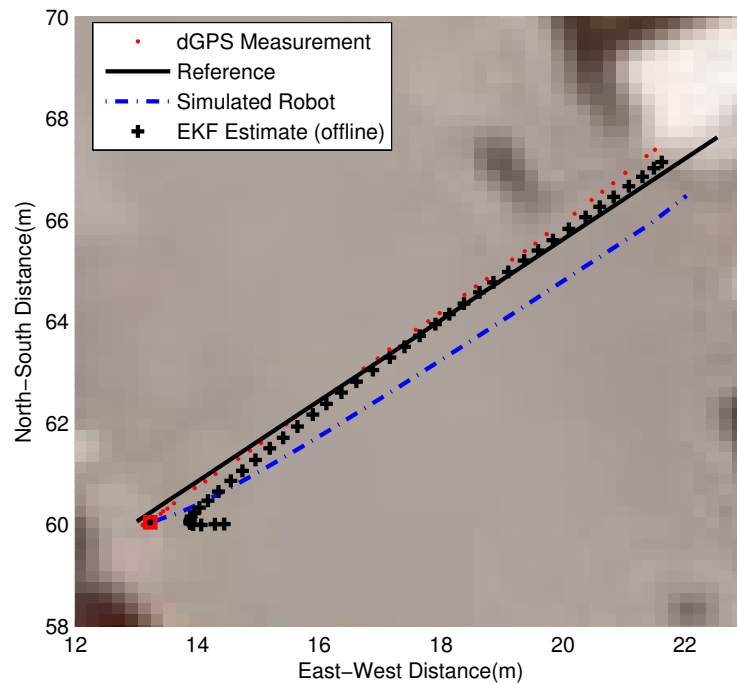


Figure 5.10: NM without Online EKF Control for Linear Path 4 Path Following (Zoomed Profile)

Table 5.1 summarizes the position error results for path 4 and 5 simulation (Figures 4.20 and 4.22) and preliminary experimental (Figures 5.6, 5.10, 5.8, and 5.11) work.

This table shows that the errors obtained are larger for experimental results versus

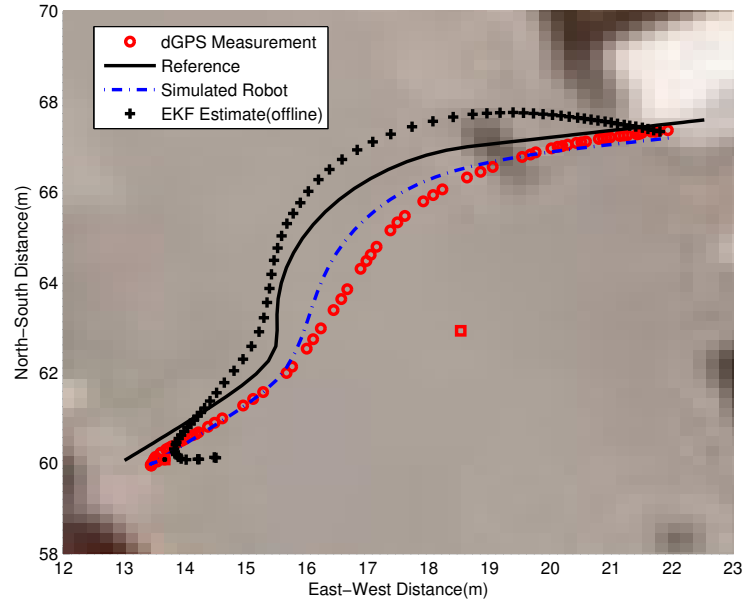


Figure 5.11: NM without Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)

Table 5.1: Preliminary Position Error Comparisons

	Simulated dGPS		Experiment dGPS	
Linear (Path 4)	with EKF Control	without EKF	with EKF Control	without EKF
Start Position Error (m)	0.07	0.07	0.47	0.22
Goal Position Error (m)	0.60	0.90	1.32	0.98
Max Error (m)	1.40	1.77	1.87	1.87
Curvilinear (Path 5)	with EKF Control	without EKF	with EKF Control	without EKF
Start Position Error (m)	0.35	0.35	1.47	0.48
Goal Position Error (m)	0.36	0.41	0.59	0.54
Max Error (m)	1.38	1.85	1.69	2.01

the simulation results. The maximum errors with EKF Control are less than those without EKF Control on average.

Secondly the result for curvilinear Fig. 5.8 show a large error between offline EKF and the dGPS measurement. This could be attributed to model simplification and EKF R_{Δ} parameter. In the section that follows, we use results shown in Fig. 5.8 and Fig. 5.11 to fine-tune the R_{Δ} term in our EKF and apply it to a new set of experiments.

5.3 Experimental Results

Since the preliminary experiments produced results that are not so agreeable for a curvilinear path. It is determined that an assumption of R_{Δ} does not fully represent the model of outdoor WMR. The value can be tuned offline using data collected from preliminary curvilinear tests with and without EKF control. After tuning the parameter value a new set of experiments linear and curvilinear is then conducted. These tests are conducted in the environment location of the previous preliminary experiments.

5.3.1 Tuning R_{Δ} parameter

To determine an acceptable value for R_{Δ} , a set of simulations are done, where the diagonal R_{Δ} matrix terms are increased. The EKF calculations for these simulations are offline but the data used is data that was collected from the preliminary curvilinear motion experiments with and without EKF. In these simulations $R_{\Delta x} = R_{\Delta y} = R_{\Delta \theta} = C_R$, where C_R is a constant that is manipulated. C_R values for these simulations are 1.0^{-10} , 1.0^{-8} , 1.0^{-7} and 1.0^{-6} . The results for varying C_R term in the EKF Estimate are plotted in Figures 5.12 and 5.13. The dGPS measurement, Simulated robot data and EKF estimate from the figure legend were previously determined in the preliminary experiments with $R_{\Delta} = 0$. The big dot plots represent the the offline EKF estimates obtained by changing C_R . From the results in Figures 5.12 and 5.13 we can see that as C_R is increased the EKF Estimate is much closer towards the dGPS measurement. Based on these simulations, an optimal value of $C_R = 1.0^{-7}$ is selected for the next set of experiments. The new experiments are conducted in the

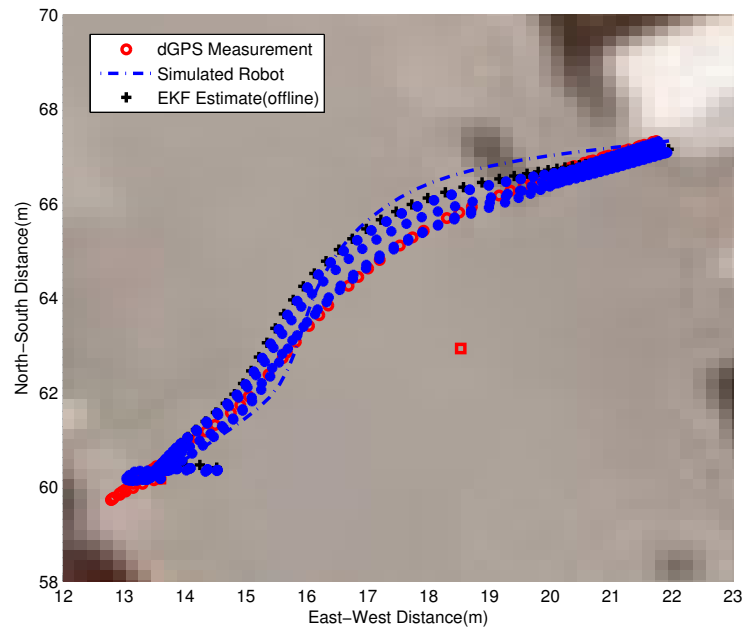


Figure 5.12: Tuning R_{Δ} NM with Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)

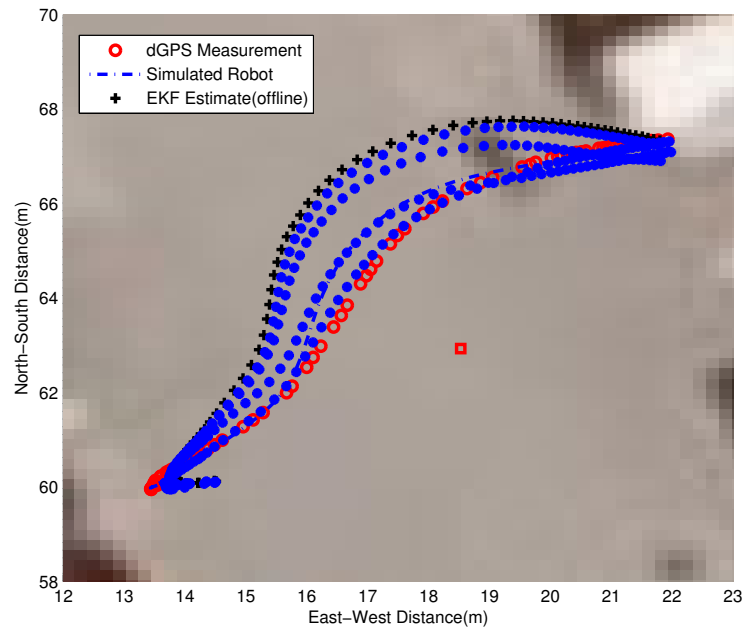


Figure 5.13: Tuning R_{Δ} NM without Online EKF Control for Curved Path 5 Path Following (Zoomed Profile)

same environment and with the start, obstacle and goal positions close to those used in the preliminary experimental tests.

5.3.2 Case 1: Rectilinear Motion

Prior to conducting a experimental test for linear trajectory, a simulation was conducted for the linear path with new value for EKF parameter R_{Δ} . The matlab code for the simulation can be found in Appendix B.

The result for the path with simulated dGPS measurement is shown in Fig. 5.14. Similar to the preliminary simulations, the simulated gaussian dGPS measurements

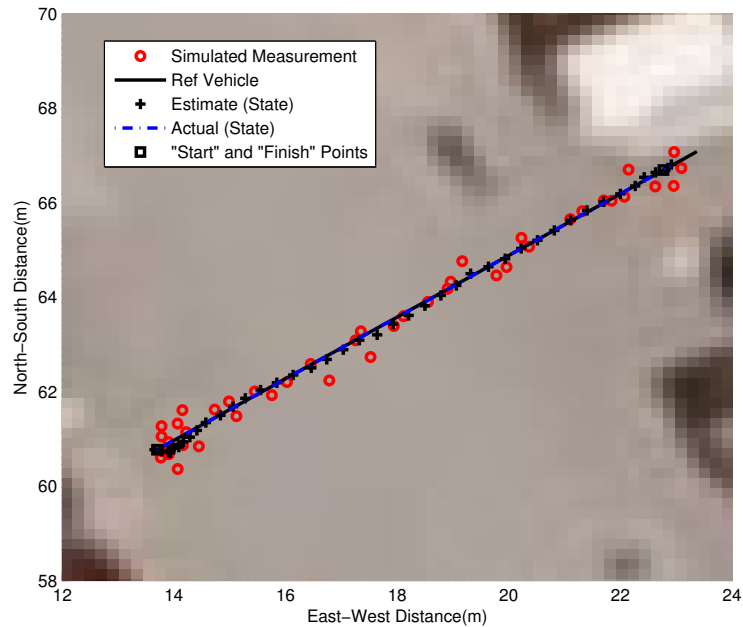


Figure 5.14: Linear Simulation Path 6 Path Following

has larger fluctuations compared to the EKF estimate. There is also an error between the actual final position and the desired final position which is also noted in the previous linear experimental test trajectory simulations from Section 4.2.1.

The linear experiment conducted to test the new improved EKF resulted in path 6 shown in Fig. 5.15. Path following of path 6 shows the robot EKF state and dGPS measure are initially far apart but eventually converge. The EKF Estimate and dGPS measurement have a smaller error than that observed in the preliminary

linear trajectory experiment. The associated velocity profile is shown in Fig. 5.16. The simulated robot and EKF estimate robot velocity profiles have some differences at the start but are more similar as the robot moves.

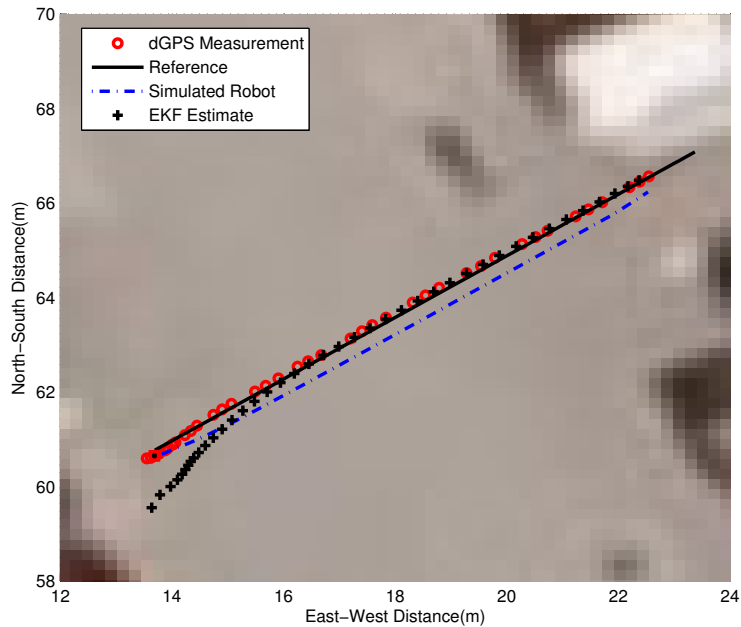


Figure 5.15: NM with Online EKF Control for Linear Path 6

5.3.3 Case 2: Curvilinear Motion

A simulation for a curvilinear path using simulated dGPS measurement was run with the new improved EKF. The result is shown in Fig. 5.17. It performs very similar to the previous preliminary experimental test curvilinear simulation.

Following the simulation, an experiment to test the improved EKF for curvilinear trajectory is run. This experiment resulted in Figures 5.18 and 5.19. From the path profiles in Fig. 5.18 the EKF estimate and dGPS measurement are much closer to each other throughout the robot motion cycle. This is unlike the preliminary case for curvilinear trajectories where they are much further apart. The velocity results for this path are shown in Fig. 5.19. There is agreement between simulated robot and state Estimate with slight deviations at the start and midway.

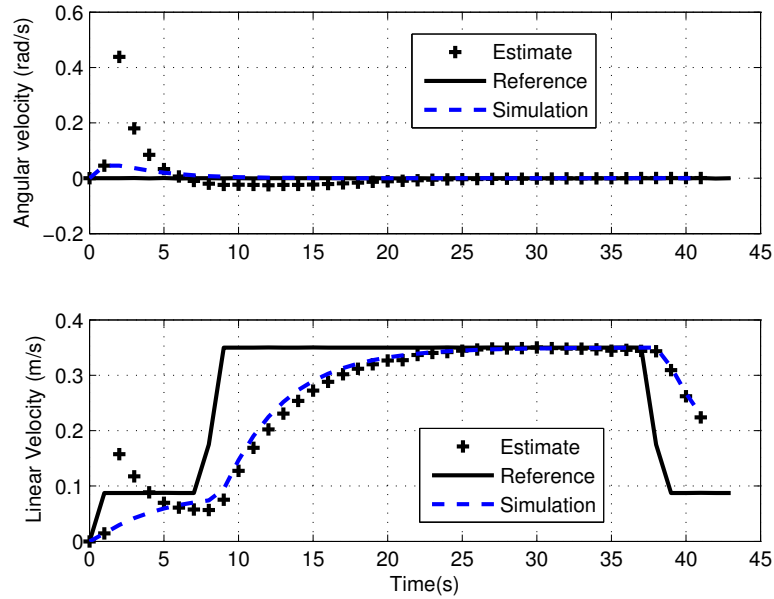


Figure 5.16: Control Velocities v_c, w_c (Path 6)

5.3.4 Summary

The improvement in EKF for online EKF control results in more closely connected EKF estimate and dGPS measurement. This in turn results in a better control of the WMR as it moves. The linear trajectory case shows close path following with the WMR dGPS measurement and EKF estimate points following on the reference path line. The velocity profiles for the linear case also agrees very closely to the simulation robot. One concern for the linear trajectory case is that the vehicle does not reach its final position in the allotted time steps. This can easily be adjusted by inserting a condition in the program such that the robot continues to move until an ‘arrival at destination’ condition is achieved. Another concern is that the simulated robot performs slightly different resulting in an error between simulated robot and EKF estimate. This can be attributed to simplified model simulation response.

For the curved trajectory case there is an improvement in error between dGPS measurement and EKF estimate when compared to the preliminary test curvilinear trajectory. The curvilinear trajectory velocity is smooth and follows the simulation results closely but has slightly less agreement in terms of linear velocity. There is also an error between simulated robot and EKF estimate during the curved motion.

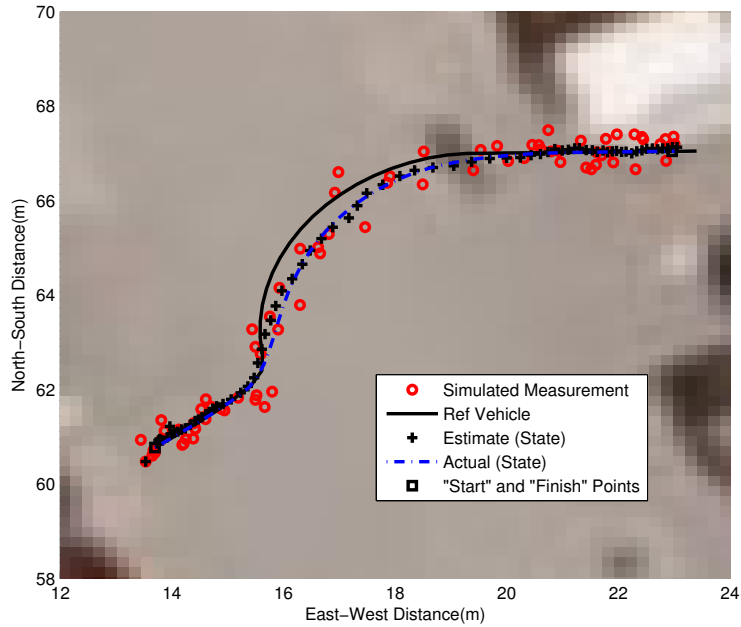


Figure 5.17: Curvilinear Simulation Path 7 Path Following

This error is likely due to either environmental factors such as uneven terrain or due to process model simplifications that do not accounting for four wheel skid steering WMR.

5.4 Data Analysis and Comparisons

Using the set-up with tuned R_{Δ} a new set of experiments that just use direct dGPS measurements (without EKF) for control were run. These test runs served as a norm that our approach (with EKF) was compared against. Criteria such as how control velocity compare to measured encoder velocity were also investigated for the two experimental cases with EKF control.

5.4.1 Case 1: Rectilinear Motion dGPS without EKF

For linear path following, an experiment is conducted using direct dGPS localization. The results of the experiment are shown in Fig. 5.20. From the results we note that there is an error between the dGPS measurement and both the simulation robot path and the reference path. This error is smaller than that observed in the previous case

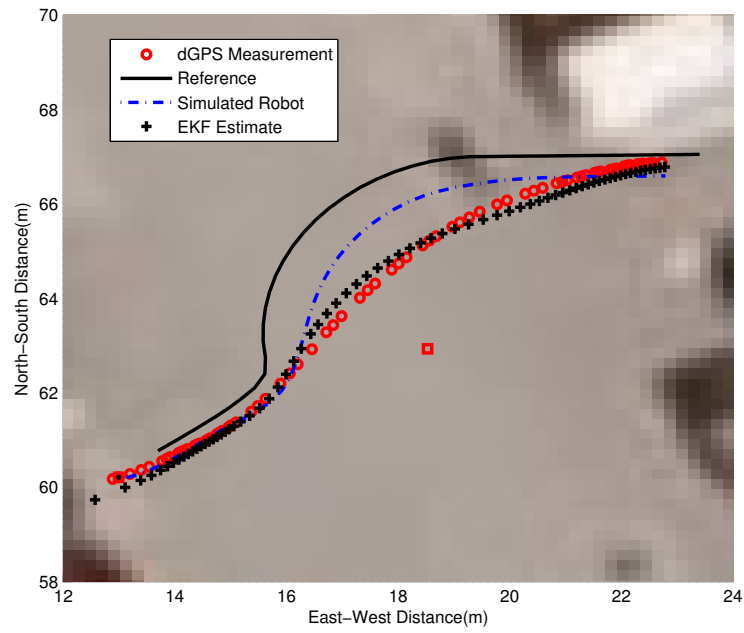
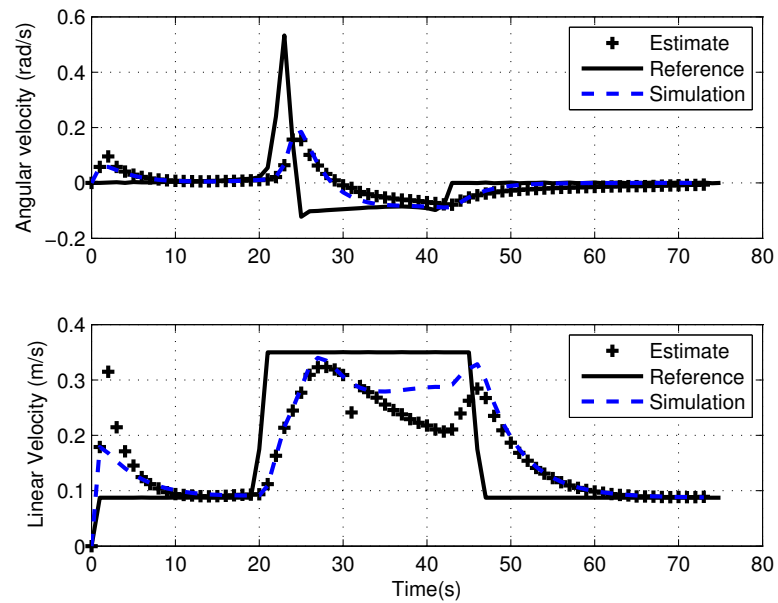


Figure 5.18: NM with Online EKF Control for Curved Path 7

Figure 5.19: Control Velocities v_c , w_c (Path 7)

with EKF control. The error with dGPS only without EKF control is smaller than the error observed in the equivalent EKF result.

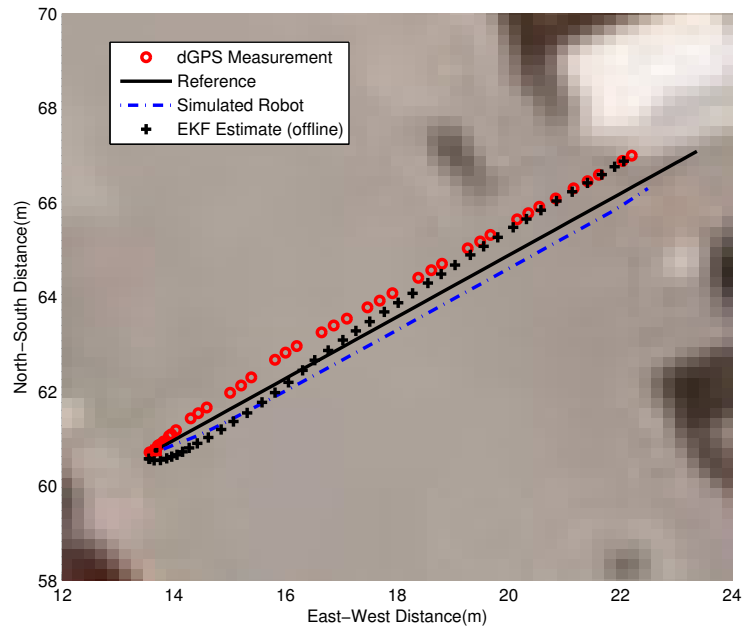


Figure 5.20: NM without Online EKF control for Linear Path 6

The resulting control velocities for linear Path 7 using dGPS measurements for control is shown in Fig. 5.21 . For the angular velocity, the dGPS measurement trend is quite similar in profile to that observed in the case with EKF estimate. There is an initial peak followed by a gradual decrease of angular velocity to zero. In the case of the linear velocity, the dGPS measurement calculated term tends to fluctuate more significantly than the linear velocity case than was observed with EKF in Fig. 5.16. With the EKF the linear velocity is much smoother.

5.4.2 Case 2: Curvilinear Motion dGPS without EKF

A curved path with dGPS control is also tested. The path following and velocity control results are shown in Fig. 5.22 and 5.23 respectively. The dGPS measurement path following in Fig. 5.22 is initially inline with the reference path but slowly veers towards the simulated robot path before veering off that path and concluding to the left side of the reference path.

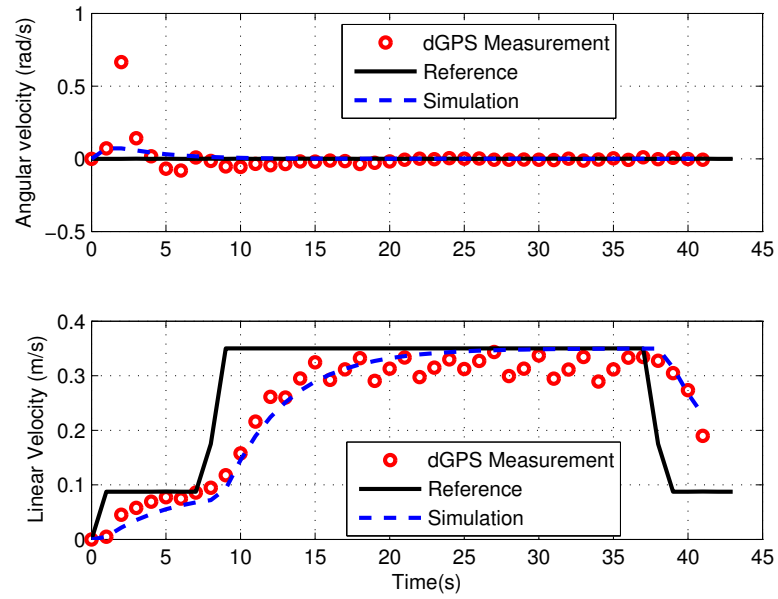


Figure 5.21: Control Velocities v_c , w_c (Path 6)

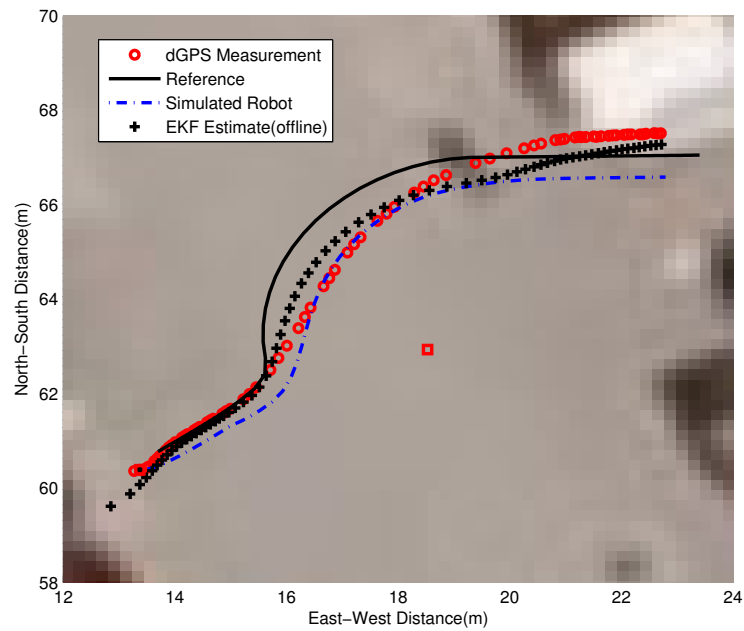


Figure 5.22: NM without Online EKF Control for Curvilinear Path 7

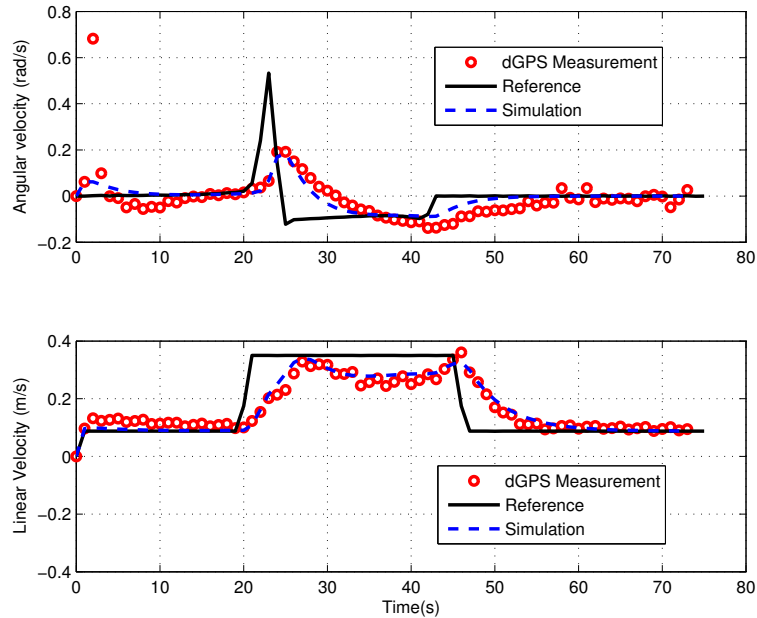


Figure 5.23: Control Velocities v_c, w_c (Path 7)

The angular and linear velocity results in Fig. 5.23 show slight dGPS measurement fluctuates when compared to the simulation. This is unlike the smooth trend found in EKF equivalent in Fig. 5.19. This reiterates that the EKF is useful in smoothing out the velocity profiles.

5.4.3 Comparisons of dGPS localization with active EKF control

Another criteria that was investigated was how well the control velocity signals represent the actual felt velocities. Investigating this was done by collecting data of the velocity values after each sample time T_s during the experiment for Paths 6 and 7 with or without active EKF control.

Figures 5.24 and 5.25 show the linear and angular velocities of path 6 with EKF respectively. The ‘solid’ line for both figures is the control signal while the ‘dash-dot’ line is the measured velocity from the robot. It is obtained using Aria C++ functions *getVel()* and *getRotVel()*. From tFigures 5.24 and 5.25 we note that although the control velocity represents the velocity we would expect the robot to experience during motion, in some cases they are instances where the control and measured velocities

don't match up for linear trajectory motion (Path 6).

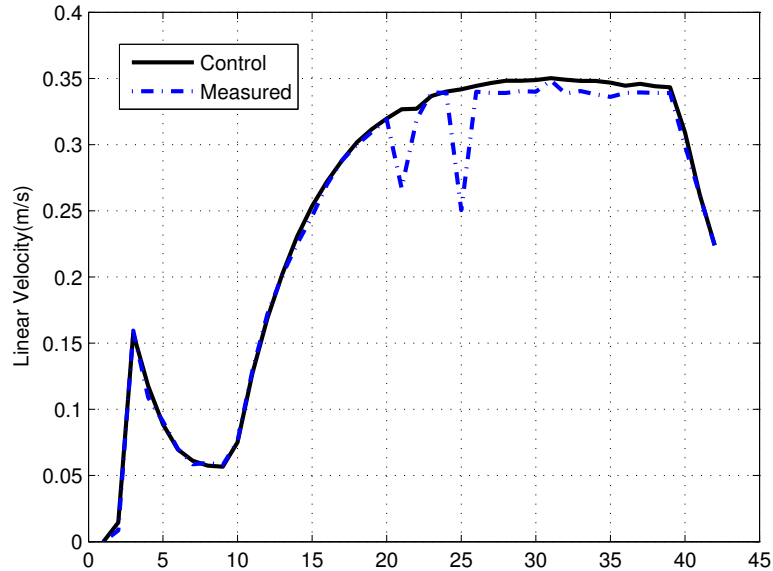
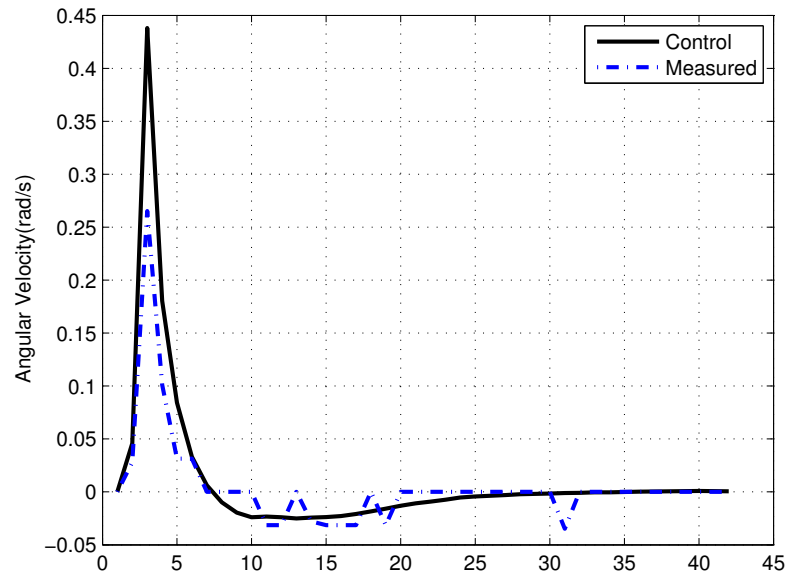
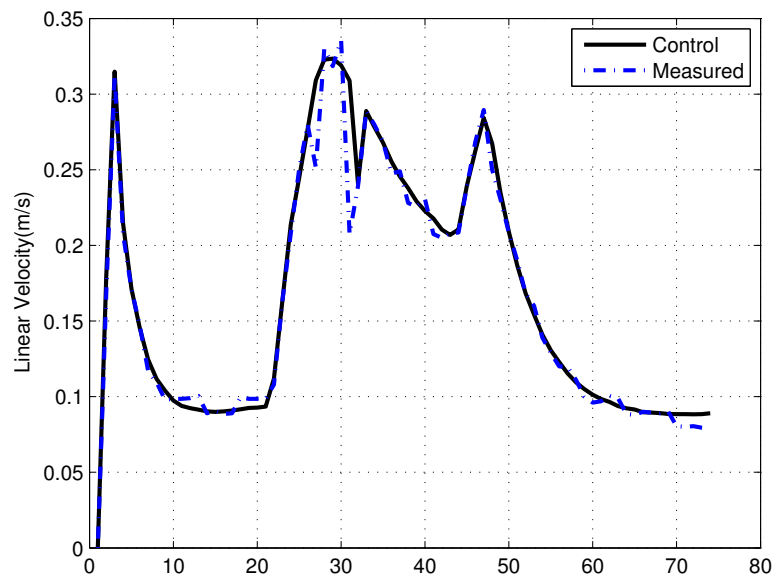


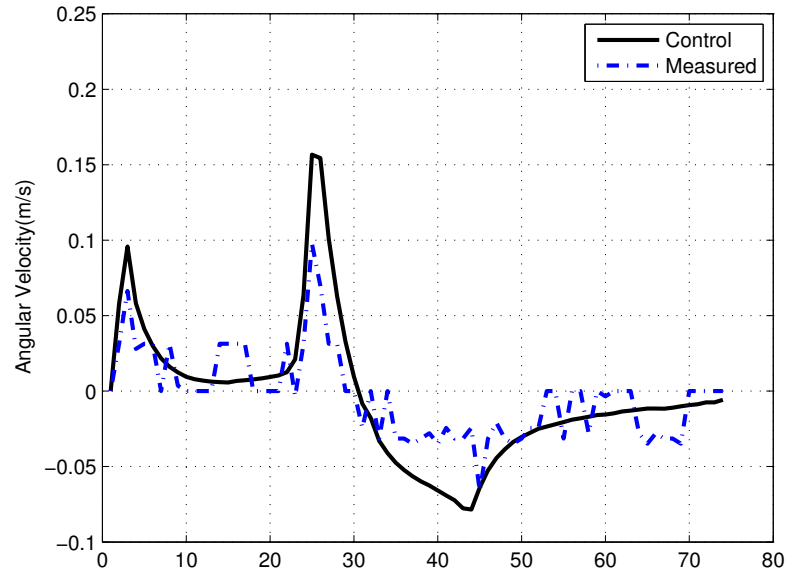
Figure 5.24: Velocity v_c (Path 6)

In the case of curvilinear trajectory this trait is also observed as is shown in Figures 5.26 and 5.27. Similar to the previous figures, the ‘solid’ line represents the control signal while the ‘dash-dot’ line is the actual velocity measurement from the robot. The differences are more significant for the angular velocities shown in Fig. 5.27.

5.5 Summary

Overall the experiments show that the dGPS data fluctuates less than predicted and in a non-gaussian manner. The non-gaussian nature of data fluctuations was expected as previous literature suggests this trend. Gaussian noise was used in simulation as the filter used was a EKF that assumes a gaussian distribution. For non-gaussian assumption a UKF would be ideal and can likely be explored in future work. The minimal fluctuations with dGPS data during path following suggests a better accuracy than the maximum instrument accuracy rating of less than $0.6m$. The preliminary results show that the ideal kinematic model requires noise process noise (R_{Δ}) consideration for completeness of the EKF aspect of the proposed technique. After tuning the (R_{Δ}) parameter, the path following results were much more representative of the

Figure 5.25: Velocity w_c (Path 6)Figure 5.26: Velocity v_c (Path 7)

Figure 5.27: Velocity w_c (Path 7)

expected outcome with regards to EKF control. When comparing EKF with dGPS direct control the results for velocity indicate the EKF control is smoother. The experiments are very dependent on initial conditions which fluctuate based on dGPS data. Table 5.2 summarizes the position error results for path 5 and 6 simulation

Table 5.2: Position Error Comparisons

	Simulated dGPS		Experiment dGPS	
Linear (Path 6)	with EKF Control	without EKF	with EKF Control	without EKF
Start Position Error (m)	0.40	0.40	1.27	0.02
Goal Position Error (m)	0.62	0.50	1.14	1.11
Max Error (m)	1.42	1.83	1.78	1.81
Curvilinear (Path 7)	with EKF Control	without EKF	with EKF Control	without EKF
Start Position Error (m)	0.03	0.35	1.62	0.68
Goal Position Error (m)	0.34	0.19	0.53	0.67
Max Error (m)	1.38	1.84	2.06	1.87

(Figures 5.14 and 5.17) and experimental (Figures 5.15, 5.20, 5.18 and 5.22) work. This table shows that the errors obtained are larger for experimental results versus

the simulation results which was previously observed in the preliminary results. The maximum errors without EKF is similar for simulated and experimental results. The error for experimental results are also shown in Figures 5.28 and 5.29.

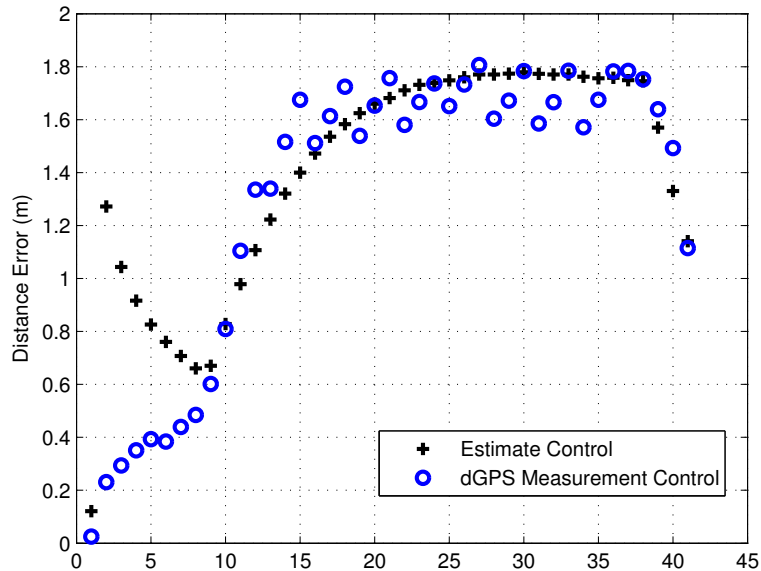


Figure 5.28: Experiment Position Error (Path 6)

For the path following the results for linear trajectory were better with EKF. As for curvilinear trajectory the results were comparable. The EKF control for curvilinear trajectory Path 7 shown in Fig. 5.18 does not perform as well as the simulated robot but this is likely due to process noise not modelled in the simulated robot. Process noise can either be due the terrain, or due to the skid steering robot. Similarly the difference between control and actual velocity, as shown in Fig. 5.26 and Fig. 5.27, play a part in increasing the error.

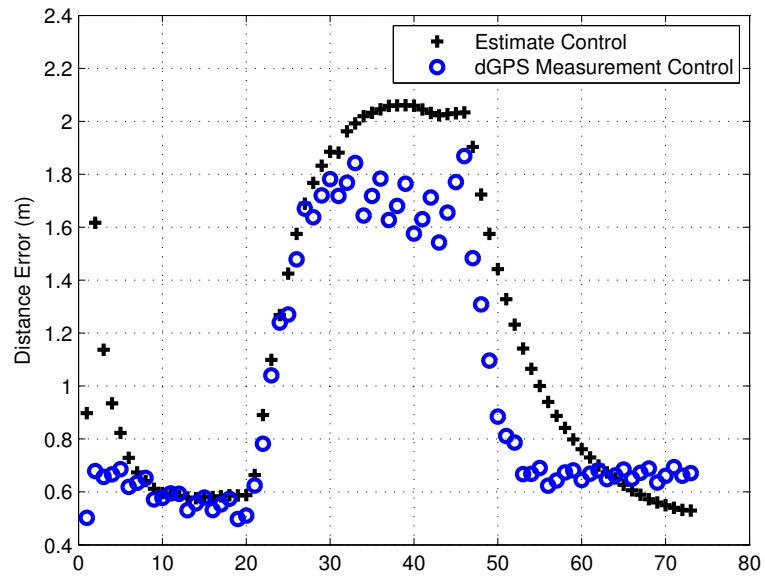


Figure 5.29: Experiment Position Error (Path 7)

Chapter 6

Conclusion

The application of a dGPS-based Path Following Approach in the Matlab simulation environment produced favourable results. The experimental results varied and were very depended on the initial conditions of the WMR, which varied quite frequently based on the dGPS measurement. Based on the initial conditions, there tends to be a disagreement and slight error in path following between reference and actual estimate.

When the algorithm was applied to the experimental platform with tuned process noise covariance, the results for linear path following, based on the proposed approach, were good. Both the dGPS measurement and EKF estimate fall on the reference path. The EKF online control provides smooth velocity profile unlike the fluctuating experimental dGPS only velocity profile.

The curvilinear trajectory experimental results, using the proposed approach, are much closer to the WMR simulated robot case at the start and goal positions. During the curved shape the simulated robot(or process model) and EKF are in disagreement which could be due to using a simplified process model. Another factor that influences the result is the differences between control signal and measured velocities.

6.1 Future Work

6.1.1 Model and Simulation

Some future work proposed to improve and obtain better results with the current EKF control set-up, would be to perform a system identification to determine an improved kinematic model that is more representative of the WMR skid steering and environmental factors. Some of these terms will likely be represented by process model noise. As the noise associated with the dGPS is non-gaussian using another filter technique such as UKF may provide more representative simulations. Since data

has already been collected from the experiments, UKF can be applied offline using the data to test. Computation time can then be investigated to see if applying online UKF control to the proposed approach is beneficial. Comparison between UKF and EKF performance can then be made to assess benefits.

Once an acceptable model has been obtained, another area worth improving will be to improve the controller as it is very dependent on initial conditions. Given the initial condition dGPS fluctuations and initial orientation fluctuation, a more robust controller able to handle these variabilities is necessary.

6.1.2 Experiment

Testing the approach in a degrading dGPS signal location should be conducted especially if the degradation is in the location of start, obstacle or goal, such as in close proximity to a building or under a canopy. During testing the robot was stopped at every time step to calculate the new online EKF value. This can produce a jerky motion, therefore using a shorter stop time or eliminating the stop time completely would improve this.

The experimental procedure is very time consuming to initially run. So improving it by incorporating a Graphical User Interface (GUI) to easily perform all the tasks would be very beneficial for testing. The C++ program can also be improved with efficient programming techniques for faster computation.

6.1.3 Expansion

This proposed technique uses an offline obstacle avoidance scheme. Expanding to include online avoidance recognition based on sensors such as sonar may aid in cases where the WMR gets too close to the offline obstacle. The use of online obstacle avoidance will also aid in identifying unknown obstacles and avoiding them.

After considering online obstacle avoidance, multiple vehicle or different robotic platforms can be explored for future use in a more generalized case. Figure 6.1 shows the other robotic platforms within the research group that may be considered.



Figure 6.1: ACM Lab Research Robotic Vehicles

Appendix A

Engineering Drawings

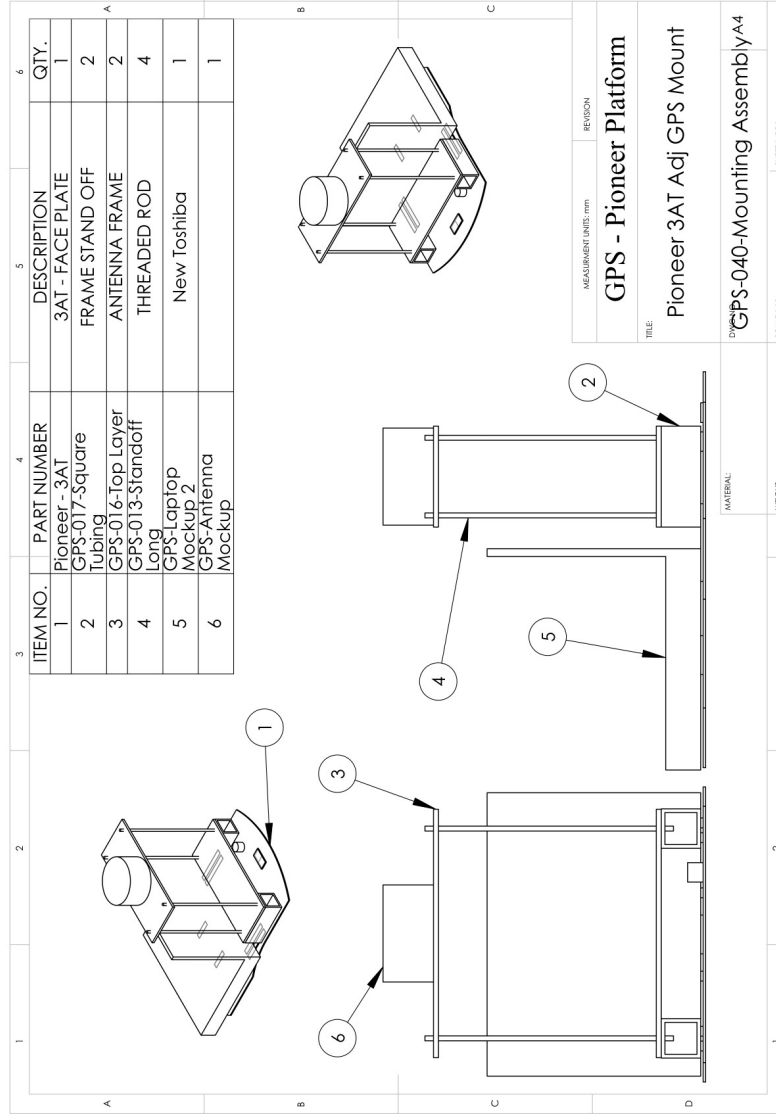


Figure A.1: Mounting Platform Assembly

Appendix B

Matlab Code

The following is the matlab code used for simulating the proposed technique for both linear and curvilinear trajectories. It is configured for path 6 and 7 simulations.

```
clear;
clc;
close all;
n=200; %number of steps
Ts=1; %sampling time in seconds
%-----*****File names
thisfilename='NM_EKF_031911a';
pftxtfile='PFpath_WeightedStep061111.txt';
pdatalimit='limitdata061111.txt';
%Note: Units for text document are in METERS
r=dlmread(pftxtfile);%
b=size(r);
r=r';
r(3,1) = atan2(r(2,2)-r(2,1),r(1,2)-r(1,1));
for j=1:b(1)-1
    r(3,j+1) = atan2(r(2,j+1)-r(2,j),r(1,j+1)-r(1,j));
    vvr(1,j+1)= sqrt((r(1,j+1)-r(1,j))^2 +(r(2,j+1)-r(2,j))^2)/Ts;
    vvr(2,j+1)= (r(3,j+1)-r(3,j))/Ts;
end
k=1; %counter
j=1;%counter 2
%-----Initial condn Mobile Robot Model (m)
c(1,1)=r(1,1);
c(2,1)=r(2,1);
```

```

c(3,1)=r(3,1);

%-----Initial Measurement (Noisy Model) (m)
%Simulated noisy system output
xn=c(1,1) +.212*randn(); %m .6
yn=c(2,1) +.212*randn(); %m .6
thn=c(3,1) +.401*randn(); %m .06
cn(:,k)=[xn;yn;thn];
%-----Control parameters and Othe
kv=0.25;
kw=0.25;
vmax =0.7; %m robot max is 700mm/s 3AT and 1400mm/s 3DX
wmax =2; % 2.44 rad/s 3AT and 5.235rad/s 3DX
%//Allocating memory for arrays
Tm = zeros(3,3);
error = zeros(3,1);
error2 = zeros(3,1);
%Kalman Filter Initialization Estimates (m)
%States
xh(:,k)=cn;
%Prediction Covariance Matrix
P(:, :,k)=(cn-c)*(cn-c)';
%Measurement noise convariance Matrix (set to 0)
%W=45.0;
W=[5.3e-5,0,0;0,3.8e-5,0;0,0,0.497]; %m^2, m^2, rad^2 0.497
%Unmodeled System source Covariance Matrix (set to 0)
V=[0.0000001,0,0;0,0.0000001,0;0,0,0.0000001];
%V=[0.,0,0;0,0.,0;0,0,0.];

origin(1)=0; %x

```

```

origin(2)=0; %y
%while abs(r(:,k)-c(:,k))>range
while (k<b(1))
%-----Error transformation and Controller Implementation
    time(k)=k*Ts -Ts;

    %Real Model (no noise)
    error = r(:,k)-c(:,k);
    %Noisy Model
    errorn = r(:,k)-xh(:,k);
    %Noisy Model with sim gps measurement error control
    errorn2 = r(:,k)-cn(:,k);
    %Determine control velocities-----
    vvc(1,k) = kv*(error(1)*cos(r(3,k))/Ts + error(2)*sin(r(3,k))/Ts); %m/s
    vvc(2,k)=kw*(error(3)/Ts); %rad/s
    %Determine control velocities (State Estimate Model)-----
    vvcn(1,k) = kv*(errorn(1)*cos(xh(3,k))/Ts
        + errorn(2)*sin(xh(3,k))/Ts);%m/s
    vvcn(2,k)=kw*(errorn(3)/Ts); %rad/s
    %Determine control velocities (State Estimate Model)-----
    vvcn2(1,k) = kv*(errorn2(1)*cos(cn(3,k))/Ts
        + errorn2(2)*sin(cn(3,k))/Ts);%m/s
    vvcn2(2,k)=kw*(errorn2(3)/Ts); %rad/s

    %Check to ensure Velocity does not exceed max (V,W)
    if (abs(vvc(1,k))>vmax)
        vvc(1,k)=vmax;%*sign(vvc(1,k));
    end
    if (abs(vvc(2,k))>wmax)
        vvc(2,k)=wmax*sign(vvc(2,k));
    end
    %Check Velocity does not exceed max (V,W) for Noisy State

```

```

if (abs(vvcn(1,k))>vmax)
    vvcn(1,k)=vmax;%*sign(vvc(1,k));
end
if (abs(vvcn(2,k))>wmax)
    vvcn(2,k)=wmax*sign(vvcn(2,k));
end
if (abs(vvcn2(2,k))>wmax)
    vvcn2(2,k)=wmax*sign(vvcn2(2,k));
end
%Check Velocity does not exceed max (V,W)
if (abs(vvcn2(1,k))>vmax)
    vvcn2(1,k)=vmax;%*sign(vvc(1,k));
end
if (abs(vvcn2(2,k))>wmax)
    vvcn2(2,k)=wmax*sign(vvcn2(2,k));
end
%KALMAN FILTER*****
%Predicted State Estimate and Covariance-----
xmid(3,1)=vvcn(2,k)*Ts + xh(3,k);
xmid(1,1)=cos(xmid(3,1))*vvcn(1,k)*Ts + xh(1,k);
xmid(2,1)=sin(xmid(3,1))*vvcn(1,k)*Ts + xh(2,k);
F=[1,0, -sin(xh(3,k))*vvcn(1,k)*Ts;
    0,1,cos(xh(3,k))*vvcn(1,k)*Ts;
    0,0,1];
Pk=F*P(:, :, k)*F' + V; %V=0 small
%-----Mobile Robot Kinematic Model (i.e Measurement)
%Simulated noisy system output
x=Ts*vvcn(1,k)*cos(cn(3,k));
xn=x +.212*randn(); %m 0.6

y=Ts*vvcn(1,k)*sin(cn(3,k));
yn=y +.212*randn(); %m 0.6

```

```

th=Ts*vvcn(2,k);
thn=th +.401*randn(); %m 0.06
%simulated measurement (m)
cn(1,k+1) = c(1,k) + xn;
cn(2,k+1) = c(2,k) + yn;
cn(3,k+1) = c(3,k) + thn;
%-----
%real state output (m)
c(1,k+1) = c(1,k) + Ts*vvc(1,k)*cos(c(3,k));
c(2,k+1) = c(2,k) + Ts*vvc(1,k)*sin(c(3,k));
c(3,k+1) = c(3,k) + Ts*vvc(2,k);
%Update-----
H=[1,0,0;0,1,0;0,0,1];
S=H*Pk*H'+W;
R= Pk*H'*inv(S); %Kalman Gain
v=cn(:,k+1)-H*xmid;
xh(:,k+1)= xmid + R*v;
P(:, :,k+1)=Pk - R*H*Pk;
%-----
k=k+1;
end

figure(1)
subplot(2,1,1)
plot(time,vvcn2(2,:), 'r.', tvvr, vvr(2,:), 'k', time,vvcn(2,:), 'k+',
time,vvc(2,:), 'b--', 'LineWidth', 2);
ylabel('Angular velocity (rad/s)');
grid on;
legend('Measurement', 'Reference', 'Estimate', 'State');
subplot(2,1,2)
plot(time,vvcn2(1,:), 'r.', tvvr, vvr(1,:), 'k', time,vvcn(1,:),
'k+', time,vvc(1,:), 'b--', 'LineWidth', 2);

```

```

ylabel('Linear Velocity (m/s)');
xlabel('Time(s)');
grid on;
legend('Measurement', 'Reference', 'Estimate', 'State');
figure(2)
hold on
ymax=85.62;%m
xmax=87.03;%m
xscale=[0;xmax];
yscale=[ymax;0];
axis([0 xmax 0 ymax])
%axis([16 36 30 54])
hold on
ch = imread('mapTrial.jpg');
image(xscale,yscale,ch);
hold on
grid on
plot(cn(1,:),cn(2:,:), 'r.', r(1,:), r(2,:), 'k', xh(1,:), xh(2,:), 'k+',
c(1,:), c(2,:), 'b-', c(1,1), c(2,1), 'ks', c(1,k-1),
c(2,k-1), 'ks', 'LineWidth', 2);
hold on
xlabel('East-West Distance(m)');
ylabel('North-South Distance(m)');
grid on;
legend('Simulated Measurement', 'Ref Vehicle',
'Estimate (State)', 'Actual (State)', '"Start" and "Finish" Points');

```

Bibliography

- [1] Kenneth Waldron, James Schmiedeler, “Chapter 1: Kinematics”, Handbook of Robotics, Springer, 2008.
- [3] Roy Featherstone, David E. Orin, “Chapter 2: Dynamics”, Handbook of Robotics, Springer, 2008.
- [2] Ulrich Nehmzow, “Scientific Methods in Mobile Robotics”, Qualitative Analysis of Agent Behaviour, Springer, 2005.
- [4] Lydia E. Kavraki, Steven M. LaValle, “Chapter 5: Motion Planning”, Handbook of Robotics, Springer, 2008
- [5] Guy Champion, Georges Bastin, and Brigitte D’Andrea-Novel, “Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots”, *IEEE Trans. Robot. Autonom.*, vol. 12, No. 1 February 1996
- [6] R. Fierro and R.L. Lewis, “Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics”, in *Proc. IEEE Conf. on Decision and Control*, New Orleans, LA, December 1995
- [7] B. d’Andrea-Novel, G. Bastin, and G. Champion, “Dynamic Feedback Linearization of Nonholonomic wheeled mobile robot”, *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp.2527-2532 vol.3, 12-14 May 1992
- [8] Giuseppe Oriolo, Alessandro De Luca, and Marilena Vendittelli, “WMR Control via Dynamic Feedback Linearization: Design, Implementation, and Experimental Validation”, in *Proc. of the IEEE Trans. on Control Systems Technology*, vol. 10, No. 6, November 2002
- [10] Chang Boon Low and Danwei Wang, “GPS-Based Path Following Control for a Car-like Wheeled Mobile Robot with Skidding and Slipping”, in *IEEE Trans. on Control Systems Technology* vol. 16 No. 2, March 2008
- [11] Luis Aguilar, Terek Hamel, Philippe Soueres, “Robust Path Following Control for Wheeled Robots via Sliding Mode Techniques”, in *Intelligent Robots and Systems, 1997. IROS ’97., Proceedings of the 1997 IEEE/RSJ International Conference on* vol.3, pp.1389-1395, 7-11 Sep 1997
- [12] Robert W. Hogg, Arturo L. Rankin, et. al, “Algorithms and Sensors for Small Robot Path Following”, in *Proc. of 2002 IEEE International Conf. of Robotics and Automation*, May 2002

- [13] Paulo Coelho and Urbano Nunes, "Path-Following Control of Mobile Robots in Presence of Uncertainties", in *IEEE Transaction on Robotics*, vol. 21, no. 2, April 2005
- [14] Kazunori Ohno, Takashi Tsubouchi, et. al, "Outdoor Navigation of a Mobile Robot between Buildings based on DGPS and Odometry Data Fusion", in *Proc. of the 2003 IEEE International Conf. on Robotics and Automation*, September 14-19, 2003
- [15] Alejandro Ramirez-Serrano, Giovanni C. Pettinaro, "Navigation of Unmanned Vehicles Using a Swarm of Intelligent Dynamic Landmarks", in *Proc. of the 2005 IEEE International Workshop on Safety, Security and Rescue Robotics*, June 2005
- [16] Yutaka Kanayama, Yoshihiko Kimura, Fumio Miyazaki, Tetsuo Moguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot", in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, vol.1, pp.384-389, 13-18 May 1990
- [17] Shi Xingxi, Wang Tiesheng, "Cooperative Multi-Robot Localization Based on Distributed UKF", in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol.6, pp.590-593, 9-11 July 2010
- [18] Khalil, H. K, "Nonlinear Systems", Third Edition, Prentice-Hall, 2002.
- [19] Isaac Skog and Peter Handel, "In-Car Positioning and Navigation Technologies - A Survey", in *IEEE Trans. on Intelligent Transportation Systems*, vol. 10, no. 1, March 2009
- [20] Hwan-Seok Choi*, Ok-DeukPark*, Han-Sil Kim, "Autonomous Mobile Robot Using GPS", in *2005 International Conf on Control and Automation*, June 27-29, 2005, Budapest Hungary
- [21] Veerachai Malyavej, Peerapong Torteeka, et. al. "Pose Estimation of Unmanned Ground Vehicle based on Dead-reckoning/GPS sensor fusion by Unscented Kalman Filter", in *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009. ECTI-CON 2009. 6th International Conference on*, vol.01, pp.395-398, 6-9 May 2009
- [22] SangJoo Kwon, Kwang Woong Yang and Sangdeok Park, "An Effective Kalman Filter Localization Method for Mobile Robots", in *Proc. of the 2006 IEEE/RSJ International Conf. on Intelligent Robots and Systems*, October 9 - 15, 2006, Beijing, China
- [23] C. Samson, "Time-varying feedback stabilization of car-like wheeled mobile robots", in *Int. J. Robot. Res.*, vol. 12, No. 1, pp. 55-64, 1993

- [24] R. Fierro and R.L. Lewis, "Control of a Nonholonomic Mobile Robot Using Neural Network", in *IEEE Trans. on Neural Networks*, Vol. 9, No. 4, pp. 589-600, July 1998
- [25] Eric Abbott and David Powell, "Land-Vehicle Navigation Using GPS", in *Proc. of the IEEE*, vol. 87, no. 1, January 1999
- [26] Atanas Georgiev and Peter K. Allen, "Localization Methods for a Mobile Robot in Urban Environments", in *IEEE Transaction on Robotics*, vol. 20, no. 5, October 2004
- [27] T.D Barfoot, C.M. Clark, "Motion Planning for formations of mobile robots", in *Robotics and Autonomous Systems*, 46, pp. 65-78, October 2004
- [28] John Markoff, "Smarter Than You Think - Google Cars Drive Themselves, in Traffic", in <http://www.nytimes.com/2010/10/10/science/10google.html>, Science, New York Times, Published October 9, 2010; viewed October 17, 2010
- [29] "The DARPA Urban Challenge", in <http://www.darpa.mil/grandchallenge/index.asp>, Held on November 3, 2007 in Victorville, Calif. USA ; viewed February 16, 2011
- [30] Jose-Luis Peralta-Cabezas, Miguel Torres-Torriti, Marcelo Guarini-Herrmann, "A Comparison of Bayesian Prediction Techniques for Mobile Robot Trajectory Tracking", in *ICM 2006 IEEE 3rd International Conference on Mechatronics*, pp. 442-447, 2006
- [31] Shahram Rezaei, and Raja Sengupta, "Kalman Filter-Based Integration of DGPS and Vehicle Sensors for Localization", in *IEE Trans. on Control Systems Technology*, vol. 15, no. 6, November 2007
- [32] Stefano Panzieri, Federica Pascucci, and Giovanni Ulivi, "An Outdoor Navigation System Using GPS and Inertial Platform", in *IEEE/ASME Trans. on Mechatronics*, vol. 7, no. 2, June 2002
- [33] Giulio Reina, Andres Vargas, Keiji Nagatani and Kazuya Yoshidai, "Adaptive Kalman Filtering for GPS-based Mobile Robot Localization", in *Proc of the 2007 IEEE International Workshop on Safety, Security and Rescue Robotic*, Rome, Italy, September 2007
- [34] Henrik I. Christensen, Gregory D. Hager, "Chapter 4: Sensing and Estimation", *Handbooks of Robotics*, Springer, 2008 pp 96-97.
- [35] Greg Welch and Gary Bishop, "An Introduction to the Kalman Filter", in <http://www.cs.unc.edu/welch/kalman/kalmanIntro.html>, Last modified: Thursday, March 8, 2007; viewed June 2010
- [36] Gregory Dudek, Michael Jenkin, "Chapter 20: Inertial Sensors, GPS, and Odometry", *Handbooks of Robotics*, Springer, 2008.

- [37] Lydia E. Kavraki, Steven M. LaValle, “Chapter 5: Motion Planning”, Handbooks of Robotics, Springer, 2008.
- [38] O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots”, in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol.2, pp. 500- 505, Mar 1985
- [39] Pu Shi, Yiwen Zhaoui, “An Efficient Path Planning Algorithm for Mobile Robot Using Improved Potential Field”, in *Proc. of the 2009 IEEE International Conference on Robotics and Biomimetics*, December 19 -23, 2009, Guilin, China
- [40] Viorel Stoian, Mircea Ivanescu, et. al. “Using Artificial Potential Field Methods and Fuzzy Logic for Mobile Robot Control”, in *Power Electronics and Motion Control Conference, 2006. EPE-PEMC 2006. 12th International*, pp.385-389, Aug. 30 2006-Sept. 1 2006
- [41] Gustavo Scaglia, Luca Quintero Montoya, et. al., “Numerical methods based controller design for mobile robots”, in *Robotica (2009)*, volume 27, pp. 269-279.
- [42] Charles W . Warren, “Multiple Robot Path Coordination Using Artificial Potential Fields”, in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp.500-505 vol.1, 13-18 May 1990.
- [43] Howie M . Choset, “Principles of robot motion: theory, algorithms, and implementation”, Cambridge MA, MIT Press, 2005
- [44] Anjan Kumar Ray, Laxmidhar Behera and Mo Jamshidi, “GPS and Sonar Based Area Mapping and Navigation by Mobile Robots”, in *2009 7th IEEE International Conference on Industrial Informatics (INDIN 2009)*, pp. 801-806.
- [45] Movable Type Ltd “Calculating distance and bearing between two Latitude/Longitude points using Haversine formula in JavaScript”, in <http://www.movable-type.co.uk/scripts/latlong.html> , visited on February 7, 2011
- [46] Derek Rowell, David N. Wormley “System Dynamics: An Introduction”, Prentice Hall, Upper Saddle River, NJ, 1997, pp. 368-75
- [47] Simon J. Julier, and Jeffrey K. Uhlmann, “Unscented Filtering and Nonlinear Estimation”, in *Proceedings of the IEEE* , ol.92, no.3, pp. 401- 422, Mar 2004
- [48] Nancy Amato, “Potential Field Methods”, in <http://parasol.tamu.edu/amato/Courses/padova04/> , Randomized Motion Planning, Fall 04, Univ. of Padova
- [49] Min Gyu Park, Jae hyun Jeon and Min cheol Lee, “Obstacle Avoidance for Mobile Robots using artificial potential field approach with simulated annealing”, in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on* , ISIE 2001, Pusan, Korea

- [50] Lei Tang, Songyi Dian- Gangxu Gu, et. al., “A Novel Potential Field Method for Obstacle Avoidance and Path Planning of Mobile Robot”, in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol.9, pp.633-637, 9-11 July 2010
- [51] Katsuya Sekiguchi, Mingcong Deng and Akira Inoue, “Obstacle Avoidance and Two Wheeled Mobile Robot Control Using Potential Function”, in *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, pp.2314-2319, 15-17 Dec. 2006
- [52] Nasser Houshangi, Farouk Azizi, “Accurate Mobile Robot Position Determination using Unscented Kalman Filter”, in *2005 IEEE CCECE/CCGEI*, Saskatoon, May 2005
- [53] Rachana A. Gupta and Mo-Yuen Chow, “Overview of Networked Control Systems”; *Network Control Systems*, pp. 1-23, 2008
- [54] Mark Ward, “‘Road train’ technology trials get rolling”, in <http://www.bbc.co.uk/news/technology-12215915>, BBC News - Technology, Published January 18, 2011; viewed February 10, 2011
- [55] Grewal Mohinder S, “Global Positioning Systems, Inertial Navigation, and Integration.”, 2nd Edition, Published Hoboken : John Wiley and Sons, Inc., 2007.
- [61] Thomas Dall Larsen, Karsten Lentfer Hansen, Nils A. Andersen and Ole Ravn, “Design of Kalman Filters for Mobile Robots; Evaluation of the Kinematic and Odometric Approach”, in *Proc. of the 1999 IEEE International Conference on Control Applications*, pp. 1021-1026, August 22-27, 1999
- [62] “WAAS in Canada”, in <http://members.shaw.ca/pdops/WAAS.html>; Last updated 2011/04/22; visited on May 10, 2011
- [63] Adept MobileRobots, “Pioneer Operation Manual”, in www.ist.tugraz.at_attach/Publish/Kmr06/pioneer-robot.pdf; visited on May 10, 2011
- [64] Microsoft Online Bing Map, Sexton Campus Parking Lot Overhead View Image, <http://www.bing.com/maps/>; viewed on October, 2010
- [65] James Thomas, “Wiiuse.cpp C++ Class for Wiimote”, in <http://www.wiiuse.net/>; viewed on January 3 2011
- [66] Hemisphere GPS, “GPS Technical Reference: Part No. 875-0175-000 Rev. D1”, Hemisphere GPS Precision GPS Applications, 2008.