A State Estimation Approach for a Skid-Steered Off-Road Mobile Robot

by

Mohammad Azam Javed

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Mechanical Engineering

Waterloo, Ontario, Canada, 2013

© Mohammad Azam Javed 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

This thesis presents a novel state estimation structure, a hybrid extended Kalman filter/Kalman filter developed for a skid-steered, six-wheeled, ARGO® all-terrain vehicle (ATV). The ARGO ATV is a teleoperated unmanned ground vehicle (UGV) custom fitted with an inertial measurement unit, wheel encoders and a GPS. In order to enable the ARGO for autonomous applications, the proposed hybrid EKF/KF state estimator strategy is combined with the vehicle's sensor measurements to estimate key parameters for the vehicle. Field experiments in this thesis reveal that the proposed estimation structure is able to estimate the position, velocity, orientation, and longitudinal slip of the ARGO with a reasonable amount of accuracy. In addition, the proposed estimation structure is well-suited for online applications and can incorporate offline virtual GPS data to further improve the accuracy of the position estimates. The proposed estimation structure is also capable of estimating the longitudinal slip for every wheel of the ARGO, and the slip results align well with the motion estimate findings.

Acknowledgements

First of all, I am thankful to God for facilitating my thirst for knowledge, and for blessing me with a family that truly understands me.

I would like to thank my supervisors Prof. William Melek, Dr. Bill Owen and Dr. Mohammad Biglarbegian, who have supported, guided and were patient with me at every step.

I would like to thank my friend Dr. Peter Won for all of his help, and for being one of the few people that had some answers for me when nobody else did.

I would also like to thank our Research Assistants: Yifeng Li and Rajan Gill, and the guys at the lab: Robert Wagner, Kevin Cochran and Jeff Graansma, for all of their help with the ARGO vehicle.

I would like to mention the continued help and support of all my friends that were always willing to lend an ear, and to anyone else that has helped me in any way or form with my research. Thank you all so much.

Dedication

To my Mother and Father.

Table of Contents

Author's Declaration iii

Abstract iii

Acknowledgements iv

Dedication v

Table of Contents vi

List of Figures viii

List of Tables x

Chapter 1. Introduction 1

Chapter 2. State Estimation Fundamentals 6

2.1. The State of a Robot 6

2.2. Probabilistic Generative Laws 7

2.3. Belief Distribution 8

2.4. Bayes Algorithm 9

2.5. Gaussian Filters 10

2.5.1. Kalman Filter 10

Chapter 3. ARGO Motion Modelling 13

3.1. ARGO Platform 13

3.2. Longitudinal Slip 15

3.3. Kinematic Model 18

Chapter 4. State Estimation 21

4.1. Structure of the Proposed Estimator 21

4.2. Extended Kalman Filter 25

4.2.1. System and input models 25
4.2.2. Measurement model 29
4.2.3. Extended Kalman filter algorithm 29
4.3. Kalman Filter 31
4.3.1. System and input models 32
4.3.2. Measurement model 32
4.3.3. Kalman filter algorithm 37
Chapter 5. Experiments 38
5.1. Experimental Setup 38
5.2. Double-lane Change Experiment 41
5.2.1. Position estimates 41
5.2.2. Longitudinal slip 46
5.2.3. Velocity estimates 48
5.2.4. Orientation estimates 49
5.3. Slalom Experiment 52
5.3.1. Position estimates 53
5.3.2. Longitudinal slip 57
5.3.3. Velocity estimates 58
5.3.4. Orientation estimates 59
5.4. Fishhook Experiment 61
5.4.1. Position estimates 61
5.4.2. Longitudinal slip 65
5.4.3. Velocity estimates 65
5.4.4. Orientation estimates 66
Chapter 6. Conclusions and Future Work 70
References 72

List of Figures

Figure 3.1. ARGO structure: hardware, sensors and actuators..... 14

Figure 3.2. A top-view schematic of the skid-steered ARGO vehicle showing the body and inertial reference frames. 15

Figure 3.3. A schematic of the distances from the center of gravity of the vehicle to the contact points of the various wheels. 17

Figure 4.1. State estimate - IMU/encoder data fused with GPS data. 23

Figure 4.2. Structure of the proposed hybrid EKF/KF state estimator. 24

Figure 4.3. GPS data representation in the inertial frame. 33

Figure 5.1. ARGO vehicle and test rig. 39

Figure 5.2. Schematic of double-lane change maneuver. 39

Figure 5.3. Schematic of slalom maneuver. 40

Figure 5.4. Schematic of fishhook maneuver. 40

Figure 5.5. EKF position estimates – double-lane change experiment. 42

Figure 5.6. EKF state prediction – double-lane change experiment..... 43

Figure 5.7. KF position estimates – double-lane change experiment. 44

Figure 5.8. Longitudinal slip – double-lane change experiment. 47

Figure 5.9. Estimated velocity in X and Y – double-lane change experiment. 48

Figure 5.10. Predicted velocity in X and Y – double-lane change experiment. 49

Figure 5.11. Yaw, pitch, roll estimates – double-lane change experiment..... 50

Figure 5.12. Yaw estimate comparison – double-lane change experiment. 51

Figure 5.13. Estimated yaw, pitch and roll rates – double-lane change experiment. 52
Figure 5.14. EKF position estimates – slalom experiment. 53
Figure 5.15. EKF prediction estimate – slalom experiment. 54
Figure 5.16. KF position estimates – slalom experiment. 55
Figure 5.17. Longitudinal slip – slalom experiment. 57
Figure 5.18. Estimated velocity in *X* and *Y* – slalom experiment. 58
Figure 5.19. . Predicted velocity in *X* and *Y* – slalom experiment. 58
Figure 5.20. Yaw, pitch, roll estimates – slalom experiment. 59
Figure 5.21. Yaw estimate comparison – slalom experiment. 60
Figure 5.22. Estimated yaw, pitch and roll rates – slalom experiment. 60
Figure 5.23. EKF position estimates – fishhook experiment. 61
Figure 5.24. EKF prediction estimate – fishhook experiment. 62
Figure 5.25. KF position estimates – fishhook experiment. 63
Figure 5.26. Longitudinal slip – double-lane change experiment. 65
Figure 5.27. Estimated velocity in *X* and *Y* – fishhook experiment. 66
Figure 5.28. Predicted velocity in *X* and *Y* – slalom experiment. 66
Figure 5.29. Yaw, pitch, roll estimates – fishhook experiment. 67
Figure 5.30. Yaw estimate comparison – fishhook experiment. 68
Figure 5.31. Estimated yaw, pitch and roll rates – slalom experiment. 68

List of Tables

Table 2.1. Bayes filter algorithm. 9

Table 2.2. Kalman filter algorithm. 11

Table 4.1. Virtual GPS point generation algorithm. 35

Table 5.1. RMS data – double-lane change experiment. 45

Table 5.2. RMS values – double-lane change experiment. 46

Table 5.3. RMS data – slalom experiment. 56

Table 5.4. RMSD values – slalom experiment. 56

Table 5.5. RMS data – fishhook experiment. 64

Table 5.6. RMS values – fishhook experiment. 64

Chapter 1.

Introduction

Unmanned Ground Vehicles (UGVs) have gained a lot of popularity in recent years and have far-reaching applications ranging from the use of UGVs in collecting soil samples on Mars to being deployed in military missions. UGVs are primarily either teleoperated by an operator or are autonomous. In the case of teleoperated UGVs in military applications, the complete attention of the operator as well as some form of high-bandwidth remote video are required to ensure the UGV performs its tasks fittingly [1]. Autonomous UGVs, on the other hand, are not limited by these requirements. Autonomous UGVs are equipped with various sensors, hardware and software systems enabling them to perform tasks in static and dynamic environments reliably, mostly without an operator [1].

The ARGO autonomous UGV system presented in this thesis is a six-wheel drive, skid-steered, ARGO all-terrain vehicle (ATV) custom-fitted with a GPS, wheel encoders, an inertial measurement unit (IMU), steering actuators and a Netbook PC, for use in military reconnaissance applications in off-road environments. The ARGO UGV system is teleoperated, and the proposed state estimator developed in this rotation lays the groundwork for the future-development of an autonomous ARGO UGV capable of localization, map-building and navigation in unstructured environments.

State estimation or probabilistic state estimation, is at the core of most robust autonomously controlled systems today [2]. This approach maintains the probability densities of states over time [3]. These calculated probability densities can be influenced by sensor measurements, control inputs, prediction models and their probabilistic uncertainties [2]. Some of the early work that adopted probabilistic state estimation in mobile robots was applied to autonomous museum tour robots [4,5] and autonomous office robots [6]. Additionally, state

estimation algorithms such as the Kalman Filter (KF) or the Extended Kalman Filter (EKF) can be used in conjunction with sensors to derive estimates from parameters that cannot directly be measured with sensors [3]. For example, parameters such as wheel slip, lateral/longitudinal velocities and the angular rates at the center of gravity of the vehicle are a few of the many parameters essential in autonomous UGV work or in simultaneous localization and mapping (SLAM).

SLAM, is a class of problems in which a mobile robot creates a spatial map of its surroundings, while simultaneously localizing itself within the created map [7]. The integration of a light detection and ranging (LIDAR) sensor on a UGV such as the ARGO can be utilized in order to enable SLAM. A LIDAR sensor can be used in acquiring 2D or 3D point clouds of data [8]. Spatial 2D or 3D maps can be generated from point cloud data using map generation algorithms. The scan matching approach [9,10] and the maximum likelihood approach [11,12], are the two most predominant categories of such algorithms for constructing maps using LIDAR [13].

Moreover, reliable and robust path planning systems [14,15] are also important in autonomous vehicles [16]. These systems generate a path for the vehicle to traverse through the created map and the vehicles path following system navigates the vehicle on its planned path while accounting for the vehicle dynamics and path constraints [17,18]. On that note, the use of 'Lyapunov techniques' are a popular approach in path following problems for skid-steered vehicles [19,20] and mobile robots [21].

To keep the vehicle on course its path, adjustments to the vehicles control actuators must continuously be made [16]. The vehicles path following system communicates with the actuators that operate the vehicles motion based on the position, velocity, heading angle, and wheel torque information extracted from the vehicles sensors and the vehicle model via state estimators. An improvement in the vehicles sensors and/or the vehicle model translates into improvements in the accuracy and reliability of the estimated states and the control of the vehicle. In [18], the use of a real time kinematic (RTK)-differential global positioning system (DGPS) instead of a stand-alone DGPS coupled with dead-reckoning techniques resulted in substantial improvements in autonomous driving accuracy. Alternatively, improvements to the vehicle model can be made to achieve improved autonomous control as well; it has been envisioned that a critical part in

developing more sophisticated vehicle control systems in the future depends on obtaining more up-to-date and accurate vehicle dynamics information [22].

The use of a dynamic model in autonomous UGVs enables accurate motion predictions in unstructured outdoor environments that consist of changes in elevation, terrain and require frequent changes in acceleration [23]. The vehicle model used for the ARGO in this thesis adopts a differential drive kinematic model from [24] to approximate the motion of the vehicle. The kinematic model is used in mapping the wheel velocities to the vehicle velocities, and is an important component in developing a vehicle dynamic model [23].

Wheeled and/or tracked skid-steered vehicles are popular in off-road mobile robot applications due to their simple mechanical configuration, improved manoeuvrability and traction [25]. Early work in the dynamics of wheeled and tracked skid-steered vehicles in [26-28] demonstrated the complexity of this traction scheme because the wheels (or tracks) roll and slide at the same time, making it difficult to develop the kinematic and dynamic equations [23]. The major difficulty in modelling this type of vehicle stems from the complex wheel terrain interaction [23], as such, approximations of these non-linear relationships are necessary for mobile robot applications [23,29-31]. [32] presents a fresh perspective to skid-steer vehicles by describing how the effects of the centrifugal force must be taken into consideration, especially when skid-steered vehicles perform turns at high speeds with a small turning radius. Other literature related to robot dynamics in four-wheeled differentially driven robots is presented in [33]; where a dynamic model is developed to account for wheel skidding with the use of a model-based nonlinear controller for accurate vehicle control.

A composite rigid-body robot dynamics algorithm is presented in [34], where the rigid-body dynamic equations have been presented in a way that makes it easy for one to process the dynamic equations using software tools. In [25] and [35], a simplified approach is presented for developing models that can provide accurate pose and motion estimates in autonomous navigation and control applications. [36] presents robust controllers called 'sliding-mode controllers' that account for the sliding phenomena exhibited by skid-steer vehicles. In [37], a sliding mode controller combined with a proportional-integral-derivative (PID) controller is proposed as a control strategy for the braking system of an eight-wheel drive ARGO vehicle, and the issues related to developing such control strategies is presented in [38].

A pre-cursor to the development of a dynamic model is a driveline model. A commonly used nonlinear mathematical model of a four-cylinder spark ignition engine is presented in [39]. Simplified models for other driveline components such as the clutch, transmission, propeller shaft, and final drive differential are presented in [40] in an off-road truck example. The fundamental components, configurations and kinematics of a continuously variable transmission (CVT) are presented in [41], where the wheel speed control problem for a vehicle equipped with a CVT is considered. The dynamic model of a vehicle transmission system is presented in [42] using a graphical modelling approach. The driveline of a eight-wheel drive ARGO vehicle was formulated, modelled, simulated, and validated experimentally in [43].

Moreover, accurate estimation of slip parameters is necessary for developing advanced localization and navigation strategies [44]. [44] proposes a vision-based approach to estimating slip, whereas, [45] suggests a simplified state estimator approach. In [46], details on a simulation platform to develop and model the longitudinal and lateral control laws for a skid-steered mobile robot are presented. [47] proposes a fuzzy-logic based navigation system that can correct wheel slippage using an IMU and wheel encoders on a Mars Rover robot platform. In this thesis, a similar approach in calculating slip for a skid-steered UGV is taken. The existing estimation methods in the literature are not suited for the design of a state observer strategy for a skid-steered ARGO UGV. Thus, a thorough analysis that consolidates the ARGO's motion, orientation and longitudinal slip is needed for the design of such a state observer. To address this issue, this thesis offers the following key contributions: 1) a thorough analysis of an ARGO's position, velocity, orientation and wheel slips while driving through experimental tracks arranged according to the SAE double-lane change, slalom and fishhook maneuvers, 2) the development of an online-ready hybrid EKF/KF state estimation technique that estimates the above stated parameters for an ARGO, 3) a virtual GPS point generation algorithm that can be used to adjust position estimates using a GPS sensor.

The novel contributions in this thesis pertaining to the longitudinal slip estimation and state observer development is essential for motion planning and control in future dynamic modelling and SLAM research using the ARGO platform.

The rest of this thesis is organized as follows:

Chapter 2: State Estimation Fundamentals

This chapter presents background information on state estimation, and touches on key concepts such as: the state of a robot, probabilistic generative laws, belief distribution, the Bayes Algorithm, and Gaussian filters. The material presented in this chapter has been primarily adopted from [2].

Chapter 3: ARGO Motion Modelling

The end goal of this chapter is to present a general overview of the ARGO and develop a simple model that can describe the motion of the vehicle. This chapter is broken down into three main sections: ARGO platform, longitudinal slip and kinematic model.

The novel contributions presented in this chapter pertain to the method by which the longitudinal slips of all six wheels on the ARGO are calculated.

Chapter 4: State Estimation

This chapter presents details on the proposed hybrid EKF/KF for the ARGO vehicle, and it is split into three sections: the structure of the proposed state estimator, the extended Kalman filter and the Kalman filter.

The novel contributions presented in this chapter pertain to the EKF/KF state estimator strategies used in estimating the ARGO's motion, orientation, and longitudinal slip, and the virtual GPS point generation algorithm that can be used to further adjust the position estimates generated by the EKF.

Chapter 5: Experiments

This chapter is split into three sections, and presents details on experimental setup and findings for the ARGO vehicle while it performs the double-lane change, slalom and fishhook maneuvers.

The novel contributions presented in this chapter are the thorough analysis of the ARGO's position, velocity, orientation and wheel slips while it is performing the three maneuvers.

Chapter 6: Conclusions and Future Work

Chapter 6 concludes the thesis and discusses the future work.

Chapter 2.

State Estimation Fundamentals [2]

This chapter presents a discussion on the fundamentals of state estimation. Section 1 introduces the concept of the 'state' of a robot, Section 2 describes some of the probabilistic generative laws that play an important role in developing state estimation algorithms, Section 3 discusses the concept of belief distributions in state estimation, Section 4 presents the Bayes Algorithm; which is the algorithm at the heart of all state estimation algorithms, and Section 5 expands the Bayes Algorithm to Gaussian Filter applications, and describes the basic structure of a Kalman filter.

2.1. The State of a Robot

The environment or the world that a robot interacts with is connected with the robot through a dynamical system, and this dynamical system possesses an internal state, which the robot keeps track of while it interacts with its environment. The robot can acquire information about its environment using its sensors; however, these sensors are noisy and there are many parameters important to the robot that cannot be sensed directly. Additionally, the robot can influence its environment through its actuators, except, the effect of doing so is often unpredictable.

As a consequence, the robot must maintain an internal belief regarding the state of its environment, thus, one can think of state as the collection of all aspects of the robot and its environment that are important to how the robot interacts in its environment in the near future. In simple terms, the state variables for a robot are a set of variables that tell the robot information about itself and its environment.

State variables can give the robot information like its position, velocity, orientation and whether or not its sensors are functioning, or information about its environment, e.g.; the distance to an object that is being tracked by the robot, whether or not there is an obstacle in front of the robot and where the robot is located with respect to a map that it has generated of its surroundings. Additionally, a robot may also keep track of its measurements (camera images of its environment), and control actions (velocity of robot in its environment).

A state can be defined continuously or discretely, and is called a state space. An example of a state defined continuously (or represented by a continuous state space) is the robot pose (location/orientation) in a particular coordinate system. Similarly, an example of a discrete or binary state variable (or a state represented in a discrete state space) is a model of a broken sensor. A state space that contains both continuous and discrete state variables is a hybrid state space.

A robot may keep track of all measurements and control actions which generate two sets of data:

1) Measurement data – provides information about a momentary state of the environment or the robot (e.g. camera images)
2) Control data – provides information on the change of state variables in the environment (e.g. velocity of the robot)

In state estimation and probabilistic robotics, measurement data is represented as z_t, and control data is represented as an input u_t.

2.2. Probabilistic Generative Laws

The evolution of state and measurements are governed by probabilistic laws; where state information x_t, measurement data z_t, and control data u_t are generated stochastically. One might think that

$$p(x_t|x_{o:t-1}, z_{1:t-1}, u_{1:t}) \tag{1}$$

where the current state x_t of a robot is a distribution of the initial state $x_{o:t-1}$, previous measurement data $z_{1:t-1}$, and current control inputs $u_{1:t}$. However, x_{t-1} is a sufficient statistic of all previous states and measurements as shown in Equation (2).

The state transitional probability and measurement probability in probabilistic robotics are two probability distributions when combined together, can describe the complete statistical system of the robot and its environment. The state transitional probability can be represented as

$$p(x_t|x_{o:t-1}, z_{1:t-1}, u_{1:t}) = p(x_t|x_{t-1}, u_t) \quad (2)$$

where x_{t-1} are all previous states of the robot and u_t new control input. The state transitional probability in (2) specifies how environment state variables evolve over time as a function of robot controls.

On the other hand, the measurement probability specifics the probability of measurements generated from the environment. It is presented as

$$p(x_t|x_{o:t-1}, z_{1:t-1}, u_{1:t}) = p(z_t|x_t) \quad (3)$$

Furthermore, the measurement probability in (3), is especially useful if one wants to model the process by which measurements from sensors are generated.

2.3. Belief Distribution

Belief for a robot reflects its internal knowledge about the state of its environment. There are states that cannot be directly measured, e.g., a robots pose might be $x_t = (15.1\ m, 11.5\ m, 0.855\ rad)$, where the pose ($x_t$) is the *x, y* position and the yaw of the robot in some global coordinate system. Therefore, a state such as pose, is not directly measureable, even with the use of a sensor such as a GPS. A state such as the pose of the robot must be inferred from various data-sets collected by the robot through its sensors and control inputs.

Moving forward, beliefs are represented through conditional probability distributions, in which, a belief distribution is assigned to each possible hypothesis with regards to the true state.

Belief distributions are posterior probabilities over state variables that are conditioned on the available data, represented as

$$bel(x_t) = p(x_t|z_{1:t}, u_{1:t}) \tag{4}$$

where $bel(x_t)$ is the posterior probability; which is the probability distribution over the state x_t at time t, condidioned on all past measurements $z_{1:t}$ and all past controls $u_{1:t}$.

A posterior can also be calculated before incorporating z_t and after control input u_t is executed. This is denoted as

$$\overline{bel}(x_t) = p(x_t|z_{1:t-1}, u_{1:t}) \tag{5}$$

where $\overline{bel}(x_t)$ predicts the state at time t, based on the previous state posterior, before incorporating the measurement at time t. In context of applications involving filtering, (5) is known as the prediction update, and calculating $bel(x_t)$ from $\overline{bel}(x_t)$ is called correction, or measurement update as discussed in the next section.

2.4. Bayes Algorithm

The general algorithm for calculating beliefs is the Bayes Algorithm or the Bayes Filter. It calculates the belief distribution $bel(x_t)$ from measurement and control data recursively. The Bayes Filter is presented in Table 2.1, followed by a line by line explanation of the filter.

Table 2.1. Bayes filter algorithm.

	Bayes Filter $(bel(x_{t-1}), z_t, u_t)$*:*
1.	$for\ all\ x_t do$
2.	$\overline{bel}(x_t) = \int p(x_t\|u_t, x_{t-1}) \cdot bel(x_{t-1}) \cdot dx\}$
3.	$bel(x_t) = \eta(p(z_t\|x_t) \cdot \overline{bel}(x_t))$
4.	$end\ for$
5.	$return\ bel(x_t)$

The inputs to the Bayes Filter are: the belief at x_{t-1}, and the most up to date measurement data z_t and control data u_t.

The Bayes Filter Algorithm in Table 2.1 has two parts: the prediction update in Line 2 and the measurement update in Line 3.

1) In the prediction update step, the control input data is integrated into the belief by applying the sum of two distributions, namely; the probability ($p(x_t|x_{t-1}, u_t)$ that the control inputs will induce a transition from x_{t-1} to x_t, and the prior belief ($bel(x_{t-1})$).
2) The measurement update multiplies $\overline{bel}(x_t)$ with the probability that the measurement z_t may have been observed. This is done for every hypothetical posterior state x_t. The resultant product is generally not a probability and may not integrate to one. Therefore, it must be normalized, as indicated by η.

In Line 5, after the update rules are applied recursively to calculate the belief at x_t from $bel(x_{t-1})$, $bel(x_t)$ is finally calculated.

2.5. Gaussian Filters

Gaussian filters are another form of recursive state estimators and share the same basic concepts:

- Beliefs are taken as multivariate normal distributions.
- These multivariate normal distributions are a set of distributions characterized by two parameters: a mean (μ) and a covariance (Σ).

Representing the posterior by a Gaussian means that the posterior distribution is unmoral (one maxima), because of the Gaussian distribution. As such, the posterior is focused around a true state with a small margin of uncertainty.

2.5.1. Kalman Filter

A Kalman filter (KF) is used for filtering and prediction in Linear Systems, whereas, the extended Kalman filter (EKF) is used in Non-Linear Systems.

In the KF, beliefs are computed for continuous states. At time t, the belief is represented by mean μ_t and covariance Σ_t. Table 2.2 shows the KF algorithm followed by a line by line explanation of the algorithm.

Table 2.2. Kalman filter algorithm.

	Kalman Filter $(\mu_{t-1}, \Sigma_{t-1}, z_t, u_t)$*:*
1.	$\bar{\mu}_t = A_t \cdot \mu_{t-1} + B_t \cdot u_t$
2.	$\bar{\Sigma}_t = A_t \cdot \Sigma_{t-1} \cdot A_t^T + R_t$
3.	$K_t = \Sigma_t \cdot C_t^T (C_t \cdot \bar{\Sigma}_t \cdot C_t^T + Q_t)^{-1}$
4.	$\mu_t = \bar{\mu}_t + K_t(z_t - C_t \cdot \bar{\mu}_t)$
5.	$\Sigma_t = (I - K_t \cdot C_t)\Sigma_t$
5.	$return\ \mu_t, \Sigma_t$

The input to the KF algorithm depicted in Table 2.2 is the belief at time $t-1$, represented by the mean (μ_{t-1}) and covariance (Σ_{t-1}). To update these parameters, the KF requires the most recent measurement (z_t) and control (u_t) data. The output from the algorithm is the belief at time t, represented by the latest μ_t and Σ_t.

In Lines 1 and 2, the predicted belief $\overline{bel}(x_t)$ represented by $\bar{\mu}_t$ and $\bar{\Sigma}_t$ is calculated. This belief is calculated before the measurement z_t is incorporated. The matrices A_t and B_t represent the state and control matrices used in incorporating the latest control inputs u_t, and R_t is the system noise matrix.

In Line 3 through 5, the belief $\overline{bel}(x_t)$ is transformed into the desired belief $bel(x_t)$ by incorporating the measurement z_t. The Kalman gain K_t calculated in Line 3, specifies the degree to which z_t is incorporated into the new state, where C_t is the matrix representing the measurement model and Q_t is the measurement noise matrix. Line 4 adjusts the mean according to the measurement, the predicted mean and the Kalman gain. And in Line 5, the updated covariance of the posterior belief is calculated, adjusting for the information gain from the measurement.

In the next chapter of this thesis, the ARGO vehicle's platform is presented, followed by the formulation of the longitudinal slip and kinematic models used in developing the state estimation strategy.

Chapter 3.

ARGO Motion Modelling

In this chapter, background information on the ARGO vehicle platform and the motion model are presented. Section 1 presents a thorough discussion on the ARGO vehicle's configuration and the assumptions pertaining to the vehicle's model, Section 2 details the longitudinal slip model developed for the ARGO, and Section 3 develops the kinematic model for the vehicle.

3.1. ARGO Platform

The vehicle platform used for this research is an ARGO ATV. This vehicle has six driving wheels (three on each side), seats two people, is powered by a small 18 HP gasoline engine, weighs about 450 kg, and is a skid-steered vehicle.

The ARGO has been equipped with a Microstrain® 3DM-GX2 IMU, a Garmin® 18x LVC GPS, Grayhill® 63R128 optical rotary encoders (with a resolution of 128 pulses per revolution) mounted on two of the rear wheels of the vehicle, a Compaq Netbook PC running Ubuntu® version 11.10 and an open-source real-time operating system called 'Robot Operating System (ROS) Electric'. Figure 3.1 shows the locations of the sensors, actuators, and other hardware on the vehicle.

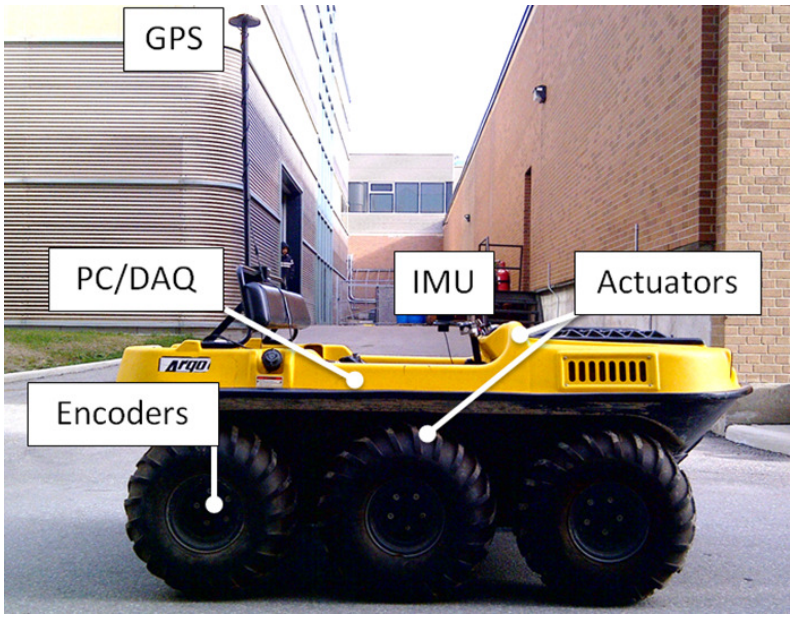

Figure 3.1. ARGO structure: hardware, sensors and actuators.

There is an Arduino microcontroller-based board mounted near the PC on the vehicle. This board basically functions as a Data Acquisition System (DAQ); labelled as PC/DAQ in Figure 3.1. This board allows the PC to connect to the actuators and encoders via USB, enabling the PC to operate the brake and throttle actuators, providing full control of the vehicle operation.

The brake and throttle actuators have been mechanically coupled to the brake and throttle assemblies on the vehicle, respectively. Two linear actuators are used to actuate the left and right brakes of the vehicle, and a single high torque servo motor actuates the throttle assembly on the vehicle's engine.

The vehicle is teleoperated through a Bluetooth joystick that has been paired with a Bluetooth device on the PC, allowing wireless drive-by-wire operation of the ARGO.

The PC on-board the vehicle communicates with all of the hardware on the vehicle and is used to run the state estimation algorithm proposed in this thesis.

The assumptions made while developing the kinematic model for the ARGO are as follows: 1) the ground contact patch between the wheels and the ground is taken to be a single point located at the center of the ground contact patch, 2) the vehicle's six wheels always stay in contact with the ground, 3) the wheels on either left or right side of the vehicle drive at the same

speed, 4) the radius of all wheels is the same and constant, 5) the vehicle has a fixed mass or payload.

In Figure 3.2, 1,..,6 are labels for the six wheels on the vehicle, x_B and y_B are the axes of the vehicle in the body frame at the center of gravity of the vehicle (CG), $\dot{\psi}$ is the yaw-rate, and X_I and Y_I are the axes of the inertial reference frame, and Z_I (not shown) is in the upwards direction. In this thesis, the excitations experienced by the vehicle in the Z_I axes are insignificant; however, it will be important in future work with the ARGO UGV. For simplicity, the orientation of the IMU mounted at the CG is the orientation used for the vehicle in the body frame.

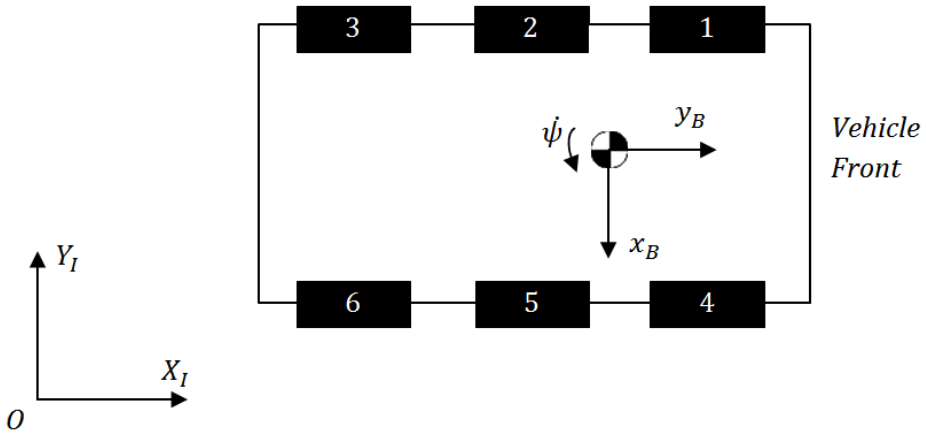

Figure 3.2. A top-view schematic of the skid-steered ARGO vehicle showing the body and inertial reference frames.

3.2. Longitudinal Slip

In this Section, a review of the braking mechanism of the ARGO is needed because it is important in establishing the procedure needed to calculate slip.

One of the most interesting aspects of this vehicle is its ability to take turns at a very small turn-radius. A differential connected to the vehicle's gearbox splits the torques from the gearbox to the two driving shafts of the differential [43]. The driving shafts of the differential are attached to a left and right brake disc which are connected to the wheels [43]. Engaging the left or right brake causes the brake callipers to engage the brake discs and initiate a left or right turn, respectively. For example, in making a sharp left turn, the driver would engage the left brakes of the vehicle. Which would lock all of the wheels on the left side of the vehicle (wheels 1, 2, 3 in

Figure 3.2), and all of the power from the engine would be transferred to the wheels on the right side of the vehicle (wheels 4, 5, 6 in Figure 3.2).

Longitudinal slip is a phenomenon that affects a pneumatic tire when a driving torque is applied to it [32]. When this driving torque is applied, a tractive force is developed at the ground contact patch of the tire [32]. Consequently, the tire treads in front of and within the ground contact patch are subjected to compression, and the side-wall of the tire undergoes shear deformation [32]. Due to these momentary physical changes, the distance travelled by a tire is reduced while its subject to a driving torque [32]. Therefore, the longitudinal slip is a percentage value that compares the linear speed of a tire against its angular velocity. For example, in an icy situation, a tire may experience 100% slip if it is rotating at high angular velocity, yet the linear speed of the tire is zero [32].

Prior to developing the longitudinal slip expression for the ARGO, the CG of the vehicle must be calculated. The CG in the $x_B - y_B$ plane is calculated by first placing the ARGO on four wheel scales and acquiring the weight distribution of the vehicle at its four corners (wheels 1, 3, 4 and 6 in Figure 3.2). The weight distribution of the vehicle is acquired based on the case where two passengers with a cumulative weight of 330 lbs, are seated inside of the vehicle. With the added weight of the two passengers, the vehicle is easier to steer.

After measuring the placement of the wheel scales, the center of mass formula is used to locate the CG. The CG in the z_B direction was provided by the ARGO manufacturer.

The location of the CG is found to be on the centreline of the vehicle (width-wise) and near the wheel-center for wheels 4 and 5 as shown in Figures 3.2 and 3.3.

The velocity of the vehicle's CG in the inertial frame is denoted as V_{cg_I}. The orientation-rates of the vehicle are represented by the rate of change of the three Euler angles: yaw-rate ($\dot{\psi}$), pitch-rate ($\dot{\theta}$) and roll-rate ($\dot{\phi}$).

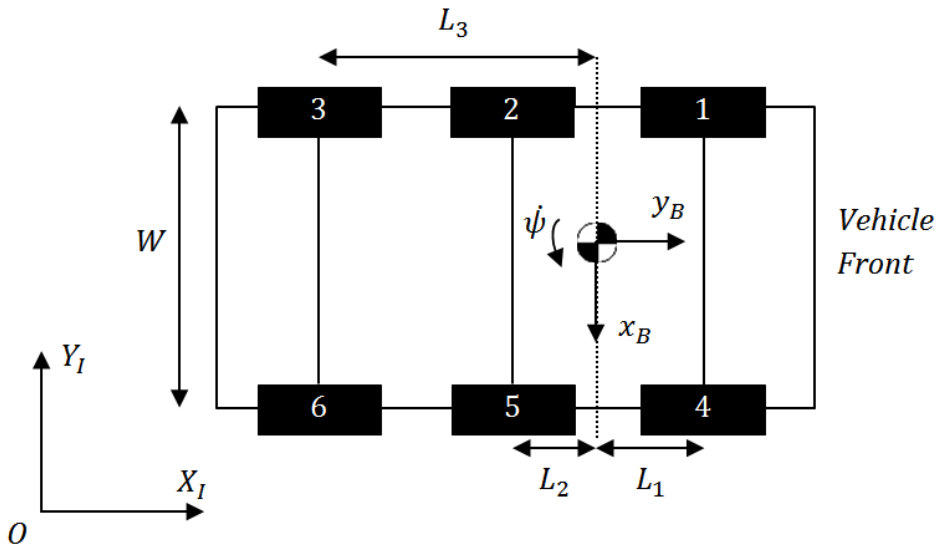

Figure 3.3. A schematic of the distances from the center of gravity of the vehicle to the contact points of the various wheels.

L_1, L_2, L_3 in Figure 3.3 are the distances from the wheel centers to the CG for wheels 1 or 4, 2 or 5, and 3 or 6, respectively, and W represents the distance between parallel wheel contact points.

Given the vehicle's velocity at its CG in the inertial frame (V_{cg_I}) and its yaw-rate ($\dot{\psi}$), the velocity of the wheel centers for all left and right side wheels in the X_I direction can be calculated as follows:

$$V_{left_X} = V_{cg_{X_I}} - \frac{W}{2} \cdot \dot{\psi} \tag{6}$$

$$V_{right_X} = V_{cg_{X_I}} + \frac{W}{2} \cdot \dot{\psi} \tag{7}$$

Similarly, the velocity of the wheel centers for a given wheel in the Y_I direction is also calculated in (8)-(10). Wheels 1/4, 2/5, and 3/6 share the same wheel centers; so calculated velocities for these wheels have been combined:

$$V_{Y_{1,4}} = V_{cg_{Y_I}} + L_1 \cdot \dot{\psi} \tag{8}$$

$$V_{Y_{2,5}} = V_{cg_{Y_I}} - L_2 \cdot \dot{\psi} \tag{9}$$

$$V_{Y_{3,6}} = V_{cg_{Y_I}} - L_3 \cdot \dot{\psi} \tag{10}$$

The resultant velocities of the wheel centers in the inertial frame are denoted as V_i, $(i = 1,..,6)$. Using (6)-(10), the velocities in the X_I and Y_I directions are known, and the resultant wheel center velocity can then be found as

$$V_i = \sqrt{{V_{X_i}}^2 + {V_{Y_i}}^2} \tag{11}$$

Since the wheels on the left-side of the vehicle rotate at the same angular velocity and the wheels on the right-side at the same velocity as well (assumption 3 in Section 1), the longitudinal slip can be calculated as follows:

$$\alpha_i = \frac{\omega_i \cdot r_{wheel} - V_i}{\omega_i \cdot r_{wheel}} \tag{12}$$

where the wheel angular velocities denoted as ω_i, $(i = 1,..,6)$, are obtained directly from the wheel encoders, α_i $(i = 1,..,6)$ represents the longitudinal slip of every wheel on the vehicle, and r_{wheel} represents the radius of a free-rolling wheel.

Since the slip value is a percentage ranging from 0-100%; positive or negative depending on tractive or braking effort, respectively, the calculated slip must be constrained as [45]:

$$\alpha_i \in [-1,1] \tag{13}$$

Equation (13) is critical in ensuring the slip calculations do not produce abnormal values due to erroneous IMU or encoder data.

3.3. Kinematic Model

The kinematic model of the ARGO is used for estimating the vehicle states that cannot directly be measured by the on-board sensors.

The vehicle orientation defined by its yaw (ψ), pitch (θ), and roll (ϕ) Euler angles are estimated using the angular velocities obtained from the IMU. The angular velocity measurements are represented in the body frame. The accelerations of the vehicle are also expressed in the body frame. On the other hand, the velocity and position data retrieved from the encoders and GPS are both expressed in the inertial frame.

In order to describe the vehicle kinematics in the inertial frame, a relationship that transforms the vehicles orientation from the body frame to the orientation in the inertial frame must be developed. To do this, a rotation about z_B is necessary, then about y_B and finally, about x_B. An additional rotation of $\pi/2$ about Z_I is also necessary to produce a resultant rotation matrix that orients the vehicle correctly in the inertial frame. This rotation matrix is represented as

$$R_B^I = \begin{bmatrix} c\theta c\psi & -c\phi s\psi + s\phi s\theta c\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\theta s\psi & -s\phi c\psi + c\phi s\theta s\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \cdot K \tag{14}$$

where K is a placeholder variable described as

$$K = \begin{bmatrix} c(\pi/2) & -s(\pi/2) & 0 \\ s(\pi/2) & c(\pi/2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{15}$$

where $c\psi$ refers to $\cos\psi$ and $s\psi$ refers to $sin\psi$. The position and velocity vectors in the inertial frame are

$$P_I = \left[P_{X_I}, P_{Y_I}, P_{Z_I}\right]^T \tag{16}$$

$$V_I = \left[V_{X_I}, V_{Y_I}, V_{Z_I}\right]^T \tag{17}$$

The IMU provides six inputs to the state estimator; the vehicles accelerations and angular velocities in the body frame:

$$A_B = \left[A_{x_B}, A_{y_B}, \left(A_{z_B} - G\right)\right]^T \tag{18}$$

$$\omega_B = \left[\omega_{x_B}, \omega_{y_B}, \omega_{z_B}\right]^T \tag{19}$$

It is worth nothing that the effects of gravity are directly accounted for in (18) by subtracting the acceleration in Z_I by the gravitational constant G. The accelerations and angular velocities retrieved from the IMU are used to calculate state estimates for the vehicles position, velocity, orientation, and orientation-rate. The state and input vectors to the state estimator are defined as follows:

$$x = \left[P_I, V_I, \psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}\right]^T \tag{20}$$

$$u = [A_B, \omega_B]^T \tag{21}$$

The rates of change for the vehicles motion can be described by the following equations [48]:

$$\dot{P}_I = V_I \tag{22}$$

$$\dot{V}_I = R_B^I(A_B) - G \tag{23}$$

$$\dot{\psi} = \frac{\omega_{x_B} \cdot sin\phi + \omega_{z_B} \cdot cos\phi}{cos\theta} \tag{24}$$

$$\dot{\theta} = \omega_{x_B} \cdot cos\phi - \omega_{z_B} \cdot sin\phi \tag{25}$$

$$\dot{\phi} = \omega_{y_B} + \left(\omega_{x_B} \cdot sin\phi + \omega_{z_B} \cdot cos\phi\right) tan\theta \tag{26}$$

where G is the gravitational constant parallel to the Z_I axes.

In the next chapter, a step-by-step walk-through of the hybrid EKF/KF architecture is presented.

Chapter 4.

State Estimation

In this chapter, details are presented on the state observer that is used in estimating the following twelve states (20): $P_I, V_I, \psi, \theta, \phi, \dot{\psi}, \dot{\theta}$ and $\dot{\phi}$. P_I and V_I represent the position and velocity of the vehicle in the inertial frame, respectively (see (16), (17), (22)), ψ, θ, ϕ are the Euler angles that define the ARGO's orientation, and $\dot{\psi}, \dot{\theta}, \dot{\phi}$ are the rates of change for the ARGO's orientation.

Section 1 presents the structure of the proposed estimator which is a hybrid EKF/KF, and the reasoning behind proposing such a structure. Sections 2 and 3 provide the details of the hybrid EKF/KF state estimator strategy and the governing equations.

4.1. Structure of the Proposed Estimator

The state estimation algorithm developed in this rotation uses a hybrid EKF/KF design approach to achieve accurate state estimates. The justifications for this design approach are as follows:

1) High end GPS sensors (military grade) are capable of providing precise position (centimeter accuracy) and velocity data to a state estimator for mobile robots or in automotive applications at a high cost ($20000-$40000 CAD) [18]. However, the use of a military-grade GPS on a smaller off-road vehicle like the ARGO is not feasible due to cost. On the other hand, a commercial-grade GPS cannot provide accurate enough position and velocity estimates especially in autonomous navigation where vehicle localization and SLAM application are required [18]. Hence, the approach taken in this thesis uses a commercial-grade GPS alongside other sensors (which is a more standard sensing approach for this category of off-road vehicles), for accurate state estimation for the ARGO. Accurate state estimation is a prerequisite to application of localization and mapping methods when the ARGO platform is expanded to enable autonomous driving.
2) The GPS sensor used in this work receives position data at a frequency of 0.5 Hz, while the proposed state estimator is designed for frequencies of 50 Hz (real-time). This means the position data received from the GPS are always two seconds old. Hence, if this two second old GPS data was integrated into the state estimator without processing the data to correct for this lag, the state output will fluctuate back and forth because it will attempt to fuse the late GPS position read-out with the current position of the vehicle (obtained from other sensors such as an IMU or wheel encoders). This effect is shown in Figure 4.1 with a state estimator that does not account for the GPS lag.

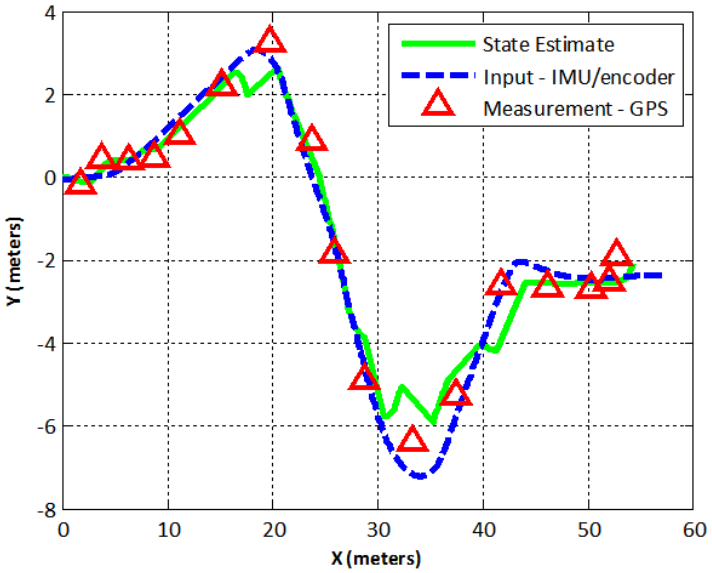

Figure 4.1. State estimate - IMU/encoder data fused with GPS data.

3) Figure 4.1 shows the state output when delayed GPS data is fused with IMU/encoder data. As the ARGO travels on a given trajectory, the IMU/encoders provide the most up-to-date position data for the vehicle. As soon as a GPS measurement is available, the state estimator attempts to fuse this measurement data with the IMU/encoder data, thus, the state outputs will fluctuate to account for the GPS data (as seen in Figure 4.1), this is because the GPS data that has just been received, is outdated and provides old position estimate data.

4) It is most desirable to design a state estimator that could fairly accurately (+/- two meters) track the position of an ARGO only using an IMU and wheel encoders. The reason that these two sensors (IMU/encoders) have been chosen is because they are easy to incorporate into the vehicle and are fairly simple to acquire data from. The goal in this rotation is to build a state estimator that is modular and separates the IMU/encoder estimation algorithm from the integration of the GPS sensor. The reason being, if the position outputs from the IMU/encoder estimator do not produce accurate enough results for a particular application, the modular GPS state estimator can be added to improve the accuracy of position state estimates.

5) The EKF algorithm in the proposed structure shown in Figure 4.2 has been designed for real-time/online applications but not the KF. So, if in a particular application the GPS sensor and the KF estimator are not used, then the EKF algorithm can be used online. Therefore, the proposed hybrid EKF/KF estimator is flexible and ready for both offline and online use. The hybrid EKF/KF architecture is depicted in Figure 4.2.

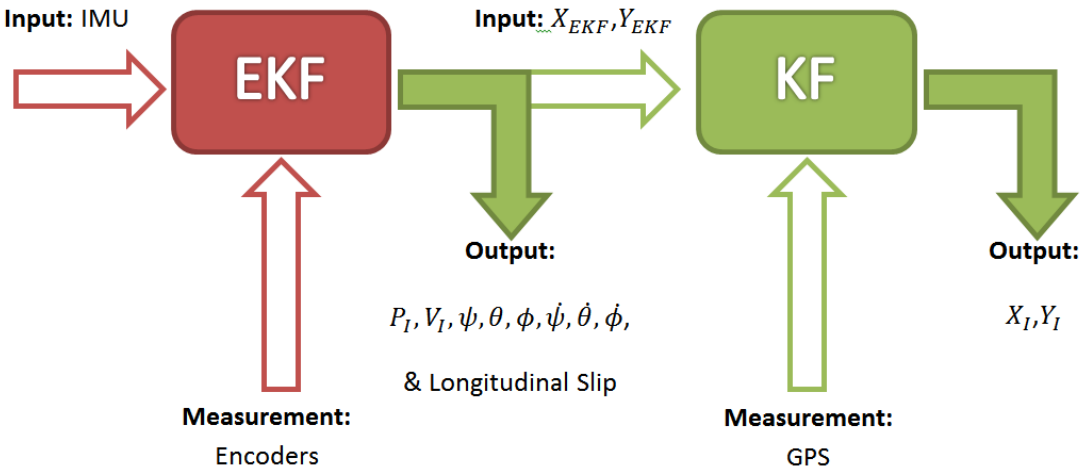

Figure 4.2. Structure of the proposed hybrid EKF/KF state estimator.

Figure 4.2 illustrates the use of the IMU and encoders as the input and measurements to the EKF state estimator, respectively. The resultant outputs from the EKF are state estimates of the position (P_I), velocity (V_I), orientation angles (ψ, θ, ϕ), rate of change of the orientation angles ($\dot{\psi}, \dot{\theta}, \dot{\phi}$), and the longitudinal slip of the ARGO. Figure 4.2 also shows that the position of the ARGO can further be corrected using the KF algorithm by taking the position estimates from the EKF (X_{EKF}, Y_{EKF}) as an input and the GPS sensor data as the measurement to produce a corrected state estimate of the vehicles position (X_I, Y_I).

As shown in (24)-(26), the kinematic model of the ARGO has nonlinear terms making it necessary to adopt a nonlinear observer such as an EKF as the initial state estimator. Correcting the position estimates of the EKF using the GPS sensor data is a linear mapping, consequently resulting in the use of a KF in the second stage of the estimation structure in (Figure 4.2).

4.2. Extended Kalman Filter

The state variables for the EKF are the position, velocity, orientation and orientation rates in the inertial frame, expressed as $P_I, V_I, \psi, \theta, \phi, \dot{\psi}, \dot{\theta}, \dot{\phi}$. The accelerations ($A_{x_B}, A_{y_B}, A_{z_B}$) and angular velocities ($\omega_{x_B}, \omega_{y_B}, \omega_{z_B}$) from the IMU are considered as the inputs to the EKF (i.e., (21)). Similarly, position ($X_I, , Y_I$) and velocity ($\dot{X}_I, \dot{Y}_I$) data from the wheel encoders are treated as the measurements to the KF. The generalized nonlinear state and measurement equations, respectively, are as follows [49]:

$$x_k = f(x_{k-1}; u_k; w_k) \tag{27}$$

$$z_k = h(x_k; v_k) \tag{28}$$

where x_k and z_k represent the state and measurement vectors at the time step k, $f(\cdots)$ and $h(\cdots)$ are the system and measurement functions, u_k is the input to the system, w_k and $v_k$are the system and measurement noise. The state and measurement estimates, in terms of their probability distributions are presented as [49]

$$\bar{x}_k = f(x_{k-1}, u_k, 0) \tag{29}$$

$$\bar{z}_k = h(x_k, 0) \tag{30}$$

Furthermore, the Gaussian system and measurement noise with zero mean are given as [49]

$$p(w_k) = N(0, Q_k) \tag{31}$$

$$p(v_k) = N(0, R_k) \tag{32}$$

4.2.1. System and input models

The state and input vectors as well as the prediction equations can be written, respectively, as

$$x_k = \left[X_{I_k}, Y_{I_k}, Z_{I_k}, \dot{X}_{I_k}, \dot{Y}_{I_k}, \dot{Z}_{I_k}, \psi_k, \theta_k, \phi_k, \dot{\psi}_k, \dot{\theta}_k, \dot{\phi}_k\right]^T \tag{33}$$

$$u_k = \left[A_{x_{B_k}}, A_{y_{B_k}}, A_{z_{B_k}}, \omega_{x_{B_k}}, \omega_{y_{B_k}}, \omega_{z_{B_k}}\right]^T \tag{34}$$

$$f(x_k) = \left[f_{X_{I_k}}, f_{Y_{I_k}}, f_{Z_{I_k}}, f_{\dot{X}_{I_k}}, f_{\dot{Y}_{I_k}}, f_{\dot{Z}_{I_k}}, f_{\psi_k}, f_{\theta_k}, f_{\phi_k}, f_{\dot{\psi}_k}, f_{\dot{\theta}_k}, f_{\dot{\phi}_k}\right]^T \tag{35}$$

The system (A_k) matrix is given as

$$A_k = \begin{bmatrix} I_3 & (I_3)\Delta T & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_3 & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & I_3 & (I_3)\Delta T \\ 0_{3\times3} & 0_{3\times3} & \beta & 0_{3\times3} \end{bmatrix} \tag{36}$$

$$\beta = \begin{bmatrix} 0 & a & b \\ 0 & 0 & c \\ 0 & d & e \end{bmatrix} \tag{37}$$

where ΔT is the loop rate of the EKF (50 Hz or 0.02 s), β, a, b, c, d, and e are place-holder variables expressed as

$$a = \left(\omega_{x_B} \sin\left(f_{\phi_{k-1}}\right) + \omega_{z_B} \cos\left(f_{\phi_{k-1}}\right)\right) sec(f_{\theta_{k-1}}) \cdot \tan(f_{\theta_{k-1}}) \tag{38}$$

$$b = \frac{\omega_{x_B} \cos\left(f_{\phi_{k-1}}\right)}{cos(f_{\theta_{k-1}})} - \frac{\omega_{z_B} \sin\left(f_{\phi_{k-1}}\right)}{cos(f_{\theta_{k-1}})} \tag{39}$$

$$c = -\omega_{x_B} \sin\left(f_{\phi_{k-1}}\right) - \omega_{z_B} \cos\left(f_{\phi_{k-1}}\right) \tag{40}$$

$$d = \left(\omega_{x_B} \sin\left(f_{\phi_{k-1}}\right) + \omega_{z_B} \cos\left(f_{\phi_{k-1}}\right)\right) sec^2(f_{\theta_{k-1}}) \tag{41}$$

$$e = \left(\omega_{x_B} \cos(f_{\phi_{k-1}}) - \omega_{z_B} \sin(f_{\phi_{k-1}})\right) tan(f_{\theta_{k-1}}) \tag{42}$$

Next, the input (B_k) matrix is given as

$$B_k = \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ (R_B^I)\Delta T & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & \xi \end{bmatrix} \tag{43}$$

$$\xi = \begin{bmatrix} f & 0 & g \\ h & 0 & i \\ j & 1 & k \end{bmatrix} \tag{44}$$

where ξ, f, g, h, i, j and k are place-holder variables, described as

$$f = \frac{\sin\left(f_{\phi_{k-1}}\right)}{\cos\left(f_{\theta_{k-1}}\right)} \tag{45}$$

$$g = \frac{\cos\left(f_{\phi_{k-1}}\right)}{\cos\left(f_{\theta_{k-1}}\right)} \tag{46}$$

$$h = \cos\left(f_{\phi_{k-1}}\right) \tag{47}$$

$$i = -\sin\left(f_{\phi_{k-1}}\right) \tag{48}$$

$$j = \sin\left(f_{\phi_{k-1}}\right) \cdot \tan\left(f_{\theta_{k-1}}\right) \tag{49}$$

$$k = \cos\left(f_{\phi_{k-1}}\right) \cdot \tan\left(f_{\theta_{k-1}}\right) \tag{50}$$

The yaw, pitch and roll rate prediction equations in (35) introduced in (24)-(26), are now expressed in a discretized form as

$$f_{\dot{\psi}_k} = \frac{\omega_{x_B}\sin(f_{\phi_{k-1}}) + \omega_{z_B}\cos(f_{\phi_{k-1}})}{\cos(f_{\theta_{k-1}})} \tag{51}$$

$$f_{\dot{\theta}_k} = \omega_{x_B}\cos(f_{\phi_{k-1}}) - \omega_{z_B}\sin(f_{\phi_{k-1}}) \tag{52}$$

$$f_{\dot{\phi}_k} = \omega_{y_B} + \left(\omega_{x_B}\sin(f_{\phi_{k-1}}) + \omega_{z_B}\cos(f_{\phi_{k-1}})\right)\tan(f_{\theta_{k-1}}) \tag{53}$$

where $f_{\dot{\psi}_k}$represents the predicted yaw-rate at the time instant k and $f_{\dot{\psi}_{k-1}}$ represents the predicted yaw-rate at the previous time step; the same notation applies for the pitch and roll rates. The discretized prediction equations for the yaw, pitch, and roll angles are

$$f_{\psi_k} = f_{\psi_{k-1}} + \Delta\mathrm{T} \cdot f_{\dot{\psi}_k} \tag{54}$$

$$f_{\theta_k} = f_{\theta_{k-1}} + \Delta T \cdot f_{\dot{\theta}_k} \tag{55}$$

$$f_{\phi_k} = f_{\phi_{k-1}} + \Delta T \cdot f_{\dot{\phi}_k} \tag{56}$$

where ΔT is the sampling time for the estimation algorithm (set at $0.02\ s$). Next, the discretized prediction equations for the velocities in the inertial frame are given by

$$f_{\dot{X}_I k} = f_{\dot{X}_I k-1} + \Delta \mathrm{T} \cdot R^I_{B\,1,1..3} \cdot A_{B\,1..3,1} \tag{57}$$

$$f_{\dot{Y}_I k} = f_{\dot{Y}_I k-1} + \Delta \mathrm{T} \cdot R^I_{B\,2,1..3} \cdot A_{B\,1..3,1} \tag{58}$$

$$f_{\dot{Z}_I k} = f_{\dot{Z}_I k-1} + \Delta \mathrm{T} \cdot R^I_{B\,3,1..3} \cdot A_{B\,1..3,1} \tag{59}$$

where $f_{\dot{X}_I k}$ is the predicted velocity in the X_I axes at the time instant k and $f_{\dot{X}_I k-1}$represents the predicted velocity in the previous time step; the same notation applies for the velocities in the Y_I and Z_I axes. $R^I_{B\,1,1..3}$ is the first row of the rotation matrix in (14); the same notation applies for the second and third rows of the rotation matrix. Similarly, $A_{B\,1..3,1}$ is the 3×1 acceleration vector from (18).

Lastly, the discretized prediction equations for the position of the ARGO in the inertial frame are as follows:

$$f_{X_I k} = f_{X_I k-1} + \Delta \mathrm{T} \cdot f_{\dot{X}_I k} \tag{60}$$

$$f_{Y_I k} = f_{Y_I k-1} + \Delta \mathrm{T} \cdot f_{\dot{Y}_I k} \tag{61}$$

$$f_{Z_I k} = f_{Z_I k-1} + \Delta \mathrm{T} \cdot f_{\dot{Z}_I k} \tag{62}$$

where $f_{X_I k}$ and $f_{X_I k-1}$ are the predicted positions of the ARGO in the X_I axes at the current and previous time steps, respectively. The same notation applies for the positions in the Y_I and Z_I axes.

4.2.2. Measurement model

The measurements for the EKF are the position and velocity measurements from the wheel encoders in the X_I and Y_I axes. The position measurements are calculated as [24]

$$h_{X_{I_k}} = h_{X_{I_{k-1}}} + \Delta\mathrm{T} \cdot \left(\frac{\omega_l+\omega_r}{2}\right) r_{wheel} \cdot \cos\left(f_{\psi_k}\right) \tag{63}$$

$$h_{Y_{I_k}} = h_{Y_{I_{k-1}}} + \Delta\mathrm{T} \cdot \left(\frac{\omega_l+\omega_r}{2}\right) r_{wheel} \cdot \sin\left(f_{\psi_k}\right) \tag{64}$$

where $h_{X_{I_k}}$ is the measured position in the X_I axes at the time instant k, $h_{X_{I_{k-1}}}$represents the same position at the previous time step. The same notation applies for the measured position in the Y_I axes. ω_l and ω_r represent the measured angular velocities from the left and right wheel encoders, respectively. The measurement equations for the velocities in the X_I and Y_I axes are

$$h_{\dot{X}_{I_k}} = \frac{h_{X_{I_k}} - h_{X_{I_{k-1}}}}{\Delta\mathrm{T}} \tag{65}$$

$$h_{\dot{Y}_{I_k}} = \frac{h_{Y_{I_k}} - h_{Y_{I_{k-1}}}}{\Delta\mathrm{T}} \tag{66}$$

where $h_{\dot{X}_{I_k}}$ and $h_{\dot{Y}_{I_k}}$ represent the measured velocity at time step k in the X_I and Y_I axes, respectively. The measurement matrix (H_k) in an EKF, aids in comparing the prediction estimates against the measured data is

$$H_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{67}$$

4.2.3. Extended Kalman filter algorithm

The system model equations in the previous two sections can be re-written as a set of time update equations as part of the proposed EKF algorithm [49]. The state prediction ($\bar{x}_k^*$) and covariance prediction (P_k^*) steps of the algorithm are as follows:

$$\bar{x}_k^* = f(x_{k-1}, u_k, 0) \tag{68}$$

$$P_k^* = A_k \cdot P_{k-1} \cdot A_k^T + Q_k \tag{69}$$

where $\bar{x}_k^*$ is the predicted state estimate in terms of its probability distribution, P_k^* is the covariance estimate, x_{k-1} and P_{k-1} are the state and covariance estimates of the previous time step and Q_k is the system/input noise covariance adapted from [49] as

$$Q_k = system\_noise + B_k \cdot \text{input\_noise} \cdot B_k^T \tag{70}$$

where $system\_noise$ and $input\_noise$ are the covariance matrices that were found empirically as

$$system\text{\_noise} = 5 \cdot [I_{12}] \tag{71}$$

$$input\text{\_noise} = \begin{bmatrix} 0.5\ m/s^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5\ m/s^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5\ m/s^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1\ rad/s & 0 & 0 \\ 0 & 0 & 0 & 0 & 1\ rad/s & 0 \\ 0 & 0 & 0 & 0 & 0 & 1\ rad/s \end{bmatrix} \tag{72}$$

The Kalman gain (K_k) equation is represented as

$$K_k = P_k^* \cdot H_k^T (H_k \cdot P_k^* \cdot H_k^T + R_k)^{-1} \tag{73}$$

where R_k is the measurement noise covariance matrix found empirircally as

$$R_k = \begin{bmatrix} 0.1\ m & 0 & 0 & 0 \\ 0 & 0.1\ m & 0 & 0 \\ 0 & 0 & 0.1\ m/s & 0 \\ 0 & 0 & 0 & 0.1\ m/s \end{bmatrix} \tag{74}$$

The state ($\bar{x}_k$) and covariance (P_k) update equations are finally obtained as

$$\bar{x}_k = \bar{x}_k^* + K_k\big(z_k - h(\bar{x}_k^*, 0)\big) \tag{75}$$

$$P_k = (I - K_k \cdot H_k)P_k^*(I - K_k \cdot H_k)^T + K_k \cdot R_k \cdot K_k^T \quad (76)$$

The proposed EKF uses the Joseph form of the covariance update equation (76). This equation is a more computationally-stable form of the original covariance update equation [49].

4.3. Kalman Filter

The state variables for the KF are X_I and Y_I as shown in Figure 4.2. The position estimates from the EKF (X_{EKF}, Y_{EKF}) are treated as the inputs to the KF and the GPS position data (X_{GPS}, Y_{GPS}) are treated as the measurements in the KF design. A discretized version of the Kalman filter adapted from [2,49] is used in this work, and the generalized state and measurement equations are given as

$$x_k = A_k \cdot x_{k-1} + B_k \cdot u_k + \epsilon_k \quad (77)$$

$$z_k = C_k \cdot x_k + \delta_k \quad (78)$$

where x_k and z_k represent the state and measurement vectors at the time step k, x_{k-1} the state at the previous time step, A_k and B_k are the system and input matrices, u_k is the input to the system, ϵ_k and δ_k are the system and measurement noise, respectively. The state and measurement estimates, in terms of their probability distributions are taken to be

$$\bar{x}_k = A_k \cdot x_{k-1} + B_k \cdot u_k \quad (79)$$

$$\bar{z}_k = C_k \cdot x_k \quad (80)$$

The Gaussian system/input and measurement noises with zero mean for the KF are taken as

$$p(\epsilon_k) = N(0, Q_k) \quad (81)$$

$$p(\delta_k) = N(0, R_k) \quad (82)$$

4.3.1. System and input models

The state and input vectors are as follows:

$$x_k = \left[X_{I_k}, Y_{I_k}\right]^T \tag{83}$$

$$u_k = \left[X_{EKF_k}, Y_{EKF_k}\right]^T \tag{84}$$

where X_{EKF_k} and Y_{EKF_k} represent the output from the EKF in the X_I and Y_I axes, respectively. Since the KF simply fuses the position estimates obtained from the EKF with the GPS data, the system (A_k) and input (B_k) matrices are designed as

$$A_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{85}$$

$$B_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{86}$$

4.3.2. Measurement model

This section presents details on how raw latitude/longitude GPS measurements are converted into usable measurement data for the proposed KF algorithm.

The Equirectangular approximation from [50] is the approach taken to convert the latitude/longitude location of the ARGO into a location described in Cartesian coordinates. The Equirectangular approximation assumes the Earth is a sphere and ignores its ellipsoidal effects in order to convert latitude/longitude data to Cartesian data. The Equirectangular approximation is as follows:

$$x_{GPS_k} = \Delta long \cdot \cos(lat) \cdot (6.371 \times 10^6) \tag{87}$$

$$y_{GPS_k} = \Delta lat \cdot (6.371 \times 10^6) \tag{88}$$

where x_{GPS_k} and y_{GPS_k} are the GPS coordinates of the vehicle, $\Delta long$ and Δlat represent the change in longitude and latitude (radians), respectively, and 6.371×10^6 is the spherical radius of the Earth in meters.

Before these values can be used as a measurement in the KF, they must be multiplied by a rotation matrix to express them in the inertial frame – this is illustrated in Figure 4.3.

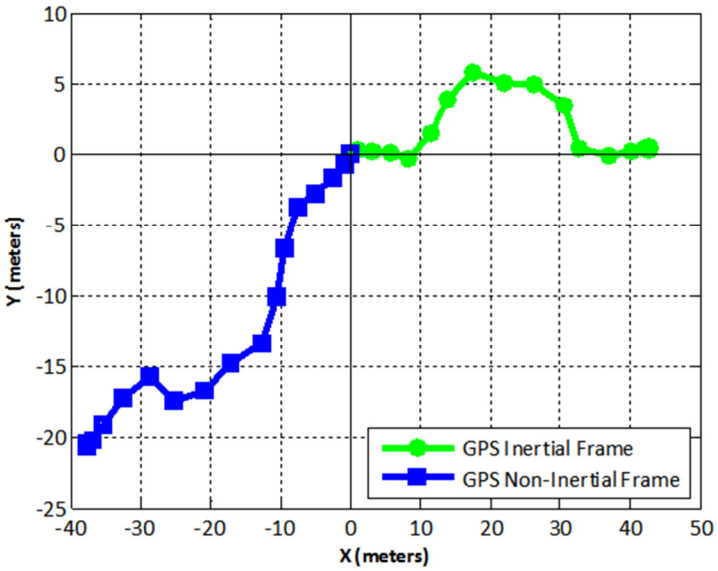

Figure 4.3. GPS data representation in the inertial frame.

The primary difference between GPS data in the inertial frame from data that is not is, the non-inertial data appears in quadrant three. Since the inertial frame in this work lies in quadrant one of the Cartesian coordinate plane, the non-inertial data must be adjusted. Therefore, the non-inertial GPS coordinates of (87) and (88) are adjusted as follows:

$$GPS_{I_k} = \begin{bmatrix} c(\psi_{init}) & -s(\psi_{init}) \\ s(\psi_{init}) & c(\psi_{init}) \end{bmatrix} \cdot \begin{bmatrix} x_{GPS_k} \\ y_{GPS_k} \end{bmatrix} \tag{89}$$

where $c(\psi_{init})$ and $s(\psi_{init})$ are short for $\cos\psi_{init}$ and $\sin\psi_{init}$, respectively, ψ_{init} is the angle obtained empirically by setting the straight-line components of the inertial frame GPS data in Figure 4.3 to coincide with the X_I axes. The resultant GPS_{I_k} vector is finally presented as

$$GPS_{I_k} = \left[X_{GPS_k}, Y_{GPS_k}\right]^T \tag{90}$$

where X_{GPS_k} and Y_{GPS_k} are the final rotated GPS coordinates in the X_I and Y_I axes. Furthermore, the measurement matrix (C_k) is simply taken as

$$C_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{91}$$

Equations (87)-(90) are used in conjunction with a virtual GPS point generation algorithm described in the next subsection, and (91) is used in the state update step of the KF algorithm.

4.3.2.1. Virtual GPS point generation algorithm

In this Section, the virtual GPS point generation algorithm is introduced. This algorithm is an efficient approach in resolving the delayed GPS data problem mentioned in Chapter 4, Section 1. This algorithm assumes that every two consecutive GPS points are connected by a straight line (as depicted in Figure 4.3). The equations for every straight line connecting two GPS data points are used to generate virtual GPS data points.

With this algorithm, the offline KF can be operated at a high loop-rate, allowing the position data from the EKF to be fused with available virtual GPS data. The basic form of the virtual GPS point generation algorithm is shown in Table 4.1 followed by a line by line explanation of the algorithm.

Table 4.1. Virtual GPS point generation algorithm.

	Virtual GPS Point Generation Algorithm $(X_{I_k}^*, X_{GPS}, Y_{GPS})$*:*
1.	$X_{current} = X_{I_k}^*$
2.	$if\ (X_{GPS_i} < X_{current} \leq X_{GPS_{i+1}})$
3.	$\{(i = i + 1)\}$
4.	$if\ (0 < X_{current} \leq X_{GPS_i})$
	$\{$
5.	$slope = (0 - Y_{GPS_i})/(0 - X_{GPS_i})$
6.	$Y_{current} = slope(X_{current} - X_{GPS_i}) + Y_{GPS_i}$
7.	$X_{GPS_{vir}} = X_{current}$
8.	$Y_{GPS_{vir}} = Y_{current}$
9.	$\}elseif\ (X_{GPS_{i-1}} < X_{current} \leq X_{GPS_i})$
	$\{$
10.	$slope = (Y_{GPS_{i-1}} - Y_{GPS_i})/(X_{GPS_{i-1}} - X_{GPS_i})$
11.	$Y_{current} = slope(X_{current} - X_{GPS_i}) + Y_{GPS_i}$
12.	$X_{GPS_{vir}} = X_{current}$
13.	$Y_{GPS_{vir}} = Y_{current}$
	$\}$
14.	$return\ X_{GPS_{vir}}, Y_{GPS_{vir}}$

There are three inputs to the algorithm shown in Table 4.1; $X_{I_k}^*$, X_{GPS} and Y_{GPS}. $X_{I_k}^*$ is the *x*-coordinate of the ARGO position estimated by the EKF; which is an input to the KF and is available after the state prediction step (95), X_{GPS} and Y_{GPS} are vectors containing the *x* and *y* coordinates of all GPS data received from the GPS sensor – it is assumed that this data is pre-processed before the KF is initialized in the hybrid EKF/KF state estimator.

In Lines 1 through 3 of the above algorithm, the predicted *x*-coordinate ($X_{I_k}^*$) is compared against the current GPS *x*-coordinate and the next GPS *x*-coordinate; if the comparison returns true, the current set of consecutive GPS points cannot be used to generate virtual GPS points because $X_{current}$ cannot be bound by this set of GPS points. So, the next set of GPS data points must be used. As such, i is incremented up by one.

Lines 4-8 in Table 4.1 represent the case where the first set of virtual GPS points are generated. Line 4 ensures that $X_{current}$ is a positive real value bounded by the inertial origin $(0, 0)$ and the first GPS data point. Line 5 calculates the slope of a virtual line that connects the origin to the first GPS data point. This slope is then used to generate the *y*-coordinate of a virtual GPS data point when the *x*-coordinate is taken as $X_{current}$ and the *x/y* coordinates of the GPS data point are taken as the second point on the line. Lines 7 and 8 create a set of virtual GPS data points in the *x* and *y* coordinates.

Lines 9-13 in Table 4.1 characterize the case where $X_{current}$ cannot be bound by the origin and the first GPS data point. As such, the bounds are changed so $X_{current}$ is now bound by the previous GPS data point and the current GPS data point in the *x* direction. The slope of a line that connects the two consecutive GPS data points is calculated in Line 10. Finally, Lines 11 through 13 exhibit the use of the slope to generate the *y*-coordinate of the next virtual GPS data point, which is then saved in the appropriate set of variables.

The algorithm presented in Table 4.1 assumes that $X_{current} \geq 0$, which means that the ARGO always faces forward (in the positive X_I axes). In an exceptional case where the ARGO makes a 180° turn and begins driving in the opposite direction, the above algorithm still applies; although, changes must be made to the bounds on the if-statements in Lines 4 and 9. These changes must reflect:

$$X_{GPS_i} < X_{GPS_{i-1}} \tag{92}$$

where the next GPS data point (X_{GPS_i}) is smaller than the previous data point ($X_{GPS_{i-1}}$) with respect to the X_I axes; whereas, the algorithm in Table 4.1 assumes the opposite to (92).

Additionally, the proposed algorithm can also be used to calculate $X_{current}$, by replacing $Y_{current}$ in lines 6 and 11; to account for the case when the vehicle moves only in the Y_I axes.

4.3.3. Kalman filter algorithm

The system, input and measurement model equations in Sections 3.1 and 3.2 of this chapter are now presented as a set of discretized time update equations as a part of the KF algorithm adapted from [2,49]. The state prediction ($\bar{x}_k^*$) and covariance prediction (P_k^*) steps of the KF algorithm are as follows:

$$\bar{x}_k^* = A_k \cdot x_{k-1} + B_k \cdot u_k \tag{93}$$

$$P_k^* = A_k \cdot P_{k-1} \cdot A_k^T + Q_k \tag{94}$$

where x_{k-1} and P_{k-1} are the state and covariance estimates of the previous time step and Q_k is the input noise covariance matrix obtained empirically as

$$Q_k = \begin{bmatrix} 0.31\,m & 0 \\ 0 & 0.31\,m \end{bmatrix} \tag{95}$$

The Kalman gain (K_k) equation is obtained as

$$K_k = P_k^* \cdot C_k^T (C_k \cdot P_k^* \cdot C_k^T + R_k)^{-1} \tag{96}$$

where R_k is the measurement noise covariance matrix described as

$$R_k = \begin{bmatrix} 1\,m & 0 \\ 0 & 1\,m \end{bmatrix} \tag{97}$$

The state ($\bar{x}_k$) and covariance (P_k) update equations are finally obtained as

$$\bar{x}_k = \bar{x}_k^* + K_k(z_k - C_t \cdot \bar{x}_k^*) \tag{98}$$

$$P_k = (I - K_k \cdot C_k) P_k^* (I - K_k \cdot C_k)^T + K_k \cdot R_k \cdot K_k^T \tag{99}$$

The more computationally stable Joseph form of the covariance update equation from [49] has been used once again.

This concludes the chapter on the hybrid EKF/KF architecture. In the next chapter, details on the experimental setup and findings are presented.

Chapter 5.

Experiments

Sections 1, 2 and 3 of this chapter present the discussion on the experimental setup and results for the double-lane change, slalom and fishhook experiments, respectively.

5.1. Experimental Setup

The loop-rate (frequency at which the state estimator updates at) for the hybrid EKF/KF algorithm is 50 Hz. The hybrid filter is used to estimate the ARGO vehicle's state (33), which consists of the ARGO's position, velocity, orientation, and orientation rates. The filter uses accelerations and angular velocities obtained from the IMU (34), position and velocity information from the wheel encoders (63)-(66) and latitude/longitude data from a GPS (87)-(90) in estimating vehicle states.

All of the experiments were performed on a flat, dry, asphalt track, arranged in the double-lane change, slalom and fishhook maneuvers. Figure 5.1 shows the ARGO during one of these maneuvers.

Figure 5.1. ARGO vehicle and test rig.

A set of markers were placed on the ground near the orange pylons for each experiment (Figure 5.1); these markers serve as the ground truth for the position estimates. During the experiments, the driver would maneuver the ARGO around the pylons and overtop the markers. Figures 5.2-5.4 show the general configuration of double-lane change, slalom, and fishhook maneuvers.

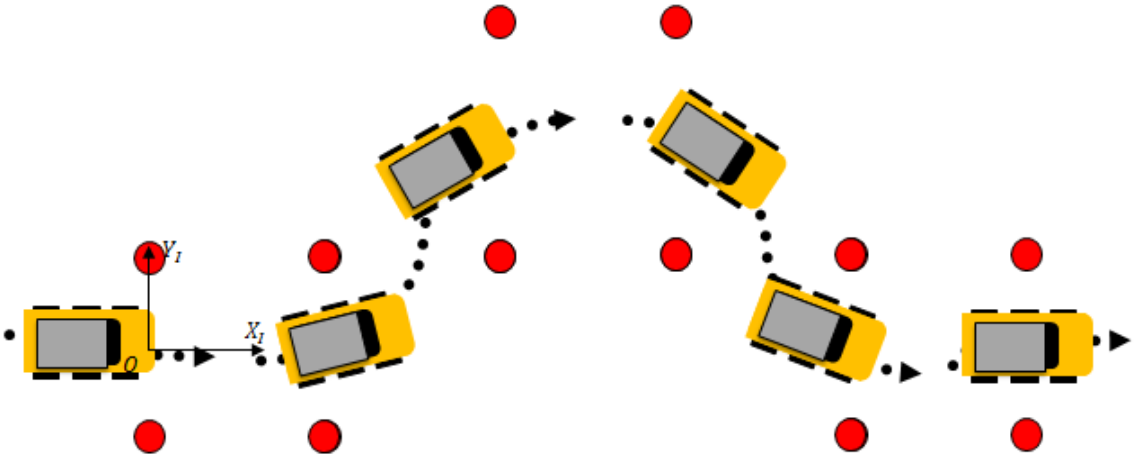

Figure 5.2. Schematic of double-lane change maneuver.

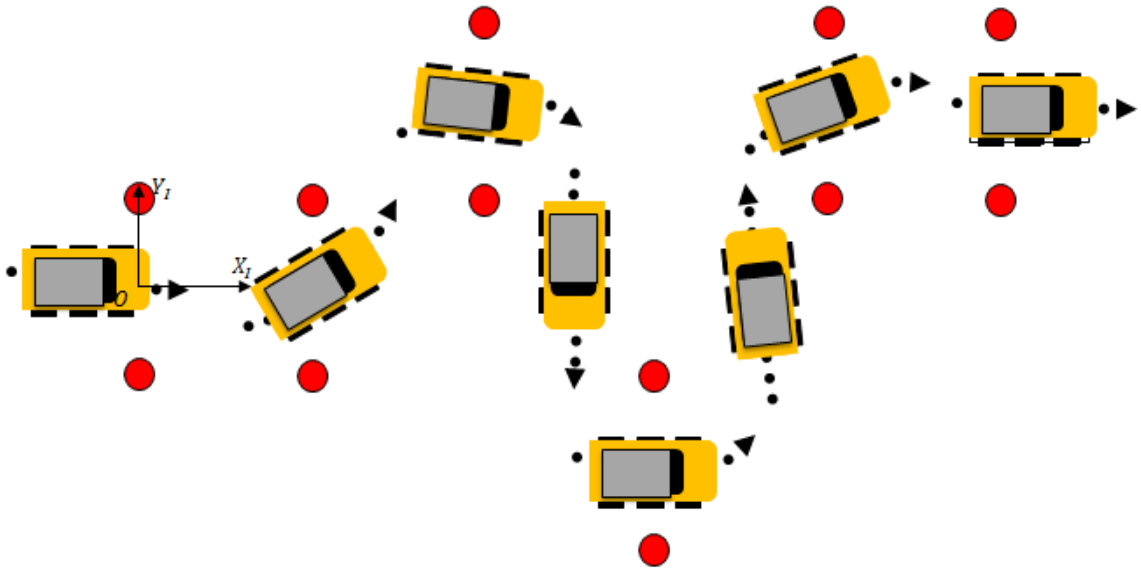

Figure 5.3. Schematic of slalom maneuver.

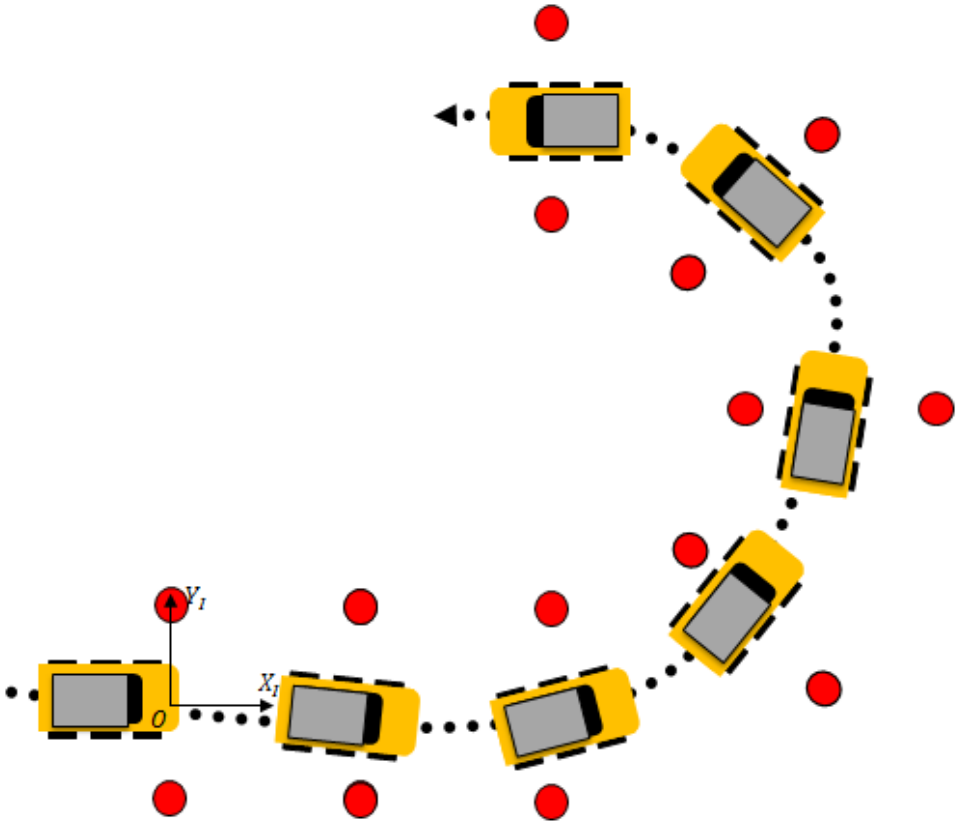

Figure 5.4. Schematic of fishhook maneuver.

The start location of each maneuver by the ARGO is indicated in Figures 5.2-5.4 as the inertial origin X_I and Y_I. The red dots in Figures 5.2-5.4 represent the guiding pylons (orange pylons in Figure 5.1) used in guiding the driver through the double-lane change, slalom, and fishhook maneuvers. Moreover, these maneuvers were chosen based on their ability to ensure the ARGO vehicle operates under standardized driving scenarios to facilitate the collection of accurate estimation data.

In the following sections, the estimated position, longitudinal slip, velocity, and orientation results for the three sets of experiments are presented.

5.2. Double-lane Change Experiment

This section details the performance of the proposed state observer from Figure 4.2, which is used to estimate the position, longitudinal slip, velocity and orientation of the ARGO during the double-lane change maneuver.

5.2.1. Position estimates

The generated position estimates by the EKF and the ground truth for the double-lane change maneuver are shown in Figure 5.5. The position results shown in Figure 5.5 are used as an input to the KF (see Figure 4.2), and are fused with GPS data in attempts to further improve the position estimates (Figure 5.7).

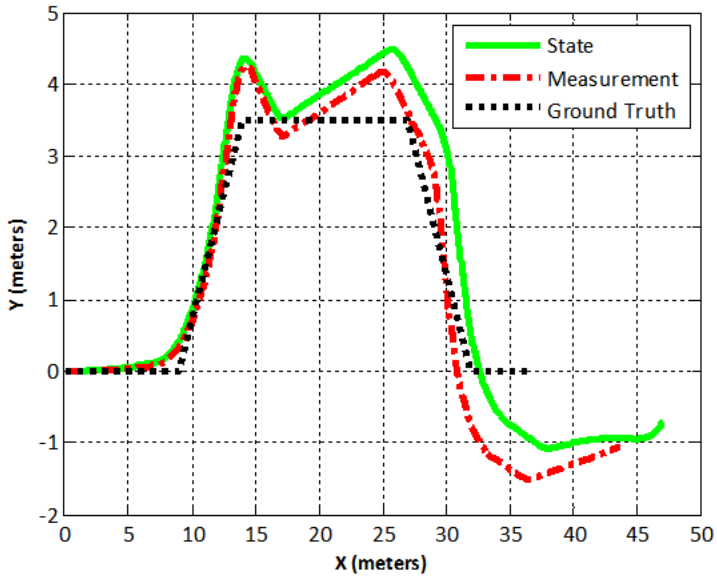

Figure 5.5. EKF position estimates – double-lane change experiment.

The state plot shown in Figure 5.5 is the EKF's estimate of the position of the ARGO based on the measurement data (obtained from encoders in (63)-(64)) and the prediction data (obtained from the IMU (60)-(62)); the prediction data is presented in Figure 5.6 separately due to the difference in scales and the large amount of error in the prediction data due to the double integration operation in (57)-(62). In state estimation, the prediction or input data plays the role of initiating the state estimation algorithm, even if the prediction estimates are erroneous, they can be adjusted by the measurement data.

The ground truth data in Figure 5.5 is the true/driven path of the ARGO; as explained in Section 1 of this chapter. This terminology will be used consistently throughout the course of this chapter.

Moreover, the state estimates in Figure 5.5 are combined with the measurement data and prediction data at the loop rate of 50 Hz, or every 0.02 seconds. The final output shown in Figure 5.5 is a result of iterating through Equations (68)-(76) thousands of times at instances of 0.02 seconds while the ARGO drives through the experimental maneuver.

The measurement data in Figure 5.5 shows that the ARGO does not follow the ground truth completely since the measurement and the ground truth data do not match perfectly. This difference in the measurement and ground truth data is due to the inherent error in the wheel

encoders. The wheel encoders have a built-in error, and the effect of this error impacts the position and velocity measurement data because the encoders, just like many sensors, are impacted by noise and vibration.

The state output data in Figure 5.5 is primarily influenced by the measurement data, and it is slightly influenced by the state prediction data shown in Figure 5.6. This has been done to ensure that if the measurement data becomes unavailable, the EKF can still produce state estimates for a short amount of time while the measurement data is unavailable. This feature has been added to increase the robustness of the EKF estimator for real-time performance. This design feature applies to all EKF position estimates.

The amount of influence the measurement and prediction data have on the state output can be adjusted through tuning the EKF system/input noise covariance matrices (70)-(72), and the measurement noise covariance matrices (74).

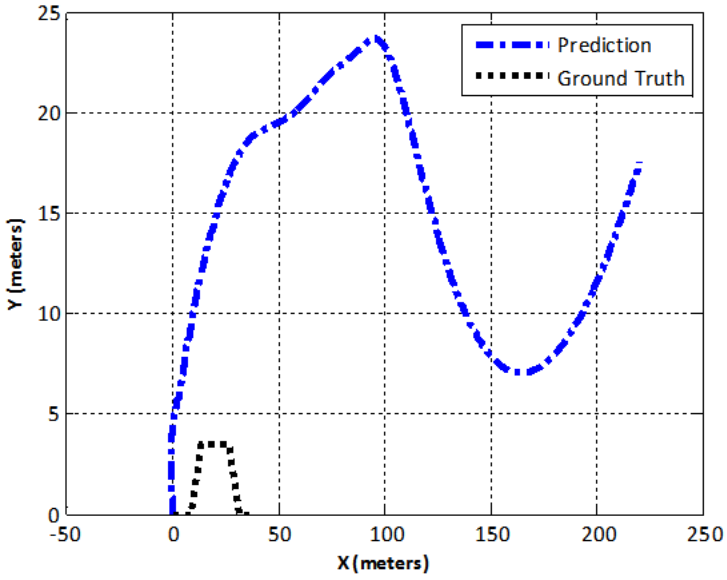

Figure 5.6. EKF state prediction – double-lane change experiment.

The state prediction data shown in Figure 5.6 are inaccurate when compared to the ground truth data because of the double integration procedure in (57)-(62). The EKF has been designed to give a slight amount of influence to the prediction data and a large amount of influence to the measurement data. This has been done to ensure that if the measurement data

becomes unavailable, the EKF can still produce state estimates for a short amount of time while the measurement data is unavailable.

The position estimates from the EKF are fused with GPS data using the proposed KF algorithm (Figure 4.2) to improve the accuracy of the position estimates; shown in Figure 5.7.

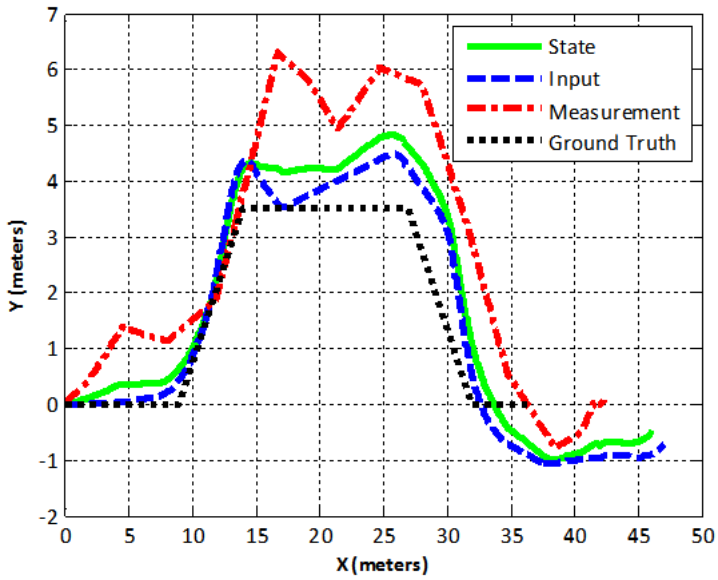

Figure 5.7. KF position estimates – double-lane change experiment.

The measurement plot in Figure 5.7 is obtained from the GPS, the input (or prediction) plot in this figure is the EKF position state estimates from Figure 5.5, the ground truth data is the same as that of Figures 5.5 and 5.6, and the state plot is the fusion of the measurement data and input data.

Since the measurement data is less accurate, when compared to the input data in Figure 5.7 (with respect to the ground truth), the measurement data is given a smaller influence when compared to that of the input/prediction data. This is because near the end of the experiment (when the vehicle has passed 35 meters in the horizontal direction in Figure 5.7), the measurement data is closer to the ground truth than the prediction data; however, this may not always be the case. As such, only a slight amount of influence was needed to draw the state estimates nearer to the ground truth.

The amount of influence of each data-set can be adjusted through the system/input noise covariance matrix for the KF in Equation (95) and the measurement covariance matrix in Equation (97).

5.2.1.1. Position error analysis

To quantify the error of the position estimates, the root-mean-square error (RMS) is used as a metric to evaluate the performance of the proposed filter. To do so, the position estimates output by each algorithm (EKF, KF) are compared to the ground truth measurements to calculate a value for each algorithm in the *X* and *Y* axes. The goal is to achieve position estimates that are within two meters accuracy.

This error analysis is performed for the position estimates produced by the proposed hybrid EKF/KF because there is ground truth data available. Table 5.1 provides the data points used in calculating the RMS for the EKF and KF position estimates.

Table 5.1. RMS data – double-lane change experiment.

Ground Truth		**EKF State Estimates**		**KF State Estimates**	
X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
0.0	0.0	0.0	0.0	0.0	0.0
4.0	0.0	4.0	0.0	4.0	0.3
6.5	0.0	6.5	0.1	6.5	0.4
9.0	0.0	8.2	0.2	8.6	0.5
14.0	3.5	13.9	4.3	13.9	4.2
20.5	3.5	20.5	3.9	20.5	4.2
27.0	3.5	27.2	4.2	27.2	4.6
32.0	0.0	32.3	0.2	32.7	0.5
34.5	0.0	34.5	-0.7	34.5	-0.3
36.5	0.0	36.7	-0.9	36.7	-0.8

The EKF and KF data shown in Table 5.1 were chosen based on the proximity of the state estimates to the ground truth data to achieve minimized RMS values. The RMS values have been summarized in Table 5.2.

Table 5.2. RMS values – double-lane change experiment.

RMS EKF-X (m)	RMS EKF-Y (m)	RMS KF-X (m)	RMS KF-Y (m)
0.30	0.52	0.27	0.60

The RMS values in Table 5.2 are generally the same for the EKF and KF and hence, both proposed algorithms are equally as reliable. Although, the KF algorithm has less error in the *X* axes when compared to the EKF algorithm, the EKF algorithm outputs less error in its *Y* axes estimates.

5.2.2. Longitudinal slip

The results that will be presented in this subsection are for the rear wheels only; since both rear wheels have wheel encoders that provide accurate wheel angular velocities. According to the third assumption made in Chapter 3-Section 1, the longitudinal slip for the other four wheels can also be calculated but at the cost of a reduction in accuracy. The longitudinal slip of the two rear wheels during the double-lane change experiment (calculated in Equations (6),(7),(10)-(13)) is shown in Figure 5.8.

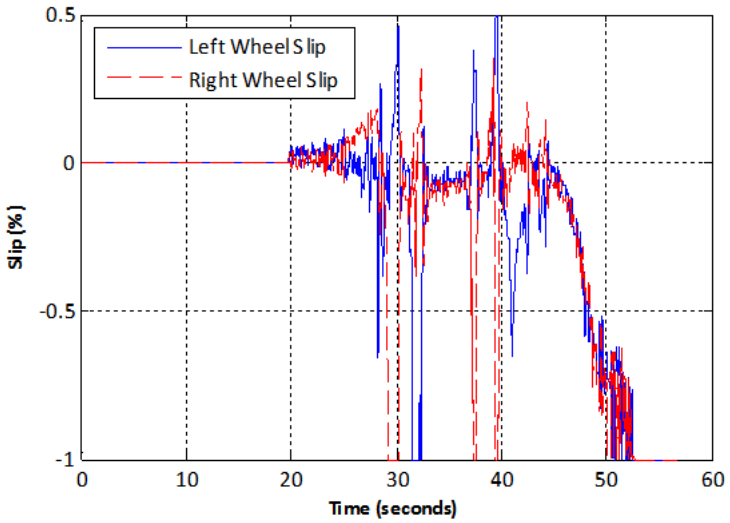

Figure 5.8. Longitudinal slip – double-lane change experiment.

The slip values in Figure 5.8 were calculated using the estimated state from the EKF (prediction data in Figure 5.7 and state data in Figure 5.5). At the beginning of the experiment in Figure 5.8, the ARGO is moving in a straight line from $time \approx 19 - 24\ s$. Both of the wheel slips during this time period hover around a value of zero. The straight-line path in this figure is represented in Figure 5.7 when $X \approx 0 - 8\ m$.

Next, the right wheel slip values begin to increase in the positive half-plane and the left wheel slip values show up slightly below zero in Figure 5.8 at $time \approx 24.5 - 28\ s$. These slip trends indicate that the vehicle is making a left turn; the right wheels are throttling forward (positive slip) and the left wheels are under braking (negative slip). This left turn is depicted in Figure 5.7 when $X \approx 8 - 13\ m$.

At $time \approx 29\ s$, and $time \approx 29.5 - 31\ s$, Figure 5.8 shows two spikes in the data; left wheel slip show positive spikes and right wheel negative spikes. During the time period between the two spikes, Figure 5.8 shows both wheels hovering around a slip value of zero. These trends indicate that the ARGO has made two right turns, the first a quick turn and the second, a longer more pronounced turn. These turns are visible in Figure 5.7 at $X \approx 13\ m$ and $Y \approx 4\ m$.

From Figure 5.8, it is evident that the right turn depicted in Figure 5.7 at $X \approx 13\ m$ and $Y \approx 4\ m$, was made possible by making two quick right turns. Additionally, in between these right turns, the ARGO rolled forward for a short amount of time as well. The remainder of the

slip values estimated in Figure 5.8 can be matched with the prediction estimates in Figure 5.7 in a similar manner.

It is worth noting that Figure 5.8 shows both slip values trend towards negative one at the end of the experiment ($time \approx 43 - 53\ s$). This happens because at the end of the experiment the driver of the ARGO applies the brakes to both left and right wheels at the same time to bring the vehicle to an immediate stop. This action of both brakes being engaged at the same time and the vehicle coming to a stop causes the slip values to trend towards negative one. These affects can also be seen in the longitudinal slip results for the slalom and fishhook experiments in the following sections.

5.2.3. Velocity estimates

The estimated velocity of the ARGO in the *X* and *Y* directions is shown in Figure 5.9.

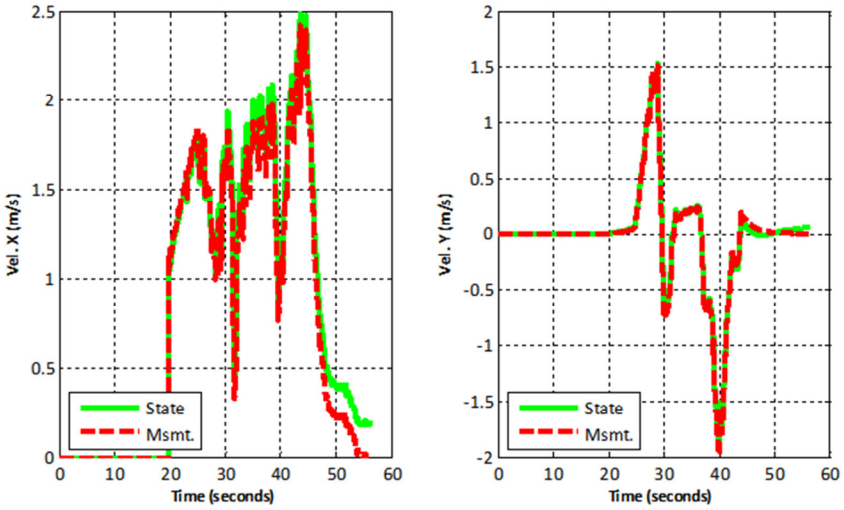

Figure 5.9. Estimated velocity in *X* and *Y* – double-lane change experiment.

The experimental hardware begins recording measurement data at $time = 20\ s$; this is when the state estimator begins generating state outputs and the measurement and state values change from a constant value of zero.

The state and measurement velocity data are similar, although there is a slight visible offset in both the *X* and *Y* plots in Figure 5.9. The reason for this is the velocity state estimates

have been designed to be primarily influenced by the measurement data, with a slight amount of influence from the prediction estimates in Figure 5.10.

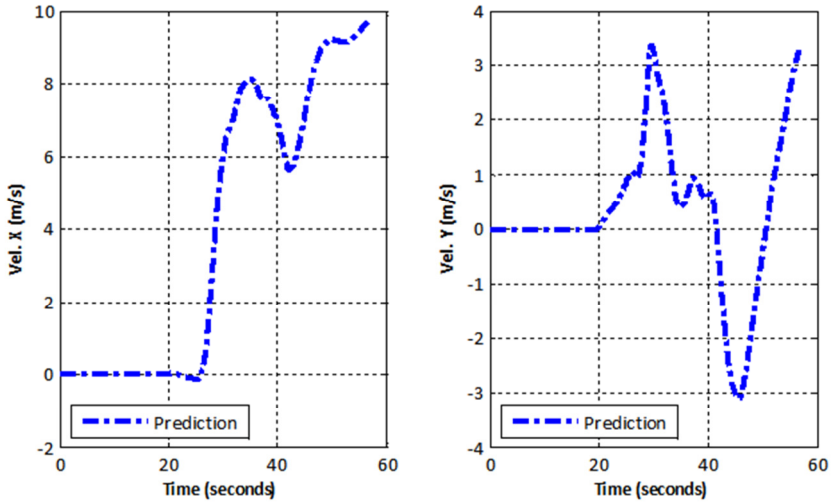

Figure 5.10. Predicted velocity in *X* and *Y* – double-lane change experiment.

The predicted velocity in *X* in Figure 5.10 is inaccurate due to an integration procedure in (57)-(59). The shape of the predicted velocity in *Y* observed in Figure 5.10 matches the general shape for the measured velocity in *Y* in Figure 5.9. The reason for this is the IMU was able to sense the accelerations in the *Y* axes well, since the vehicle experiences very pronounced accelerations in this direction since there is no suspension system on the vehicle to nullify these accelerations. In result, the state estimates are closer to the measurement data; unlike the deviations observed in Figure 5.9 for the velocity in *X* state estimates.

The velocity in *X* state estimates are offset from the measurement data in Figure 5.9 because the difference in shapes of the predicted velocities and the influence assigned to them. To attain improved state estimates in *X*, one can tune the covariance parameters to reduce the prediction influence.

5.2.4. Orientation estimates

The estimated yaw, pitch, roll angles depicted in Figure 5.11 are calculated using (54)-(56).

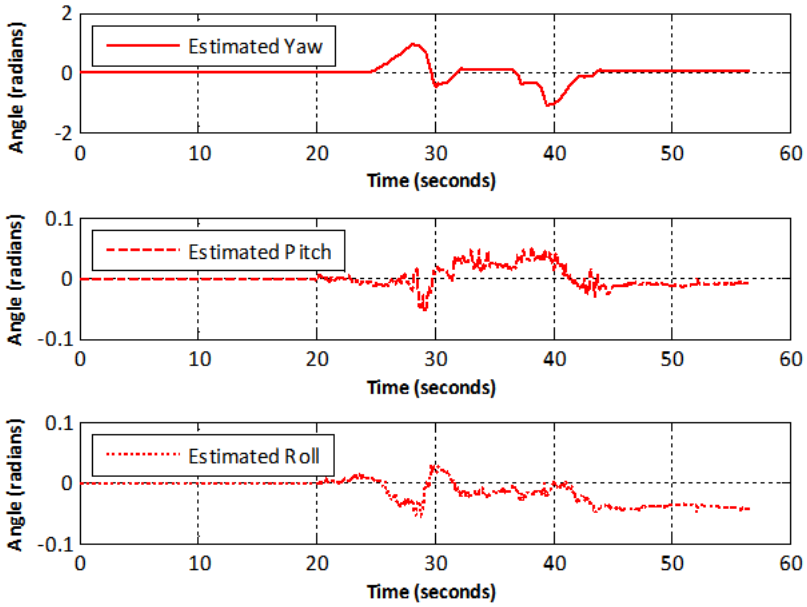

Figure 5.11. Yaw, pitch, roll estimates – double-lane change experiment.

It is worth noting that since the ARGO was driving on a flat surface, the pitch and roll angles generally do not change substantially, as shown in Figure 5.11. Any change seen in Figure 5.11 for this data is negligible. The trend for the yaw angle looks similar to the estimated velocity in *Y* in Figure 5.11. This similarity is due to the fact that the measurement equations use the most up-to-date estimate of yaw to calculate the next set of position measurement estimates in (63)-(66).

The ground truth and estimated yaw is calculated from the ground truth and EKF position data in Table 5.1, respectively. Using this data, the RMS value for the yaw estimates can be calculated as 0.13 rad, or 7.66 deg. Figure 5.12 shows a graph exhibiting the RMS data used to calculate the RMS value for yaw.

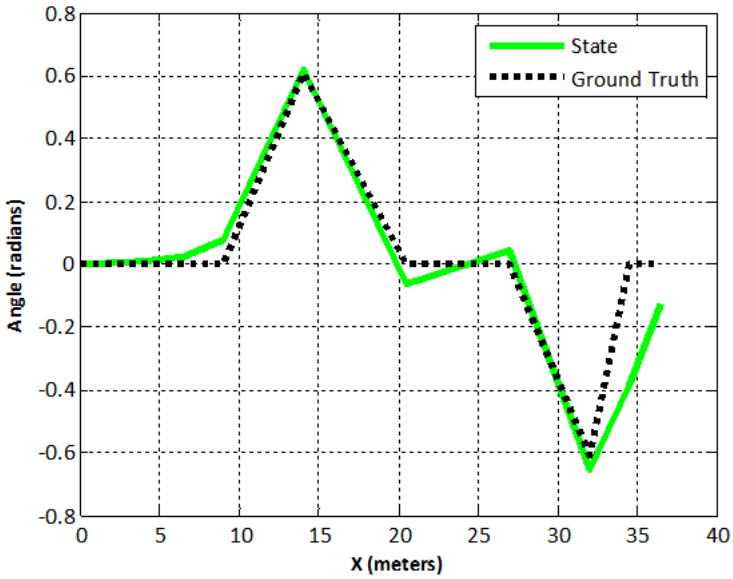

Figure 5.12. Yaw estimate comparison – double-lane change experiment.

Figure 5.12 shows that the RMS data of yaw angle estimates from the EKF are in close approximation to the ground truth values.

The rates of change of yaw, pitch and roll for the ARGO during the double-lane change experiment is presented in Figure 5.13.

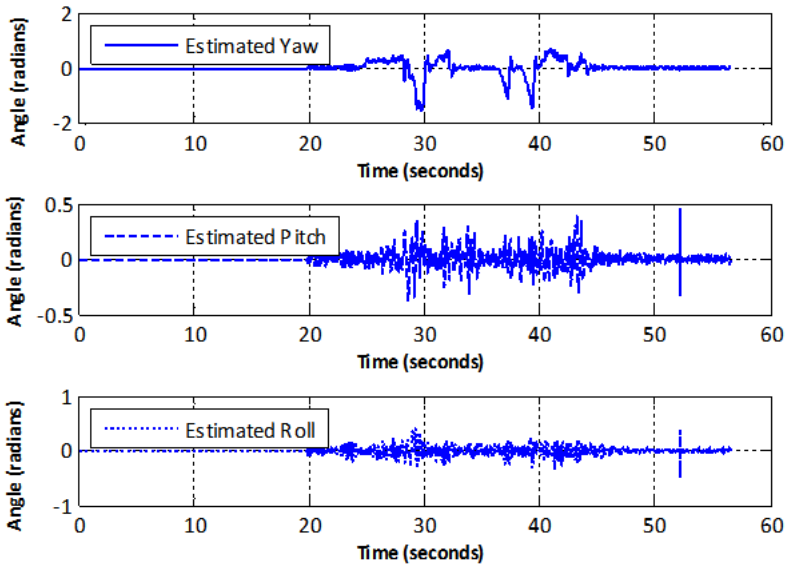

Figure 5.13. Estimated yaw, pitch and roll rates – double-lane change experiment.

The yaw, pitch and roll rate estimates depicted in Figure 5.13 were calculated in (51)-(53) using the angular velocities obtained from the IMU in the *X*, *Y* and *Z* axes.

The IMU located at the CG of the vehicle is mounted onto a bracket that is fixed to the chassis of the ARGO. While the ARGO is in motion, this bracket vibrates slightly in the Z direction. Due to this vibration, the pitch and roll rate estimates in Figure 5.13 exhibit noise in the data. However, this noise is consistent and small in magnitude, having little to no impact on the state estimates in the *X*-*Y* inertial plane.

5.3. Slalom Experiment

Similar to the double-lane change experiment, the position, longitudinal slip, velocity, and orientation estimates generated by the hybrid EKF/KF algorithm are now presented for the slalom experiment.

5.3.1. Position estimates

The position estimates generated by the proposed EKF algorithm for the slalom maneuver is exhibited in Figure 5.14.

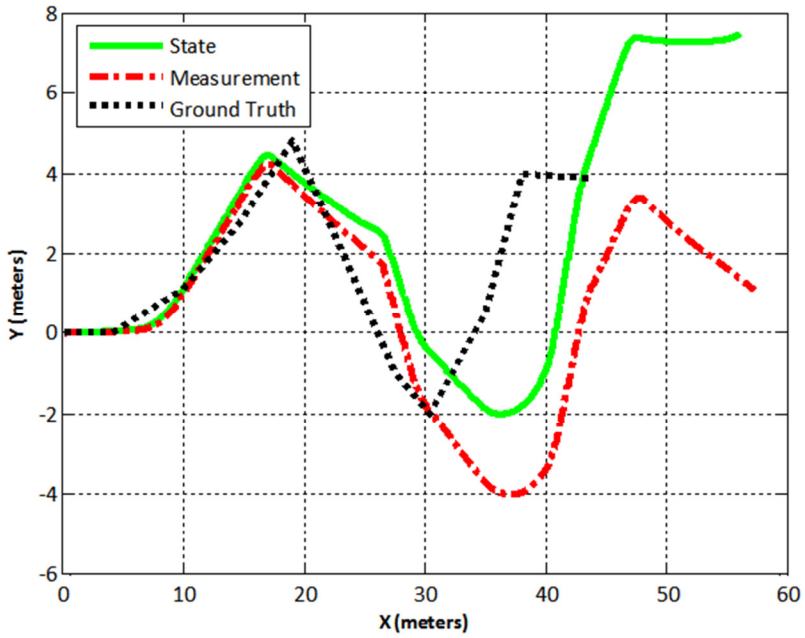

Figure 5.14. EKF position estimates – slalom experiment.

The state and measurement estimates in Figure 5.14 are considerably inaccurate after the midway point ($X \approx 28\,m$) in the slalom course. One reason why the inaccurate path is observed is due to driving error.

The driver of the ARGO missed making the sharp turn at $X = 30\,m$ during this experiment because the ARGO is a difficult vehicle to maneuver. Hence, the driver ended up taking a wider turn and negatively affected the path results for the remainder of the experiment. Figure 5.14 also shows the state output data moving in the positive Y direction near the end of the experiment. This happens because the EKF algorithm fuses the measurement data with the prediction estimate data in Figure 5.15.

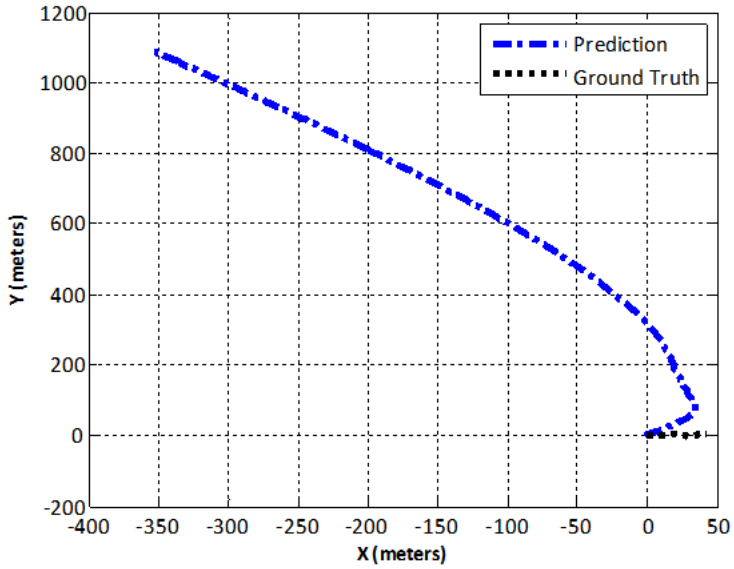

Figure 5.15. EKF prediction estimate – slalom experiment.

The prediction estimate in Figure 5.15 trends towards quadrant two in the Cartesian coordinate plane. The EKF fuses this data with the measurement data to produce an state estimate that moves towards the positive *Y* direction near the end of the slalom experiment as seen in Figure 5.14. Using the KF algorithm, the GPS data is fused with the state outputs, as depicted in Figure 5.16.

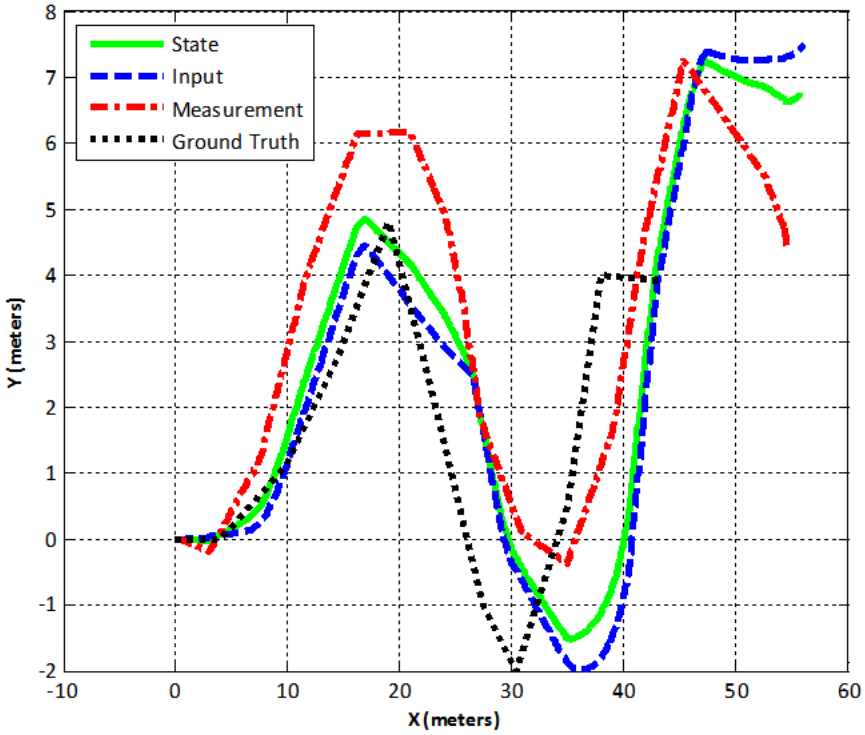

Figure 5.16. KF position estimates – slalom experiment.

The final state estimate from the KF shown in Figure 5.16 does not match the ground truth perfectly. Additionally, the measurement data does not follow the ground truth closely either.

The RMS values are now calculated for the different estimates to analyse the amount of error realised by the position estimates.

5.3.1.1. Position error analysis

Table 5.3 provides the data points used in calculating the RMS data for the EKF and KF position estimates.

Table 5.3. RMS data – slalom experiment.

Ground Truth		**EKF State Estimates**		**KF State Estimates**	
X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
0.0	0.0	0.0	0.0	0.0	0.0
4.0	0.0	4.0	0.0	4.0	0.1
9.8	1.1	9.8	1.1	9.7	1.4
14.5	2.7	14.3	3.6	14.1	3.7
19.0	4.8	18.9	4.0	18.9	4.5
23.2	2.0	23.6	3.0	23.8	3.5
27.5	-1.0	28.5	0.7	28.6	0.8
30.4	-2.0	30.6	-0.5	30.6	-0.4
35.0	0.5	40.7	-0.2	39.9	-0.1
38.0	4.0	43.1	3.9	42.7	3.8
43.5	3.9	47.0	7.3	47.2	7.2

The EKF and KF data shown in Table 5.3 were chosen based on their proximity to the ground truth data to achieve minimal RMS values. The RMS values have been detailed in Table 5.4.

Table 5.4. RMSD values – slalom experiment.

RMS EKF-X (m)	**RMS EKF-Y (m)**	**RMS KF-X (m)**	**RMS KF-Y (m)**
2.54	1.32	2.37	1.36

The RMS values for both the EKF and KF algorithms in Table 5.4 show that the *X* axes have an error greater than the target error of two meters. Although, the KF algorithm exhibits an error in the *X* axes that exceeds the target value by only fifteen percent. The *Y* axes error for both algorithms is less than one meter.

Considering the poor RMS values in Table 5.4, the slalom track must be changed and the configuration of the slalom track as presented in this thesis is not a good measure for testing the estimator using the ARGO. The configuration of the slalom experiment presented in this thesis works well for a vehicle that is more nimble than an ARGO, since the ARGO is difficult to maneuver due to its skid-steered driving scheme. In the future, the slalom experimental setup should be changed so the turns are wider and more manageable for the ARGO's driver.

5.3.2. Longitudinal slip

The longitudinal slip for the slalom experiment is shown in Figure 5.17.

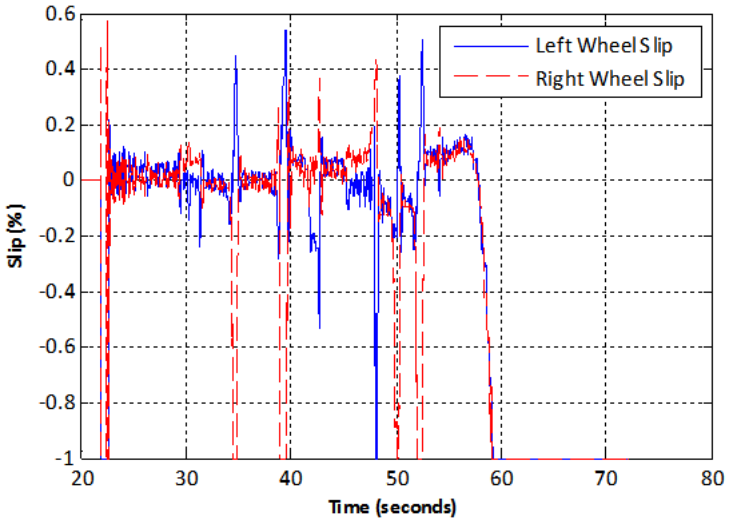

Figure 5.17. Longitudinal slip – slalom experiment.

The longitudinal slip estimates in Figure 5.17 begin with a lot of noise seen in the data at the beginning of the experiment. This occurs when the ARGO is lifted into place by the driver to position the vehicle correctly at the beginning of the experiment.

The wide left turn made at the midway point of the slalom experiment in Figure 5.16 is clearly represented in Figure 5.17 at $time \approx 40 - 47\ s$. During this time period, the right wheel slip stays consistently positive and the left wheel slip trends towards negative values; especially at $time = 48\ s$.

5.3.3. Velocity estimates

The estimated and predicted velocities of the ARGO in the X and Y directions for the slalom experiment are shown in Figures 5.18 and 5.16.

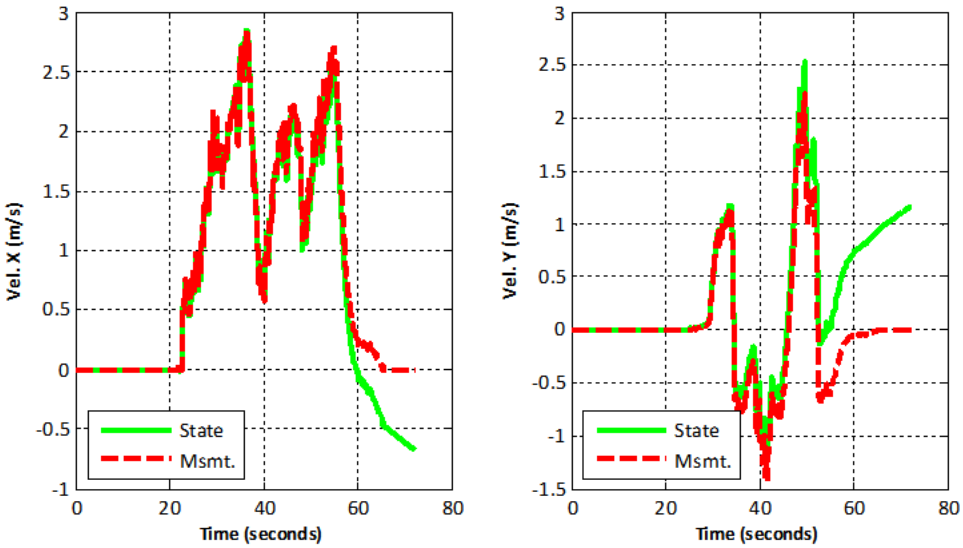

Figure 5.18. Estimated velocity in X and Y – slalom experiment.

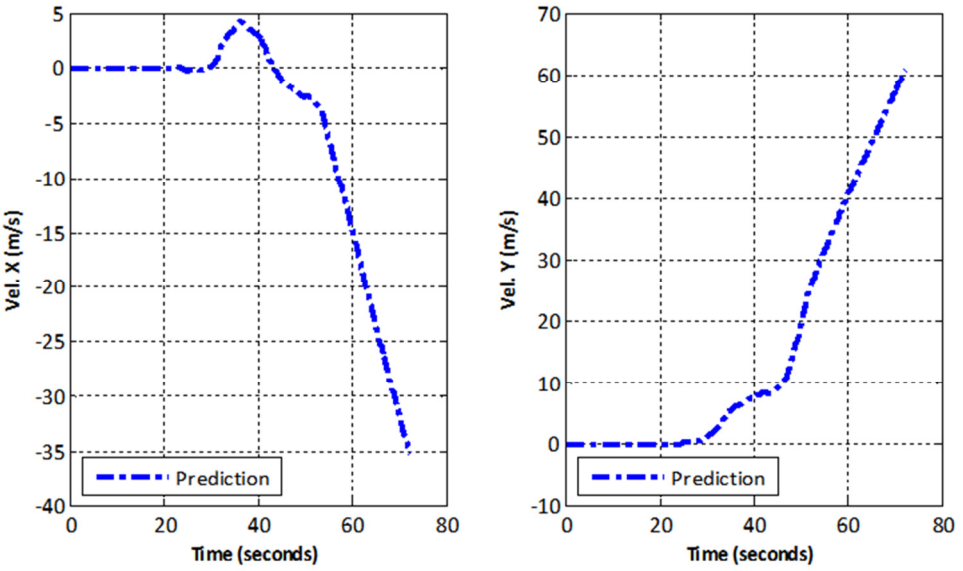

Figure 5.19. . Predicted velocity in X and Y – slalom experiment.

The measurement and state estimates for the velocities in Figure 5.18 are similar in trend until the end of the experiment, where the erroneous prediction data is fused with the measurement data. The prediction data in Figure 5.18 see the most integration error near the end of the experiment, thereby, negatively affecting the state results shown in Figures 5.18.

5.3.4. Orientation estimates

The estimated orientation angles during the slalom experiment are shown in Figure 5.20.

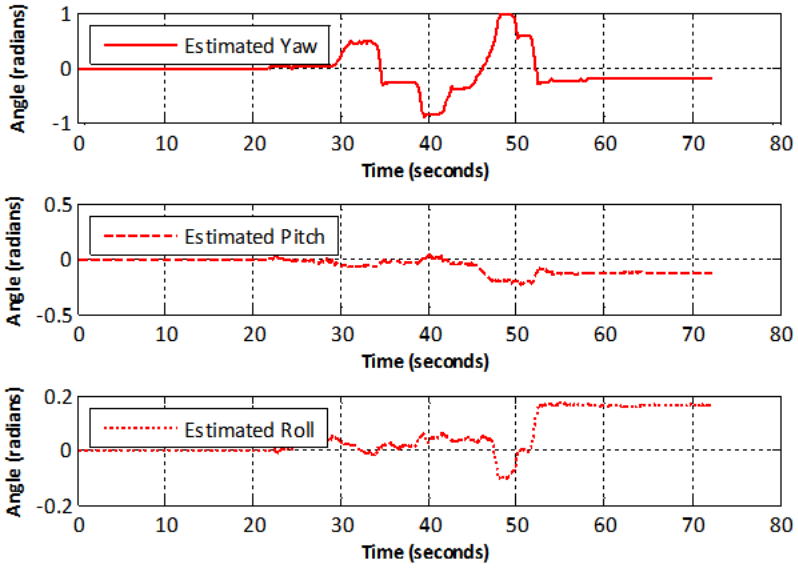

Figure 5.20. Yaw, pitch, roll estimates – slalom experiment.

It is worth noting that, near the end of the experiment in Figure 5.20 ($time > 45\ s$), the pitch and roll angles show a significant change in value. A significant change of pitch and roll angles occurs when the bracket the IMU is attached to on the vehicle is accidentally touched. When this happens, the orientation of the IMU changes slightly, as such, so does the pitch and roll orientation estimates.

Figure 5.21 compares the yaw estimate and yaw ground truth for the slalom experiment.

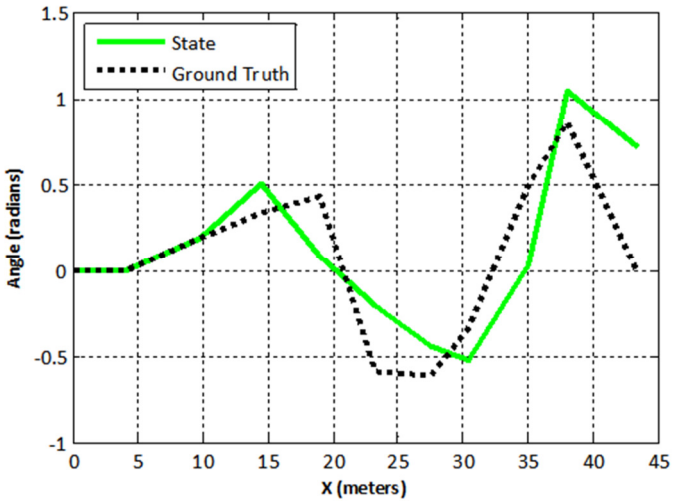

Figure 5.21. Yaw estimate comparison – slalom experiment.

It is evident from Figure 5.21 that the RMS data of yaw angle estimates from the EKF have a significant amount of deviation from the ground truth. The RMS value for the yaw estimate is 0.33 rad, or 19.39 deg.

The yaw, pitch and roll rate estimates are depicted in Figure 5.22.

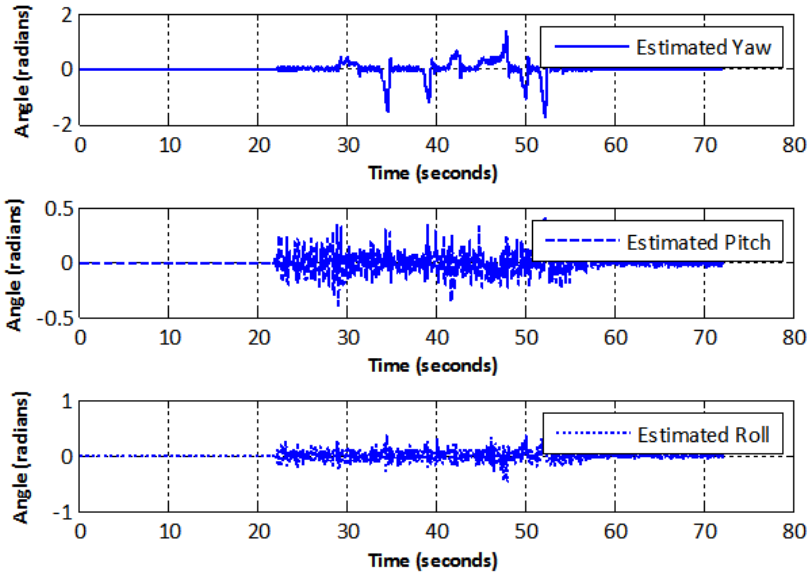

Figure 5.22. Estimated yaw, pitch and roll rates – slalom experiment.

The yaw, pitch and roll rate estimates in Figures 5.22 look similar to those shown in Figure 5.13 for the double-lane change experiment; where the yaw rate estimates are noise-free and vary significantly in magnitude, and the pitch and roll rate estimates exhibit noise and are small in magnitude.

5.4. Fishhook Experiment

In this section, the present the position, longitudinal slip, velocity, and orientation estimates generated by the hybrid EKF/KF algorithm for the fishhook experiment.

5.4.1. Position estimates

The state and prediction position estimates generated by the EKF for the fishhook experiment are shown in Figures 5.23 and 5.24.

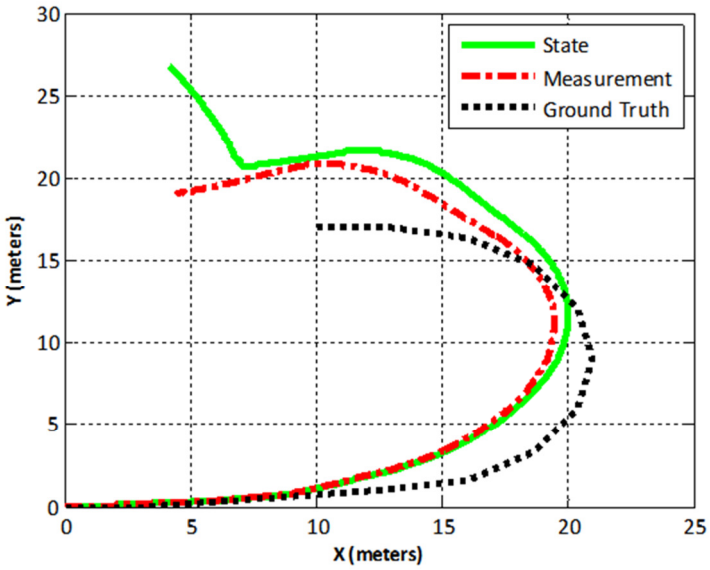

Figure 5.23. EKF position estimates – fishhook experiment.

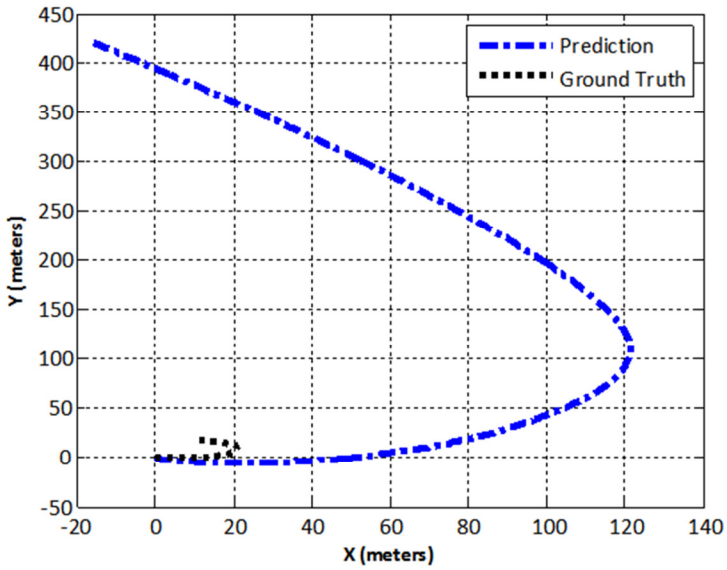

Figure 5.24. EKF prediction estimate – fishhook experiment.

The prediction estimates shown in Figure 5.24 cause the state estimates at the end of the fishhook maneuver in Figure 5.23 to trend in the positive *Y* direction. Additionally, both measurement and state estimates are consistently dissimilar from the ground truth path in both figures.

The KF state estimates for the fishhook experiment are shown in in Figure 5.25.

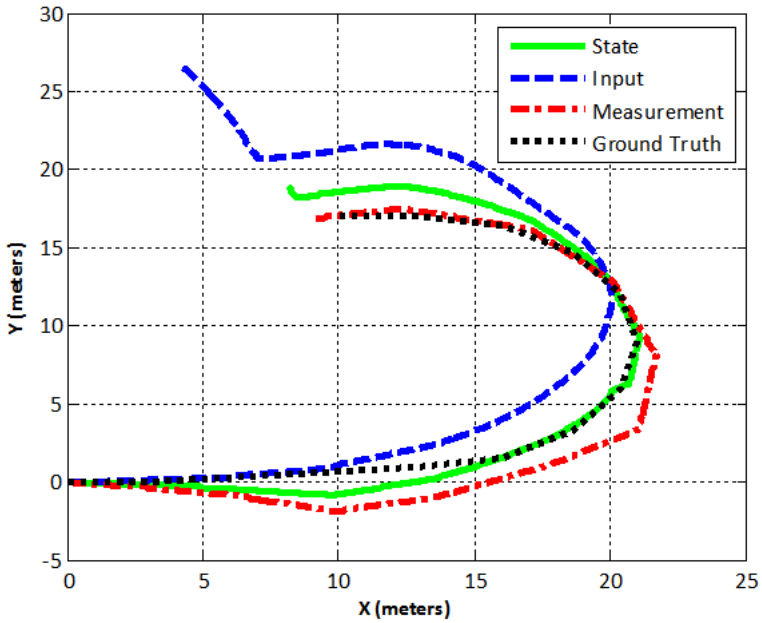

Figure 5.25. KF position estimates – fishhook experiment.

The state estimates in Figure 5.25 match the ground truth measurements well. Whereas, both measurement and prediction estimates do not. The RMS analysis in the next section will provide a quantitative means to compare the accuracy of the position results.

5.4.1.1. Position error analysis

The RMS data is provided in Table 5.5 and the RMS values for each algorithm are shown in Table 5.6.

Table 5.5. RMS data – fishhook experiment.

Ground Truth		**EKF State Estimates**		**KF State Estimates**	
X (m)	Y (m)	X (m)	Y (m)	X (m)	Y (m)
0.0	0.0	0.0	0.0	0.0	0.0
3.0	0.0	3.0	0.1	3.0	-0.2
13.0	1.0	12.9	2.2	13.2	0.1
16.1	1.6	15.8	3.8	16.1	1.6
18.7	3.3	18.5	6.7	18.6	3.6
20.4	5.9	19.4	8.5	20.4	6.1
21.0	9.0	20.0	10.7	21.1	9.0
20.4	12.1	19.7	14.0	20.3	11.9
18.7	14.7	18.8	15.9	18.7	14.9
16.1	16.4	16.2	19.0	16.1	17.4
13.0	17.0	12.9	21.5	12.9	18.8

Table 5.6. RMS values – fishhook experiment.

RMS EKF-X (m)	**RMS EKF-Y (m)**	**RMS KF-X (m)**	**RMS KF-Y (m)**
0.47	2.54	0.08	0.80

There are significant differences in the RMS values for the EKF and KF algorithms in Table 5.6. In fact, the RMS values for the EKF in *Y* do not meet the target error of two meters. On the other hand, the RMS values for the KF indicate errors less than one meter. Thus, the KF algorithm greatly improved the position estimates for the fishhook experiment, when compared to the EKF results.

5.4.2. Longitudinal slip

The longitudinal slip estimates for the fishhook experiment are shown in Figure 5.26.

Figure 5.26. Longitudinal slip – double-lane change experiment.

It is observed in Figure 5.26 that the slip for the right wheel trends to higher positive values as the ARGO drives through the fishhook course, whereas, the left wheel estimates trend toward increasingly negative values. At $time \approx 40 - 42\ s$, the ARGO goes in a straight line; this was done by the driver to keep the vehicle on the ground truth path.

Additionally, a sharp right turn is observed in Figure 5.26 at $time \approx 46\ s$. This is visible in Figure 5.25 at $X \approx 8\ m$ and $Y \approx 18\ m$.

5.4.3. Velocity estimates

The estimated and predicted velocities of the ARGO in the X and Y directions for the fishhook experiments are shown in Figures 5.27 and 5.28.

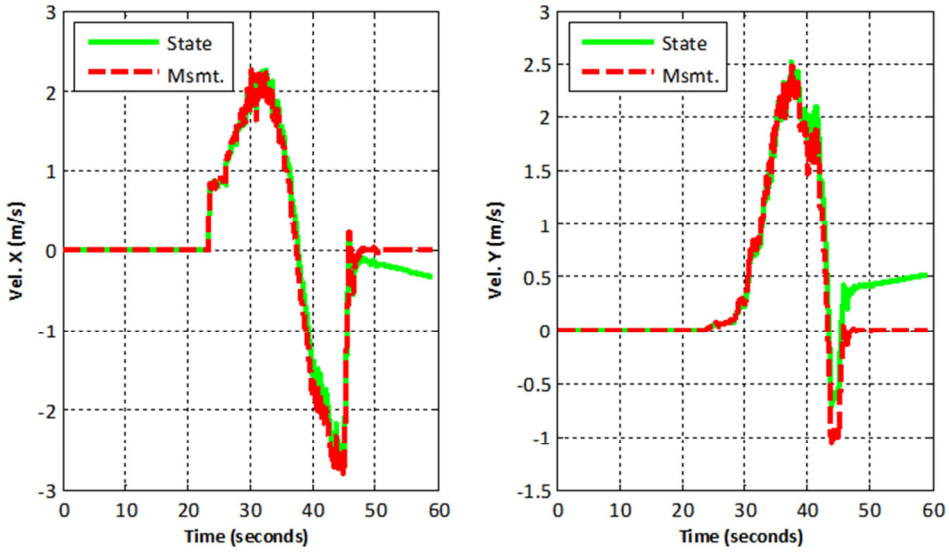

Figure 5.27. Estimated velocity in *X* and *Y* – fishhook experiment.

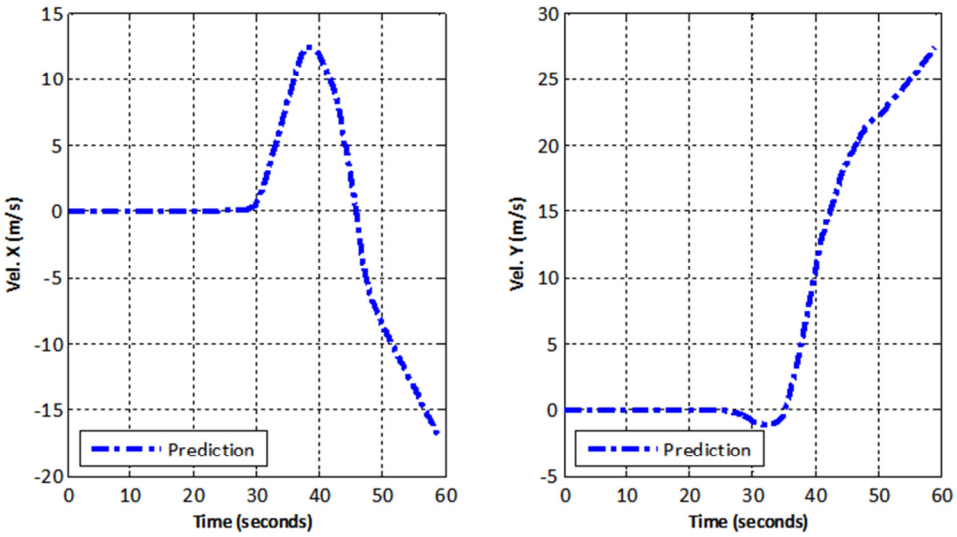

Figure 5.28. Predicted velocity in *X* and *Y* – slalom experiment.

The measurement and state estimates in Figure 5.27 have similar trends to what was observed in the double-lane change and slalom experiments.

5.4.4. Orientation estimates

The estimated orientation angles during the fishhook experiment are shown in Figure 5.29.

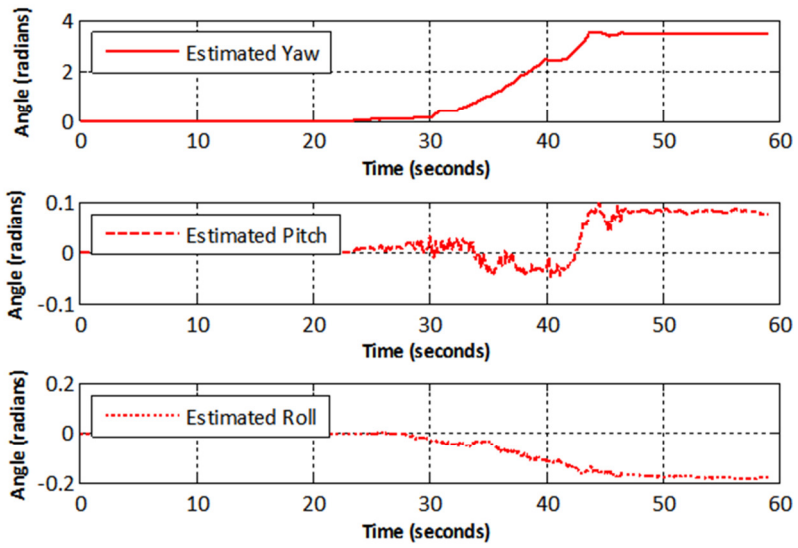

Figure 5.29. Yaw, pitch, roll estimates – fishhook experiment.

It is evident from Figure 5.29 that the ARGO has made a complete turn since the yaw angle estimate changes drastically from $time \approx 30\ s - 45\ s$. The pitch and roll angle estimates stay relatively unchanged through the course of the experiment.

Figure 5.30 depicts the yaw angle estimate compared against the ground truth value in the Y direction using the RMS data from Table 5.5. This was done to keep the yaw comparison graph simple, since the ARGO's motion through the fishhook experiment can be tracked completely in this direction (as seen in Figure 5.30).

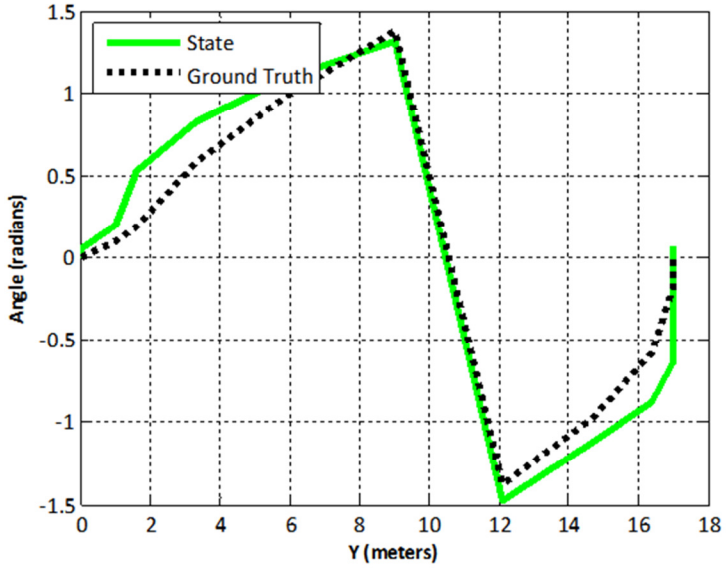

Figure 5.30. Yaw estimate comparison – fishhook experiment.

Figure 5.30 shows the yaw estimates track well with the ground truth in the *Y* direction, and the RMS value for this experiment is 0.21 rad, or 12.00 deg.

Figure 5.31 depicts the rates of change for the orientation estimates.

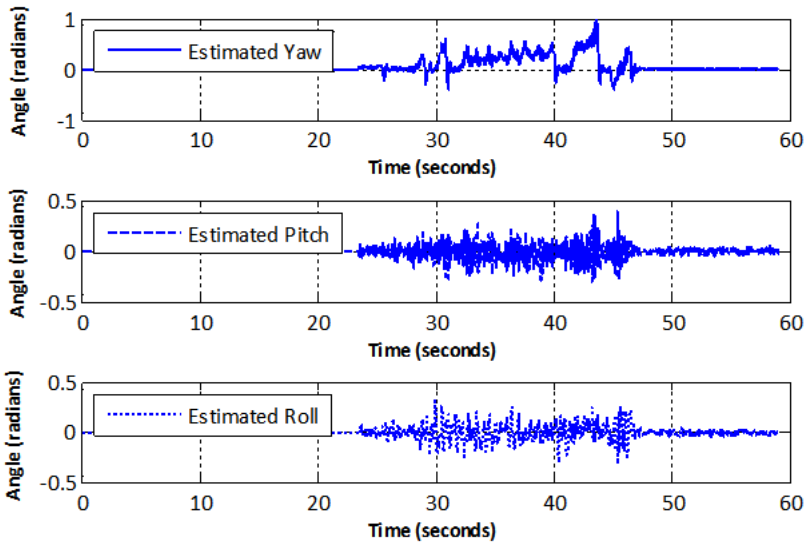

Figure 5.31. Estimated yaw, pitch and roll rates – slalom experiment.

Once again, the yaw, pitch and roll rate estimates in Figures 5.31 look similar to those shown in Figure 5.13 and 5.22 for the double-lane change and slalom experiments; where the yaw rate estimates are noise-free and exhibit significant change in magnitude, and the pitch and roll rate estimates exhibit noise and are small in magnitude.

This concludes the discussion on the experimental results, and the next chapter draws the conclusions for this thesis.

Chapter 6.

Conclusions and Future Work

This thesis presented a hybrid EKF/KF state estimator designed for a skid-steered ARGO UGV. The development of a state estimation approach that can estimate the ARGO's motion, velocity, orientation and wheel slips using an IMU, wheel encoders and a GPS sensor was presented.

The hybrid EKF/KF state estimator is modular: it estimates the ARGO's position, velocity, longitudinal slip, and orientation parameters. Initially, these parameters are estimated using an IMU/encoder with the EKF algorithm. The position estimates from the EKF algorithm can further be adjusted based on GPS data using the KF algorithm.

This separation between the two state estimation techniques (EKF, KF) allows the EKF part of the hybrid state estimator to run in real-time and as such, it is online-ready. The KF is used offline, and only in applications when an improvement in position accuracy is required and reliable GPS data is available.

Additionally, the hybrid EKF/KF state estimator was tested with experimental data gathered from experiments that were arranged in the double-lane change, slalom and fishhook maneuvers. The position estimates generated by the proposed algorithm were compared against ground truth measurements for each of these experiments and so were the yaw estimates. Using the proposed state estimator, the motion of the ARGO was tracked within one meter for the double-lane change experiment and within three meters for both slalom and fishhook experiments; which showcases the capability of the proposed methodologies in estimating the states with a reasonable accuracy.

In this thesis, an approach for estimating the longitudinal slip for every wheel on the ARGO is also presented. The slip results reported align well with the motion estimate results from the state estimator and give more insight into how the ARGO's skid-steering drive-scheme affects its driveability through the experimental scenarios. The longitudinal slip estimation method proposed in this thesis will serve as a stepping stone for future dynamic modelling and application of SLAM for autonomous navigation of the ARGO UGV.

For future work, the instrumentation on the ARGO will be enhanced with the addition of a LIDAR sensor. The LIDAR must be installed on the front of the vehicle and configured to communicate with the on-board ROS system on the ARGO's Netbook PC.

With the LIDAR fully integrated onto the vehicle, extensions to the hybrid EKF/KF algorithm will also be made to include a localization algorithm for SLAM purposes. This entails, changing the measurement model for the EKF to update the position of the vehicle according to markers or features around the vehicle that the LIDAR will sense. The velocity and yaw estimates of the vehicle can be obtained from this data as well. The encoders on the vehicle can be used to provide the prediction estimates for the position and velocity of the EKF algorithm and the IMU can be used to provide yaw prediction estimates to the EKF.

The longitudinal slip estimates can be combined with a dynamic model in a path planning technique that enables the ARGO to generate a path based on a map generated by the LIDAR. The path planning technique also includes a steering controller for the vehicle which will also need to communicate with the ROS system on-board the ARGO.

Field experiments must be conducted in unstructured environments to improve the precision, accuracy robustness of the autonomous ARGO UGV.

References

[1] A. Lacaze, K. Murphy, M. Del Giorno and K. Corley, "Reconnaissance and autonomy for small robots (RASR) team: MAGIC 2010 challenge," *Journal of Field Robotics,* vol. 29, issue. 5, pp. 729-744, 2012.

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

[3] T. Schmitt, R. Hanek, M. Beetz, S. Buck and B. Radig, "Cooperative probabilistic state estimation for vision-based autonomous mobile robots," *IEEE Transactions on Robotics and Automation,* vol. 18, issue. 5, pp. 670-684, 2002.

[4] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot Minerva," *International Journal of Robotics Research,* vol. 19, issue. 11, pp. 972-999, 2000.

[5] W. Burgard, A. B. Cremers, D. Fox, D. Hahnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence.,* vol. 114, issue. 1-2, pp. 3-55, 1999.

[6] R. G. Simmons, R. Goodwin, K. Z. Haigh, S. Koenig, J. O'Sullivan and M. M. Veloso, "Xavier: experience with a layered robot architecture," *SIGART Bulletin,* vol. 8, issue. 1-4, pp. 22-33, 1997.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, 2000.

[8] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence,* vol. 28, issue. 4, pp. 1373-1385, 2013.

[9] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2003, pp. 2743-2748.

[10] K. Lingemann, H. Surmann, A. Nuchter and J. Hertzberg, "Indoor and outdoor localization for fast mobile robots," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 2004, pp. 2185-2190.

[11] S. Thrun, W. Burgard and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping," in *IEEE International Conference on Robotics and Automation,* 2000, pp. 321-328.

[12] T. Vu, J. Burlet and O. Aycard, "Mapping of environment, detection and tracking of moving objects using occupancy grids," in *Intelligent Vehicles Symposium,* 2008, pp. 684-689.

[13] Z. Lu, Z. Hu and K. Uchimura, "SLAM estimation in dynamic outdoor environments: A review," in *Second International Conference, ICIRA 2009,* 2009, pp. 255-267.

[14] S. Carpin, "Algorithmic motion planning: The randomized approach," in *General Theory of Information Transfer and Combinatorics,* 2006, pp. 740-768.

[15] N. Vaskevicius and A. Birk, "Towards Pathplanning for Unmanned Ground Vehicles (UGV) in 3D Plane-Maps of Unstructured Environments," *KI-Künstliche Intelligenz,* vol. 25, issue. 2, pp. 141-144, 2011.

[16] H. Tehrani Nick Nejad, Q. H. Do, R. Sakai, L. Han and S. Mita, "Real time localization, path planning and motion control for autonomous parking in cluttered environment with narrow passages," in *2012 15th International IEEE Conference on Intelligent Transportation Systems,* 2012, pp. 1357-1364.

[17] A. Elahidoost and C. Cao, "Control and navigation of a three wheeled unmanned ground vehicle by L1 adaptive control architecture," in *2012 IEEE International Conference on Technologies for Practical Robot Applications (TePRA),* 2012, pp. 13-18.

[18] B. Yoon, J. Lee, J. Kim and C. Han, "Development of a navigation algorithm with dead reckoning for unmanned ground vehicles," *International Journal of Automotive Technology,* vol. 12, issue. 1, pp. 111-118, 2011.

[19] E. Hwang, H. Kang, C. Hyun and M. Park, "Robust Backstepping Control Based on a Lyapunov Redesign for Skid-Steered Wheeled Mobile Robots," *International Journal of Advanced Robotic Systems,* vol. 10, 2013.

[20] K. Kozłowski and D. Pazderski, "Modeling and control of a 4-wheel skid-steering mobile robot," *International Journal of Applied Mathematics and Computer Science,* vol. 14, issue. 4, pp. 477-496, 2004.

[21] R. Fierro and F. L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Transactions on Neural Networks,* vol. 9, issue. 4, pp. 589-600, 1998.

[22] K. T. Leung, J. F. Whidborne, D. Purdy and P. Barber, "Road vehicle state estimation using low-cost GPS/INS," *Mechanical Systems and Signal Processing,* vol. 25, issue. 6, pp. 1988-2004, 2011.

[23] W. Yu, O. Chuy, E. G. Collins and P. Hollis, "Dynamic modeling of a skid-steered wheeled vehicle with experimental verification," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009,* 2009, pp. 4212-4219.

[24] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots. The* MIT Press, Cambridge, 2004.

[25] A. Mandow, J. Martínez, J. Morales, J. Blanco, A. García-Cerezo and J. Gonzalez, "Experimental kinematics for wheeled skid-steer mobile robots," in *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007,* 2007, pp. 1222-1227.

[26] W. Seeds, "Tracked Vehicles-An Analysis of the Factors Involved in Steering," *Automobile Engineer, April,* 1950.

[27] M. G. Bekker, *Theory of Land Locomotion: The Mechanics of Vehicle Mobility.* University of Michigan Press, Ann Arbor, 1956.

[28] J. Y. Wong and C. Chiang, "A general theory for skid steering of tracked vehicles on firm ground," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering,* vol. 215, issue. 3, pp. 343-355, 2001.

[29] T. Tran and Q. Ha, "Fast algorithm for UGV wheel-terrain interaction analysis," in *2008 10th International Conference on Control, Automation, Robotics and Vision, ICARCV 2008,* 2008, pp. 674-680.

[30] T. Tran, N. Kwok, S. Scheding and Q. Ha, "Dynamic modelling of wheel-terrain interaction of a UGV," in *Proceedings of the 3rd IEEE International Conference on Automation Science and Engineering, IEEE CASE 2007,* 2007, pp. 369-374.

[31] E. Lucet, C. Grand, D. Sallé and P. Bidaud, "Dynamic sliding mode control of a four-wheel skid-steering vehicle in presence of sliding," in *17th CISM-IFToMM Symposium on Robot Design, Dynamics, and Control,* 2008.

[32] J. Y. Wong, *Theory of Ground Vehicles. John Wiley and Sons*, 2008.

[33] L. Caracciolo, A. De Luca and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation,* 1999, pp. 2632-2638.

[34] R. Featherstone and D. Orin, "Robot dynamics: Equations and algorithms," in *Proceedings of the IEEE International Conference on Robotics and Automation,* 2000, pp. 826-834.

[35] J. Martínez, A. Mandow, J. Morales, S. Pedraza and A. García-Cerezo, "Approximating kinematics for tracked mobile robots," *The International Journal of Robotics Research,* vol. 24, issue. 10, pp. 867-878, 2005.

[36] Z. Song, Y. H. Zweiri, L. D. Seneviratne and K. Althoefer, "Non-linear observer for slip estimation of skid-steering vehicles," in *Proceedings of the 2006 Conference on International Robotics and Automation,* 2006, pp. 1499-1504.

[37] T. Tran, M. Nguyen, N. Kwok, Q. Ha and G. Fang, "Sliding mode-PID approach for robust low-level control of a UGV," in *2006 IEEE International Conference on Automation Science and Engineering, CASE,* 2006, pp. 672-677.

[38] Q. Ha, T. Tran, S. Scheding, G. Dissanayake and H. Durrant-Whyte, "Control issues of an autonomous vehicle," in *22nd International Symposium on Automation and Robotics in Construction, ISARC 2005,* 2005.

[39] P. Crossley and J. Cook, "A nonlinear engine model for drivetrain system development," in *International Conference on Control '91,* 1991, pp. 921-925.

[40] U. Kienecke and L. Nielsen, *Automotive Control Systems*. Springer-Verlag, Berlin, Heidelberg, 2005.

[41] P. Setlur, J. Wagner, D. Dawson and B. Samuels, "Nonlinear control of a continuously variable transmission (cvt)," *IEEE Transactions on Control Systems Technology,* vol. 11, issue. 1, pp. 101-108, 2003.

[42] R. Zanasi, A. Visconti, G. Sandoni and R. Morselli, "Dynamic modeling and control of a car transmission system," in *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics,* 2001, pp. 416-421.

[43] T. Tran, Q. Ha, R. Grover and S. Scheding, "Modelling of an autonomous amphibious vehicle," in *Proceedings of the Australian Conference on Robotics and Automation,* 2004.

[44] X. Song, L. Seneviratne and K. Althoefer, "Slip parameter estimation for tele-operated ground vehicles in slippery terrain," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering,* vol. 225, issue. 6, pp. 814-830, 2011.

[45] J. Yi, J. Zhang, D. Song and S. Jayasuriya, "IMU-based localization and slip estimation for skid-steered mobile robots," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems,* 2007, pp. 2845-2850.

[46] Y. Yi, F. Mengyin, Z. Hao and X. Guangming, "Control Laws Design and Validation of Autonomous Mobile Robot Off-Road Trajectory Tracking Based on ADAMS and MATLAB Co-Simulation Platform," *Applications of MATLAB in Science and Engineering*, 2011, pp. 353-370.

[47] L. Ojeda, G. Reina and J. Borenstein, "Experimental results from FLEXnav: An expert rule-based dead-reckoning system for mars rovers," in *Proceedings of the 2004 IEEE Aerospace Conference,* 2004, pp. 816-825.

[48] G. Dissanayake, S. Sukkarieh, E. Nebot and H. Durrant-Whyte, "The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications," *IEEE Transactions on Robotics and Automation,* vol. 17, issue. 5, pp. 731-747, 2001.

[49] E. Kiriy and M. Buehler, "Three-state extended kalman filter for mobile robot localization," *McGill University., Montreal, Canada, Technical Report TR-CIM,* vol. 5, 2002.

[50] C. Veness, Calculate distance, Bearing, and more between latitude/longitude points, http://www.movable-type.co.uk/scripts/latlong.html, Accessed. 2012-12-12.