CRANFIELD UNIVERSITY

Anthony Dumisani Masuku

PLANETARY ROVERS AND DATA FUSION

SCHOOL OF ENGINEERING

Astronautics and Space Engineering

MSc THESIS

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING

MSc THESIS

Academic Year 2013-2014

Anthony Masuku

Planetary Rovers and Data Fusion

Supervisor: Dr. Stephen Hobbs

May 2012

This thesis is submitted in fulfillment for the degree of MSc by Research

Abstract

This research will investigate the problem of position estimation for planetary rovers. Diverse algorithmic filters are available for collecting input data and transforming that data to useful information for the purpose of position estimation process. The terrain has sandy soil which might cause slipping of the robot, and small stones and pebbles which can affect trajectory.

The Kalman Filter, a state estimation algorithm was used for fusing the sensor data to improve the position measurement of the rover. For the rover application the locomotion and errors accumulated by the rover is compensated by the Kalman Filter. The movement of a rover in a rough terrain is challenging especially with limited sensors to tackle the problem. Thus, an initiative was taken to test drive the rover during the field trial and expose the mobile platform to hard ground and soft ground(sand). It was found that the LSV system produced speckle image and values which proved invaluable for further research and for the implementation of data fusion.

During the field trial,It was also discovered that in a flat hard surface the problem of the steering rover is minimal. However, when the rover was under the influence of soft sand the rover tended to drift away and struggled to navigate.

This research introduced the laser speckle velocimetry as an alternative for odometric measurement. LSV data was gathered during the field trial to further simulate under MATLAB, which is a computational/mathematical programming software used for the simulation of the rover trajectory. The wheel encoders came with associated errors during the position measurement process. This was observed during the earlier field trials too. It was also discovered that the Laser Speckle Velocimetry measurement was able to measure accurately the position measurement but at the same time sensitivity of the optics produced noise which needed to be addressed as error problem.

Though the rough terrain is found in Mars, this paper is applicable to a terrestrial robot on Earth. There are regions in Earth which have rough terrains and regions which are hard to measure with encoders. This is especially true concerning icy places like Antartica, Greenland and others.

The proposed implementation for the development of the locomotion system is to model a system for the position estimation through the use of simulation and collecting data using the LSV. Two simulations are performed, one is the differential drive of a two wheel robot and the second involves the fusion of the differential drive robot data and the LSV data collected from the rover testbed. The results have been positive. The expected contributions from the research work includes a design of a LSV system to aid the locomotion measurement system.

Simulation results show the effect of different sensors and velocity of the robot. The kalman filter improves the position estimation process.

KEYWORDS: Data Fusion, planetary rover, Kalman Filter, navigation, Laser Speckle Velocimetry, measurements, Mars.

Acknowledgements

I am sincerely and heartily grateful to my supervisor ,Dr Stephen Hobbs , for the support and guidance he showed me throughout my study and thesis writing. I am sure it would have not been possible without his help. Besides I would like to thank my mother and my friends who boosted me morally and provided me with great information resources. A thanks goes out to Jayden Nkosikhona Masuku for inspiration. Furthermore, I would like to thank the following:

Chimu - Advice about MSc study at Cranfield

Dr Daniels (Engineering Photonics) - Assistance and advice (Laser Speckle Velocimetry)

Boris Snapair - Technical support with Matlab

Dr Charrot (Engineering Photonics) - Collaborating in this project (Laser Speckle Velocimetry and the supply of measurement data.

Heather Woodfield (Social Sciences Information Specialist) - Refworks and Write n Cite training and for being kind.

Alison and Deborah(IT Training and Support Specialist) - Support using MSWord and thesis formatting

Mrs Millia Masuku - Support and encouragement from beginning to end. Having lots of faith in me.

Fuller Masuku in remembrance.

My Tribe - Zwide ka Langa, Nhlane, Gudunkomo,Zikode, Ndwandwe, Mkhatshwa, Nxumalo, Tunda bathola bafokazane.

A Psalm of David. *23 The LORD is my shepherd; I shall not want.2 He makes me to lie down in green pastures; He leads me beside the still waters.3 He restores my soul; He leads me in the paths of righteousness For His names sake.4 Yea, though I walk through the valley of the shadow of death, I will fear no evil; For You are with me; Your rod and Your staff, they comfort me.5 You prepare a table before me in the presence of my enemies; You anoint my head with oil; My cup runs over.6 Surely goodness and mercy shall follow meal the days of my life; And I will dwell[a] in the house of the LORD Forever.* .

Contents

List of Figures

List of Tables

Nomenclature

Symbols

θ_x	Orientation in the x direction
θ_y	Orientation in the y direction
γ_{ro}	Forward speed of the rover
γ_{LS}	Laser Speckle Optics on the rover
ω_{ro}	Angular speed of the rover
Vx	Velocity component in x- direction
Vy	Velocity component in y- direction
lsv_{xpos}	LSV position measurement parameter in the x- direction
lsv_{ypos}	LSV position measurement parameter in the y-direction
S_c	Scaling coefficient
dt	Inter frame time in seconds
$\bar{x}$	Input to the state vector, KF
ω_1	Angular wheel velocity for wheel 1
ω_2	Angular wheel velocity for wheel 2
ω_{meas1}	Angular velocity under noise sources for wheel 1
ω_{meas2}	Angular velocity under noise sources for wheel 2
ω_{1True}	True angular velocity of wheel 1
ω_{2True}	True angular velocity of wheel 2
$\sigma\Delta_x$	Standard deviation in the measured pixel shift
σv_x	Standard deviation in the velocity
ΔT	Change in inter-frame time in seconds
ϵ	Epsilon for wheel efficiences / soil modelling

CA	Computer Associates
CCD	Charge Coupled Device
CU	Cranfield University
COR	Common Object Repository
ESA	European Space Agency
KF	Kalman Filter
GS	Ground Segment
GUI	Graphical User Interface
IT	Information Technology
JUST	Journal of Unmanned System Technology
LSV	Laser Speckle Velocity
NASA	National Aeronautics Space Administration
NSM	Networks, Systems and Management
RAM	Random Access Memory
SES	Societe Europeenne des Satellites
SMS	Systems Management Server
SSTL	Surrey Satellite Technology Limited

Chapter 1

Introduction

Planetary Exploration/Solar System, a significant theme in space research, has been a focus for scientists over the years. The universe is large and is made up of many galaxies,planets, and asteroids. A question usually asked by scientists is what is in our universe? Is earth the only planet that maintains life? Mankind has developed spacecraft to do the job of searching for life within the solar system and further. Planetary Rovers are important space robotic vehicles for sampling and studying the surface of other bodies. Mars is mentioned here because it is a planet which resembles Earth and is closer compared to other planets. One of the key problems of efficient use of Rovers is locomotion, in particular when the wheels of the vehicle experience slippage and adverse locomotion on the terrain.

The unmanned mission to Mars has been successful over the recent years, and challenges associated with building, transporting and deploying planetary Rovers still persist. The reason the planet has unmanned missions is the hostility to humans and, therefore, building Rovers has become an option. Currently there are plans for manned missions but they are still under assessment from the science and engineering communities. Mars, a planet known for its sandy and rocky features presents a number of problems for the manoeuvring robotic vehicles. The Rover has to be autonomous since the mission is unmanned. The sensors on the Rover need to be robust in making measurements, which is not the case because of the errors associated with those sensors. Some measurement system for the problem of locomotion of the Rover would be very desirable. The vision based system can give absolute position but lag with time updates. Wheel encoders cannot detect whether the wheels are slipping very easily. An opportunity exists to tackle a research problem in terms of developing systems that can measure the locomotion of the Rover. One candidate that is promising is the Laser Speckle Velocimetry.

The Space Research Group at Cranfield, involved in Planetary Robotic Exploration and Optics Research Group involved in laser speckle technology had an interest to develop new ways of making optical measurement for the Planetary Rover. The work involved working with Astrium's Bridget Rover to position estimate using Laser Speckle Velocimetry. Unicenter, a software product from Computer Associate is considered as an option for mission management effectiveness for the locomotion of the rover.

1.1 Robotics Missions Background

NASA has been leading over the years in missions involving planetary robots and robotic spacecraft, the most recent mission (2013) being the Mars Science Laboratory, which is currently performing scientific work and traversing the surface of Mars. Mars, a planet placed number four from the Sun is of significant interest to the scientific community probably because of the fact that it closely resembles Earth, our home planet, and the intriguing possibility that life might have existed billions of years ago. There have been a number of successful missions to the red planet using planetary Rovers [6],[7],[8], as research vehicles or unmanned robots. Some of these were:

- Mars Science Laboratory launched in 2011
- Mars Exploration Rovers launched in 2003
- Viking Orbiters and Landers launched in 1975

1.1.1 Aims

The aim of this thesis is to investigate through the use of simulation, the position measurement estimation for a planetary Rover. We imagine the Rover is traversing at a Martian surface and experiences wheel slippage.

Wheel slippage can occur locally on earth should the rover be traversing in a non flat environment e.g soft sand, rocky surfaces and other rough terrains. The Laser Speckle Velocimetry system is an option for discovering or gathering novel scientific data for the locomotion of the Planetary Rover. Through the use of simulation the data collected would be useful for further research.

This dissertation, seeks to address the planetary Rover motion in places like Mars, though not limited to that planet, terrestrial planet like earth may take advantage of the research findings. Through the use of simulation and real data fusion to the simulations we gain the opportunity to learn novel ideas to solve the pose estimation in that planet, e.g. when a planetary Rover experiences wheel slippage on the surface. Laser Speckle Velocimetry is a novel idea towards making accurate measurements for the purposes of displacement. The Antarctica robot[9] used the LSV technology since that region is icy and odometric sensors, including other sensors, was not going to suffice because of the icy surface. The Rover would be easy to track if we had more reliable and accurate measurement data. We do not want to gather false measurement data in the event of wheel slippage and other adverse locomotion of the vehicle. There are a lot of advantages associated with using LSV system for odometric purposes.

- Accuracy
- Sensitivity
- Robust

So for the above reasons, the technique should enhance the motion capability of a Rover compared to current methods of measurements e.g. encoder sensors.

1.1.2 Objectives

Some objectives to be achieved are:

1. Review sensors relevant to planetary rover locomotion to enable data fusion, including speckle velocimetry.
2. Demonstrate data fusion to obtain locomotion information using a simulation which combines two or more sensor readings (a Kalman filter or similar algorithm may be used).
3. Assess the benefits of planetary rover locomotion data fusion, especially its potential to provide information relevant to planetary science.
4. Discuss the data architecture for a hypothetical planetary exploration systems to assess its similarities to IT infrastructure management and the potential for planetary exploration to therefore benefit from terrestrial infrastructure management systems.

Two-wheel Rovers have gained a lot of interest in research due to their ease on modelling and implementation. The simulation is designed to be a simple model of a rover, while allowing data fusion to be simulated. Estimating the position for a two wheel robot is achieved using MATLAB simulation as presented on appendix D, and then an attempt is performed to integrate other sensors on the simulation to see the effectiveness of sensor fusion. A mathematical algorithm known as the Kalman Filter is responsible for the fusion process for this project. There are other variants of the Kalman Filter algorithms that may be used but for the implementation of sensor fusion, the linear KF was used. The measurement data for speckle velocimetry and the Astrium rover are available, and are used to produce realistic simulations and robust results. The subject of sensor noise and modelling will be addressed, and good knowledge of the sensors involved will allow us to make better estimation.

1.2 Dissertation Structure

In Chapter 2, a presentation of literature covered to support research topics, covering planetary Rovers, optics and measurements, Kalman Filter and other relevant topics enhancing knowledge in various space exploration topics.

In Chapter 3, Rover Modelling is introduced with the method of integrating measurement data and simulation is presented. Various functional blocks of the LSV system, the implementation of the position estimation algorithm in MATLAB software and general discussions are explained.

In Chapter 4, Kalman Filter, the introduction, requirements, algorithms and results are implemented and discussed.

In Chapter 5, Results and Simulations, MATLAB code results from appendix A to appendix D

In Chapter 6, further discussion is given which affirms the subject matter of planetary Rovers. The analysis of test results is discussed and a comparison with the related work.

In Chapter 7, Data Management and related Techonolgies is presented. This chapter gives a summary of the IT Infrastructure and Enterprise Management, employs ideas to management of sensors and other aspects of space technology. It gives an account for future opportunity of the work related.

In Chapter 8, the conclusion of the research work is presented. This chapter gives a summary of the research and offers conclusive findings from the research. It gives an account for future opportunity of the work related.

Appendices and References will follow Chapter 8, all referring from earlier chapters of this dissertation to the last chapter.

Chapter 2

Literature Review

Mankind has shown a keen inclination in planetary exploration for many generations and continues to be deeply intrigued by it even in the modern era. Our quest for knowledge about what is out there in the universe, beyond our planet Earth, forms part of the equation defining what space exploration is.[10] In the past 60 years, space agencies from different nations have taken the initiative to answer the fundamental question, Is Earth the only planet where life exists? Freedman [11] explains the theory behind the universe through the use of the hubble telescope. Mars, the planet that has gained major attention of the scientists due to the possibility that life might have once existed on it. The planet is currently being assessed as a platform for localization and navigation for human exploration[12]. Coustenis,[13] ,on the contrary, focus their attention on the icy moons(Titan and Ganymede) of the giant planets to determine factors that led to the creation of the solar system, and also to investigate astro-biological activities. These moons exhibit Earth-like features and have engaged the mind and exploratory study of the scientific community.

Planetary exploration requires experts from different fields of knowledge, such as missions, technologies, science, and people, to put their efforts together. It is, therefore, fair to consider planetary exploration as a multi-disciplinary subject and agenda[14],[15]. Paar [16] throws light upon the initiative taken by ESA (European Space Agency), which is a major space agency constituted largely by the countries from the European Union. An independent body (PRoViscout) was set up to study and demonstrate sending robotic missions to other planets. Among the many objectives of this organisation, the following aims are of primary significance for space exploration:

- Definition of space exploration
- Design of efficient payloads
- Optimisation of science data from spacecraft
- Collection of planetary samples (e.g. from Mars) for study on Earth

- Public awareness education

2.1 Planetary Rovers

Just like humans have diverse sensors to see, hear and detect, planetary Rovers are equipped with sensors to aid in navigation and performing scientific objectives [17],[18],[19]. For the Martian environment it would not have been be possible for the Rover to navigate if sensors were not part of the subsystem of the vehicle. A sensor may be defined as a device capable of measuring a physical property. Some of the sensor devices that might be found on a planetary Rover are:

- Sun Sensor
- Sonar
- Temperature
- Camera
- Encoders
- Accelerometers

Sensors may be classed as proprioceptive or exteroceptive. Examples of internal sensors or proprioceptive sensors are encoders and gyroscopes or sensors measuring the internal of a system while exteroceptive sensors measure the external of a system or sense the world. Examples of these are sonar sensors, radars, laser range finders and cameras. In many cases these are implemented depending on the application of the particular robot. Figure 2.1 is an example of sensors that might be found on a Rover.

2.1.1 On board Sensor

For a planetary Rover to make a good traverse it needs a sensing mechanism to aid during the navigation process,[20],[17],[21]. For a planet like Mars, specific sensors are needed for the operation. These sensors will need to be combined through fusion process in order to provide an optimized navigation trajectory. A sensor fusion process is needed to compensate for the errors associated with the sensors. For a wheel encoder the the reading from the odometry tends to be inaccurate due to the adverse terrain associated with the environment, visual cameras may not be calibrated properly or in some cases lens defects from manufactures may cause undesirable errors during the traverse of the rover. A common algorithm used for the fusion process is the Kalman Filter,[22],[23]. Some sensors for the Rover might

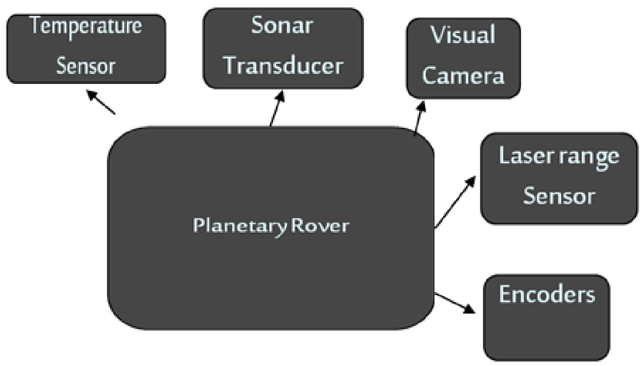


Figure 2.1: Planetary Rover

be close to error free but through the traverse of the vehicle these errors may grow adversely after a period of time.

The sensors of interest found in most planetary Rovers are wheel encoders, inertial sensors, accelerometers and visual cameras,[24],[25]. It is also possible to utilize beacons or landmarks to make measurements. Our interest is mainly on board sensors and their usefulness in the measurement process.

The odometry model is based on encoders that measure the revolution of the wheels. However, though these may be good for measuring the relative position of the Rover, they often accumulate errors on long runs,[26]. It is a common practice to fuse encoders with other sensors or absolute measurement to have an optimal position estimate using the KF. The absolute measurements are based on landmarks on the environment,[27].

Taking sonar as an example, the sensor has the ability to send sound waves and interpret the reflecting wave in the form of measurements.[28],[29],[30],[31] An opportunity exists to fuse the sensor with encoders to optimize the position estimation process for a planetary Rover. So for the navigation process may be looked at as a complex agenda in the sense that various models to sensors need to be understood or taken into account before implementation on a Rover vehicle. Furthermore, this takes a lot of simulation and time in the entire process of implementing sensor fusion for the Rover.

One of the apparent applications of Rovers for planetary exploration is the traversing of the vehicle in hostile environment and exploiting the environment by collecting samples and scientific data which is further relayed back to Earth. The Rover is designed to be semi-autonomous since onsite human intervention is not possible.

Curiosity Rover, designed by NASA, has been traversing on the Martian surface to perform some science objectives. It is currently examining the environment including the recent completion [July 2013] of the analysis of soil and rocks at a region called the Glenelg [32]. The area is said to have different types of terrain intersecting. The mission is expected to last at least two years. The robotic arm has been a busy module responsible for collecting samples from the environment and depositing it to the container system for further analysis [7]. Lindemann [33] gives further insight by mentioning the traversing application of the Curiosity Rover. The Rover has been designed to counter-manoeuvre obstacles and navigate on an uneven terrain and loose soils.

Howard [34] mentions some of the specifics of the recent mission to Mars and the planetary Rovers involved for the specific task. He describes the planetary Rovers systems pertaining to Mars Science Laboratory (named Curiosity), and Mars Exploration Rovers (MER) which is made up of two Rovers, Spirit and Opportunity.

2.2 Locomotion Subsystems

The planetary Rover is a machine / unmanned vehicle or robot made up of diverse subsystems which enable it to move, sense and perform diverse automated tasks. The unmanned vehicle is usually deployed in a foreign planet. Examples of known planetary Rovers in planetary exploration are MER mentioned earlier, Pathfinder, Curiosity and others. All the subsystems of a planetary Rover should work collectively for the success of the science mission.

Spirit and Opportunity, (MER) are examples of Rovers carrying scientific payload for Mars mission. Both of these Rovers where designed to the same specifications in terms of subsystems and platform structure but were deployed in different regions on the Mars planet to achieve science goals [33]. An example of a typical Mars Rover subsystems is shown on Figure 2.2:

Farritor [35] describes the mobility system of the Sojourner. The Rover was designed using the rocker-bogie wheel suspension. This is a part of the mobility subsystem as shown in figure 2.2. The Lightweight Survivable Rover (LSR-1), a prototype, showed the Rover to move around slopes of the terrain, climb rock obstacles and have the wheels function independently from each other since each had its own motor.

Another subsystem of interest is the Telecommunications system under the heading Comm in fig 2.2. Hilland [36] explains the module. Telecommunications Subsystem in the context of Spacecraft command / control and telemetry information and the limited space inside the Rover and on the Rover deck which determines the build of

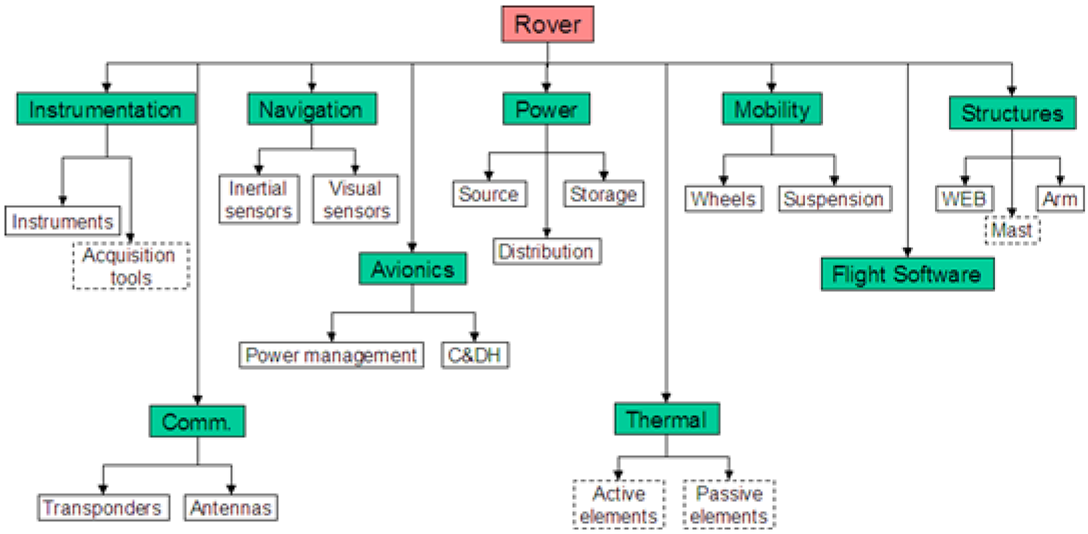


Figure 2.2: Rover Subsystems,[1]

the Telecommunications Subsystem (TEL). He further outlines the sub-modules as listed

- the Radio Frequency Subsystem (RFS) operating at 7.1/8.4GHz (X-band), the Ultra High Frequency (UHF) Subsystem operating at 401/437MHz, RFS and UHF
- Antenna Subsystem and the Radar Altimeter Subsystem (RAS) operating at 4.3Ghz (C-band)
- The RFS solid state power amplifiers (SSPA) are the only hardware redundant subassemblies
- The MER Telecommunications subsystem provides command uplink/downlink/science data direct to Earth (DTE), high volume science data and back-up command and telemetry via orbiter relay.
- The RFS and UHF subsystems are mounted inside the Rover

2.2.1 Rover Navigation

The process where the Rover can traverse on a certain path, planned or unplanned, is known as Rover navigation. In the case of Mars, the terrain or environment is often hostile and different mechanisms are developed to assist the Rover to traverse in such conditions. In such a process, some mechanisms which need to be considered are utilizing the Rovers on board sensors / in situ instruments to be aware of its environment surroundings, Where am I question and where am I going.

Woodman [37] introduces the topic of inertial navigation which is a measurement technique for determining position and orientation from a known location. This is valuable for a planetary Rover because often when a Rover is traversing we would like to know where the vehicle is located. In this instance, the Rover may utilize accelerometers and gyroscopes for the inertial navigation.

Another navigation sensor technique which is common to planetary Rovers is the odometer. These are also common on robots and cars. Odometry is the measurement of distance by counting pulses from wheel encoders. Odometry determines the position through the use of pulses over time. However, though acquired data from the Odometry is accurate on an even terrain or surface, the odometry data is unreliable for uneven surface or rough terrain which is often the case in Mars (surface has rocks) or under loose sandy soil. This is due to wheel slippage and other adverse actions of the wheel experienced during the traverse.

Adam [38] gives an example of fusing sensor readings from diverse sensors with the Odometry in order to correct the position of the Autonomous Ground Vehicle. He mentions the use of range sensors, and map reading fused with odometry information to establish position. Information was obtained from known landmarks and reflected laser signals from known sites.

Meanwhile Chenavier [39] sees location as basic in defining navigation and looks at diverse techniques for estimating position. He mentions position relative to known landmarks and localization to global environment. Martinelli [40] reaffirms the importance of odometry measurement for mobile navigation. There is a mention of fusing encoding data from diverse sensors, as mentioned by Adam [38], for the localization process. One of the sensors has to have a robust external sensing mechanism to avoid the accumulation of errors as the Rover traverses. During the fusion process the noise accumulation is going to be defined by the odometry modelled error covariance which enables the errors from the odometry data to be known and then dealt with. In determining errors the mobile Rover can be designed in such a way so as to reduce errors while traversing and knowing the state of the Rover system.

Visual odometry is the acquiring of position and orientation data through the use of images. The use of stereo cameras for visual odometry has proved invaluable for recent Mars missions. Planning route path in detail using stereo vision cameras has echoed success for Mars Exploration Rovers. The only downside to visual odometry is its complex computational image processing algorithm and its use of demanding computer power. In the MER mission, visual Odometry was only turned on when the alternative basic Odometry struggled as the Rover traversed on the surface of Mars e.g. unforeseen obstacles [41].

Delgado [42] describes how visual odometry works. Images are compared at different times; in this way a change in position and orientation is determined. Some features of interest are easily trackable via image comparison. Also, as the Rover is traversing or is in motion in relation to the features detected, displacement can be measured. To sum up: the specific features in the images are measured and thus defining the operation of a visual odometry system. In his experiment he used various hardware (two Logitech web-cams, as stereo cameras, mobile robot) and software tools (Camera Calibration Toolbox for MATLAB, public domain MATLAB toolbox) to find key points with the use of an algorithm, from the matching of images, of which accurate results were obtained [42].

Probably a good reason for the use of Stereo vision for Rovers is that it is not active technology whereby the camera will require a lot of power to operate but instead it is passive, using illumination provided by the Sun [43].

Se [44] looks at visual odometry as the extraction of the points in image sequences and calculation of motion between frames. It is not a demanding task for comparing image features while computing and extracting information from images. He mentions the advantage of using stereo images because it offers an efficient mechanism for tracking motions and scenes.

2.2.2 Wheel Bogie System

For the ExoMars Rover mission a study was conducted to design a locomotion system for the Rover. There were several wheel arrangements which were revised and a decision was taken to adopt the 6-wheel arrangement. This likely offered good traversability on the Martian surface and was in line in managing the constraints of the ExoMars Rover costs, structure and subsystems [45]).

The Wheel Bogie System (WBS) is a subsystem of the mobility functionality of the Rover and some of the system that makes up the WBS are:

- Actuators
- Rocker
- Differential Bar
- Wheel Encoder sensors
- Motor Electronics.
- Robot Technology for navigation and localization for the mobile subsystem [45] .

It is no surprise the Rocker Bogie is being used by Curiosity Rover due to its success with its predecessor, the Mars Exploration Rover. It was NASA that initially developed the concept of the Rocker-Bogie system. The Rocker Bogie is a unique mechanical design of the suspension in relation to the chassis and the functionality of the mobility independent wheels of the Rover. The fact that it does not have springs assures that the wheels are in ground contact all the time. This is an added advantage for the mobility of the Rover [45].An example of the Rocker Bogie for Curiosity is shown in Figure 2.3:

Figure 2.3: Curiosity Rover Rocker bogie,[2]

The recent traverse of Curiosity has accumulated damage on the wheels. This damage is unlikely to have a major impact on the mission since engineers are hard at work and have analysed the situation.[3] It seems to me sensors for predicting path planning are critical and position measurements are as important. The area of research of dealing with the locomotion of a planetary rover is critical as this would address the issue of errors on sensors as well as understanding the terrain traversing such hostile environments. The outcomes for such research would produce better engineered sensors and superior mechanically designed wheels to counter manage adverse terrains. Figure 2.4 below shows the damage to the wheel (Curiosity Rover)

Figure 2.4: Curiosity Rover Wheel Damage,[3]

2.2.3 Simulation of a two wheel rover

With a differential drive robot it is necessary to develop a simulation model and later implement sensor integration in order to measure and test the motion or the traversing of a robot in various situations. The environment might be non-obstacle and wheel friendly, unstable soft sand region (wheel slippage) and obstacles bound area which affects motion of the vehicle. There are many sensors which can be integrated on a robot but for simplicity focus on this section presents some basic equations to allow the motion of a simple planetary Rover to be simulated. The model can then be used to test various sensors.

Since each wheel is independently controlled, we can control the turns i.e. turn left or turn right. Some of the sensors that might be implemented or considered are:

- Encoders for measuring the rotation of the wheels
- Position measurement sensors
- Speckle velocimetry to measure the forward speed

2.3 Optics and Measurements

There is an interest in industry for the exploitation of 3D measurement on robotic vehicles. The measurement technique is currently assessed in industry and an opportunity for accurate measurement for diverse applications in the robotic world has encouraging outcome [46]. Techniques that can be utilized for the purposes of 3D measurement:

- Laser Scanner
- Interferometry
- Photogrammetric

The above techniques are explained in detail by Chen [47]. He further describes industrial application of 3D measurement and the optics involved. Meanwhile, Yachide [48] describes this measurement technique utilizing light section method smart image sensor for the purpose of range calculation or mapping. In robotic applications, 3D measurements are yet to be extensively exploited since the accuracy of these can be of major benefit especially for the localization problem for a planetary Rover.

2.4 Speckle Velocimetry

Some options for the measurement of velocity has been studied and implemented. Aizu [49] mentions the two optical measurements, coherent and incoherent. In the prior, it is the phase of the wave fronts and amplitude which is measured from the laser light, while the incoherent uses the information from the light intensity on the observed scene or object source, [49].

Aliverdi [9] suggests that the use of GPS and odometry signal is unreliable for an Antarctica Rover. The reason for this is that the Antarctica environment is considered hostile (snowy and icy conditions) and measurement from such a difficult region as the Rover is in motion tends to be challenging. He introduces the subject of Laser Speckle Velocimetry which he further exploits in order to get more precise measurement reading for Rover applications in such an environment or difficult region.

Laser optics is used for illuminating the surface, the profile of the surface will be subject to interference due to the rough surface, and thus speckles are produced. These speckle representations are due to the different sizes of points of the speckled image. The phenomenon is based on the light source from the laser and the roughness of the surface. This again defines the subject of laser speckles. He mentions the advantage of using laser velocimetry to solving robot navigation for high accuracy and fast response applications [41]. Though the measurement technique is a promising technology with fast turnaround in accuracy, it is a technology that is in its infancy and challenges are significant:

- the setup of the Laser Speckle based system is complex.
- A highly sensitive system in turn producing inaccurate measurement results

- Measurement of the x and y is no problem but along the z direction is a challenge.
- Safety when dealing with lasers optics.

In further defining the velocimetry system, he elaborates speckle velocimetry as the movements of parts i.e. the camera and the vehicle, in relation to the ground speckle being measured, [41].

2.4.1 Speckle Patterns

Zizka [50] defines the interference of wave fronts all propagating towards a target as a speckle pattern. This pattern is often observed through a noisy image. The combination of speckle and laser may have a desirable application in the industry, especially for the purpose of measurements. The result of speckle imagery is due to the combination of diverse waves based on the same frequency producing an interesting resultant wave with unique properties of phase and intensities. Usually these waves can be added together using a mathematical formula and the phase relationship can be realized. The apparatus which can be used for this type of observation are:

- Laser diode
- Projector
- Image sensor

Brillaud [51] describes the application of speckle pattern being used for developing strain mapping. The method used for calculating the strain values is taking the spatial derivative of the measured displacement vectors. By comparing two images of the sample, the values of the vectors are calculated. This is the basis of speckle pattern measurement.

He further elaborates the techniques used to perform this type of measurement and the results produced from the interferometric laser speckle application also mentioning the benefits for high sensitive application to strain measurements. In order to get accurate measurements from speckle images the speckles from one image should be correlated with speckles from the previous image. In this way there will be cross correlation or position shift of the features from the two images. [41]

2.4.2 Laser Speckle Method

For the measurement technique, a laser, coherent light will illuminate a surface of interest, and the light source will produce a scattering on that surface. Depending on the structure of the surface, the effect of amplitude and phases of the wave can be observed including sporadic processes. Francis [41]describes two types of speckles namely, the subjective speckle and the objective speckle. For the subjective speckle summary, the viewing parameters like lens and location of the image detector affect the speckle pattern. The objective speckle on the contrary utilizes the geometry of the moving optics system and the wavelength of the coherent light (laser light).

Figure 2.5 is an example of laser speckle imaging setup:

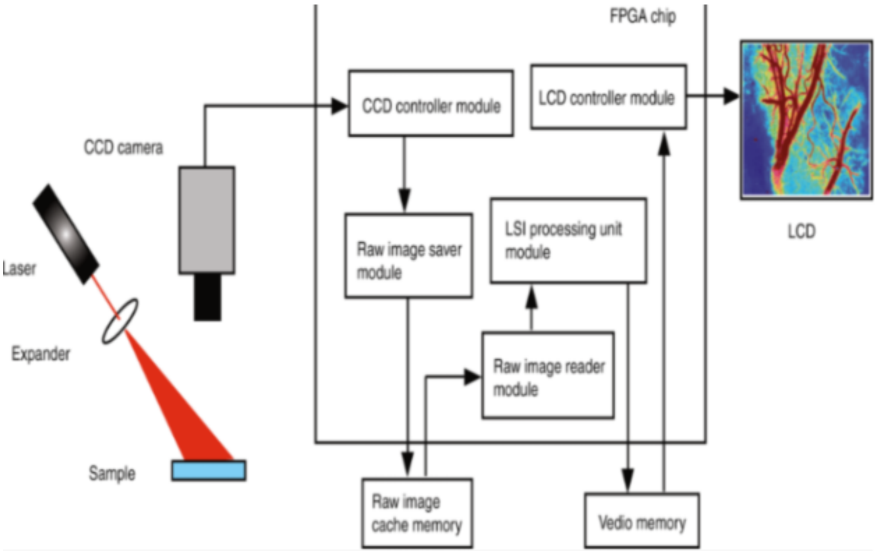


Figure 2.5: Image Speckle Apparatus,[4]

Today the industry has diverse applications using the speckle pattern technology. Some of these being:

- Electronic Speckle Interferometry
- Medical (eye testing)
- Stress, Strain and Vibration to objects

For robotic vehicles there is an opportunity to exploit the technology for measurement endeavour. Laser Speckle Velocimetry extends the idea to measure velocities for x and y coordinates system (vx and vy) and has the ability to see cross correlation.

The technique we are probably interested in is dynamic speckle which basically measures the optical flow in time or the changing speckle pattern in time. There are diverse algorithms which can be evaluated for the purposes of this measurement and these are mentioned in Francis paper [41]. For small translations of the rover the recorded speckle pattern translates. Hence the velocity can be measured by recording to two closely timed images and determining the shift in the speckle pattern. This is illustrated from figure 2.6 to figure 2.8. The computed normalised cross-correlation for shifts is upto 32 pixels.

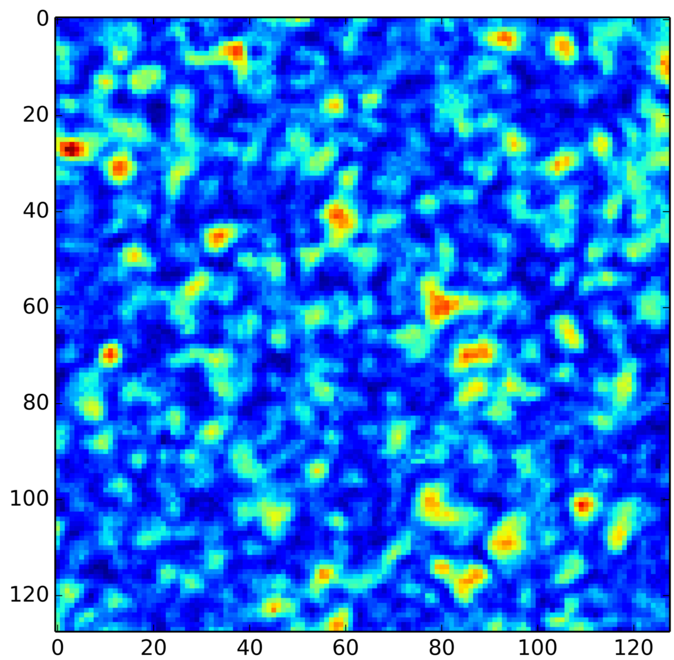


Figure 2.6: Speckle Pattern

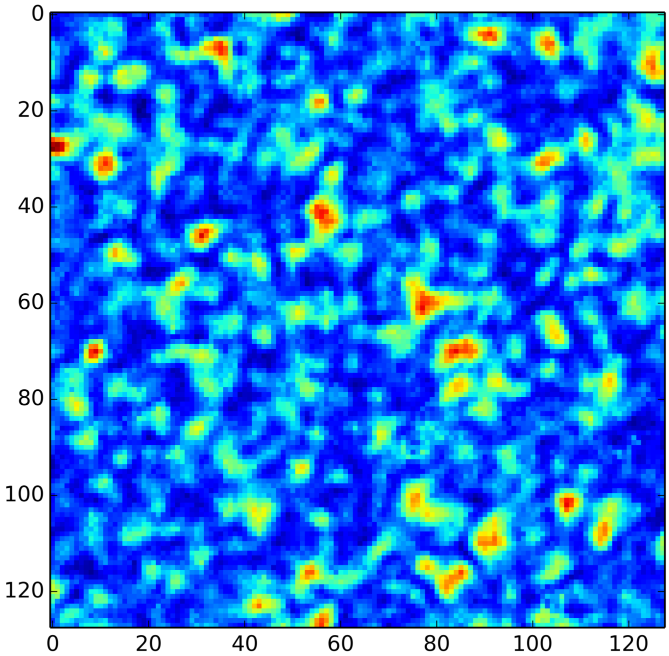


Figure 2.7: Speckle translated due to rover motion

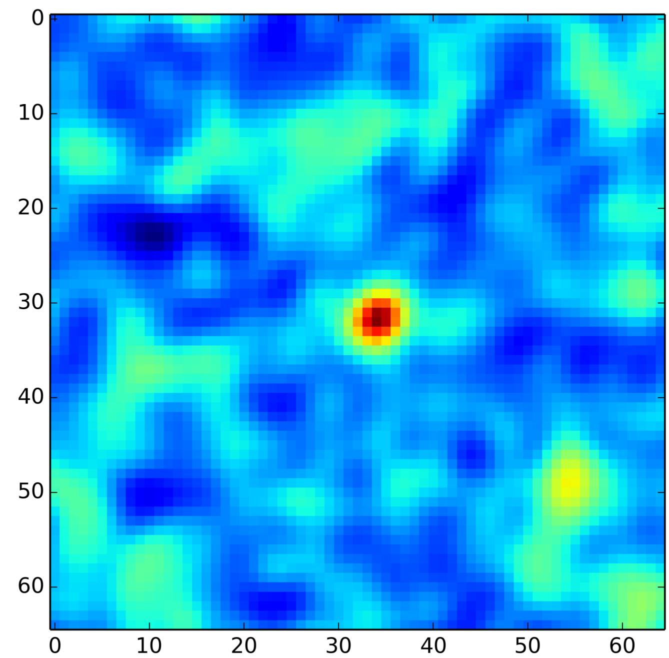


Figure 2.8: Normalised cross-correlation for shifts

2.5 Speckle Categories

2.5.1 Subjective Speckle

The process when speckle is imaged the term subjective speckle is often used. Coherent light is used to form this image. What actually determines the speckle pattern are the components of the imaging system. So, if the optics of the speckle system is altered in anyway so do the speckle pattern image. Some components which determine speckle pattern are:

- Laser beam
- Optic lens resolution
- Position of the optics

2.5.2 Objective Speckle

The speckle pattern formed from the adjacent scattered light is known as objective speckle. So this type of speckle is lensless meaning it does not depend on the optics like the subjective speckle image system setup.

The variation of the waves from the scattering phenomena contributes to objective speckle. So the wavelength of the individual waves are of not equal size and thus affect the size of the speckle. So if we would want to determine the size of the speckle, some information that would be of interest and needed to be known may be the following:

- Light wavelength expressed as lambda
- Distance from speckle pattern (e.g from electronic detector) and illuminated region surface.
- Area where light got scattered
- Phase correlation of light scattered

Objective speckles can also occur in either the fresnel region or the fraunhofer region. This means that speckles are observed further away from the scene of the object(fraunhofer diffraction) and can also be observed close to the object(fresnel diffraction)

2.6 Data Fusion

Data fusion may be taken as the process of combining multiple data or sensor information in the case of a Rover and other systems to turning the result of thereof to a useful output or outcome [52],[?].The combination of useful and not so useful data is applied or exploited.

The improvement of accuracy is due to be achieved in order to sustain a specific task. Since this thesis is about Planetary Rovers and sensors, data fusion can be useful for combining diverse sensors to auto correcting locomotion and positioning of the Rover[27] . Data fusion research is usually found in diverse fields and some of these are :

- Aerospace
- Space Technology
- Military
- Finance

The expectation is that fused data is more informative and synthetic than the original inputs.

Once all the sensors of interest have been modelled and data results have passed satisfactorily, the fusion simulation needs to take place as a final step to data fusion. Sensors of interest for our sensor fusion process are:

- LSV sensor
- Accelerometers
- Encoders
- Compass
- Sonar
- Visual Camera

More can be added to improve the estimation process. Initially the simulation was performed in MATLAB, the fusion process started with two sensors, LSV sensors and the encoders. All the mathematical model of this being explained in chapter 3.

2.6.1 Kalman Filter

Today in research diverse filters are currently being assessed and utilized for the purposes of estimation process in diverse applications.[53],[54], [53], [55] Joseph [56] presents the predictive tracking algorithm as an alternative for basis for user position for VR system or tracking new images for each frame. The unique basis for this algorithm is the double exponential smoothing measurement model.This algorithm may also be beneficial for our position estimation endeavour.The KF is chosen as an option for its popularity in research and ease of use.

The Kalman Filter has been integrated to a lot of applications in diverse industries since the early 60s, when Rudolf discovered the power and benefits it produces [57]. Since this thesis is about data fusion, a common implementation for the fusion process is making use of the KF. There are diverse KF algorithms for specific application but for our purpose a simple linear KF should enable the sensor fusion sufficiently.

The KF, a recursive mathematical algorithm should be able to deal with noisy sensors and provide us with a good estimate for our two wheel Rover application[58]. For the filter to work optimally specific parameters for the algorithm need to be defined and updated. This will be explained in chapter 3 when the implementation of LSV and other sensors are formalized.

2.7 IT Infrastructure

Since there is an interest to proactively manage the operation of the Rover from ground segment, a unique product from Computer Associates (CA) has the ability to add value to the space operation activities. CA Unicenter NSM is an Enterprise Management product from CA; a company based Islandia, New York. This author has extensive experience working with this product, gained over 6 years while working for several IT services companies in the UK, USA and South Africa. The following observations regarding CA Unicenter NSM are derived directly from this authors experiences.

CA claims that its IT Management e-business suite is robust and can offer e-business management regardless of IT platform. Computer Associates [60] through the implementation of agent software and management, IT infrastructure can be comprehensively managed regardless of platform and hardware. It is possible to extend the functionality to space exploration activities ranging from ground segment infrastructure and space segment management. Some platforms of interest that can be managed:

- Windows platform (All versions)
- UNIX (Most versions)

- Linux (All versions)
- AS/400
- Netware
- Networks(TCP/IP,SNA,IPX/SPX, DECnet)
- Mainframes

The above platforms are not all necessarily part of the space operation infrastructure but a summary or idea what can be managed utilizing Unicenter.

2.8 Literature Review Summary

In this paper we present various topics outlined in the literature chapter and implement a solution for our research topic. Various topics from diverse authors have been gathered to gain an understanding of data fusion and planetary rover.These topics help in understanding the invaluable knowledge involved in our research endeavors. Planetary Rovers research can be valid for wide application including here on Earth. So it is a very important topic which will enable the opportunity for further research particularly utilizing KF and LSV data collected from the field trial.

Since the author has been involved in IT Management, an introduction in the literature review last subsection enables or suggests an option to further study IT and Planetary Rovers as it contributes to data fusion and rovers. The next chapter is the methodology drawn from the literature section and further conclusions are later discussed in later chapters.

Chapter 3

Rover Modelling

3.1 Design of the LSV

For the LSV to have a successful implementation different blocks of apparatus were needed. These ranged from embedded computer to the Bridget Rover itself. The setup was made possible by the combination of engineering optics group and space research group (author part of the group) from Cranfield University.

3.1.1 Aim

The primary goal for this project is to gather measurement data from the LSV system and evaluate this data x and y for position estimation measurement purposes. This would further be run under a kalman filter to improve the position estimation process.

3.1.2 Constraints

The Laser Speckle Velocimetry, (LSV) system parameters consists of laser optics that enables the measurement of speckles in an image [41]. The translation of the speckle pattern in the image plane between consecutive images is calculated and then scaled to determine the translation of the vehicle [61], .The change in position of the vehicle from a particular starting point is found through integration of translation measurements.

The wheel encoder accumulates errors over time. This is the case especially when the vehicle experiences wheel slippage. Different wheel sizes could also contribute to the uncertainty in the measurement of position.

Visual odometry is another technique but requires demanding computation for the image processing. Close to true trajectory of the vehicle can be achieved with LSV and any fusion with other sensors enhances the opportunity for position estimation.

There are a number of techniques that can be used for computing speckle translation, such as the optical flow algorithms and the cross correlation technique. The optical sensors are attached to specific body frame coordinate system of the Rover.

Unless the optical laser sensor is placed at the point of interest on the robot, measurements directly from the optical sensor will produce significant errors [62]. Additionally, because the optical sensor only provides displacements in the x and y directions, information about the angular displacement may prove to be a challenge to read, though it is possible to implement sensors that measure, θ_x,θ_y.(orientation on the x and y directions) Some immediate suggestions are probably implementing several laser optic sensors, and measure the angular velocity in respect to the robot body frame and other sensors. The change in distance of the Rover may be influenced by velocity at the sensor (dx/dt and dy/dt) within the body frame.(here dx/dt and dy/dt are the change in x and y directions over time, dt)

3.1.3 System Modelling

The kinematic constraints of a differential drive robot allow the calculation of movement, given this velocity. Specifically, the centre of the robot is assumed to move only in a direction perpendicular to the wheel axis. The following equation relates the velocity of a point on the axis of rotation of a rigid body (robot centre) with a point not on that axis of rotation (the sensor location):

$$\gamma_{ro} = \gamma_{LS}.\gamma_{ro}.\omega_{ro}.d \tag{3.1}$$

where γ_{ro} is the forward speed of the rover ,γ_{LS} is the laser speckle optics measuring speed within the LSV optics coordinate system on the rover , ω_{ro} is the angular speed of the rover and d, is the distance between the center of the rover and the laser speckle optic. If we had several sensors mounted on the rover. The equations expand to the following :

$$\begin{aligned}\gamma_{ro} &= \gamma_{LS1} + \omega_{ro} \times d_1 &(3.2)\\ \gamma_{ro} &= \gamma_{LS2} + \omega_{ro} \times d_2 &(3.3)\\ \gamma_{ro} &= \gamma_{LSn} + \omega_{ro} \times d_n &(3.4)\end{aligned}$$

It is important to note that γ_{LSn}, measures vx and vy, the distance d_1 is dx by dy and therefore the vector becomes

$$d_{LSn} = \begin{pmatrix} d_x \\ d_y \end{pmatrix} \tag{3.5}$$

Equation 3.4 can be rearranged to solve for γ_{LSn},

$$\gamma_{LSn} = \gamma_{ro} - \omega_{ro}.d_n \tag{3.6}$$

Again, placement or mounting of the LSV sensor needs some planning for good reading of the velocity in the direction they are pointing at, within the body vehicle coordinates. If that is not the case we accumulate errors [62]. The LSV sensors do come with errors associated with them and will need to be modeled for the Kalman filter.

$$\gamma_{LSx1} = \gamma_{ro} - \omega_{ro}.d_1.c_1.randn \tag{3.7}$$

The above equation 3.7 is a simple model equation of a Laser speckle which can be implemented on matlab to test Vx measurement with *c1* taken as a constant, measuring noise source, and the *randn()* producing a 0 mean and standard deviation of 1 of the measurement. n, symbolizing the number of LSV sensors mounted on the vehicle.

3.2 Assembly of LSV

The LSV was mounted onsite at the Astrium premises, Stevenage. This involved bolting the T shaped aluminium plate (which held the actual laser and detector optics) to the front middle of the Rovers aluminium structure, plugging in cables which were going to transmit data to the embedded computer.

Camera (detector optics) consisted of the following:

- Baumer HXC-13 CMOS camera,
- Large Sensor (1.4 sensor)
- 1024 X 1280 pixels (full frame, we are operating with a region-of-interest mode 128 X 128 pixels)
- each $14\mu m^2$ with data output in either 8- or 10 bit (we used 8bit mode for higher frame rate)
- Maximum frame rate of 500FPS (again smaller region of interest allows frame rates of up to 10 kHz We used a frame rate of 1000Hz to ensure speckle pattern was not shifted too far between frames.

Laser consisted of the following:

- Photop DPGL-3020 DPSS system

- 20mW of radiation emission at about 530nm

- Used a laser line band-pass filter at (532nm) to reduce background signal

Figure 3.1 shows the mounting of a LSV equipment taking place on Bridget Rover:

Figure 3.1: LSV Assembly

Figure 3.2 shows the mounted LSV and the Rover taking measurements as it is in motion.

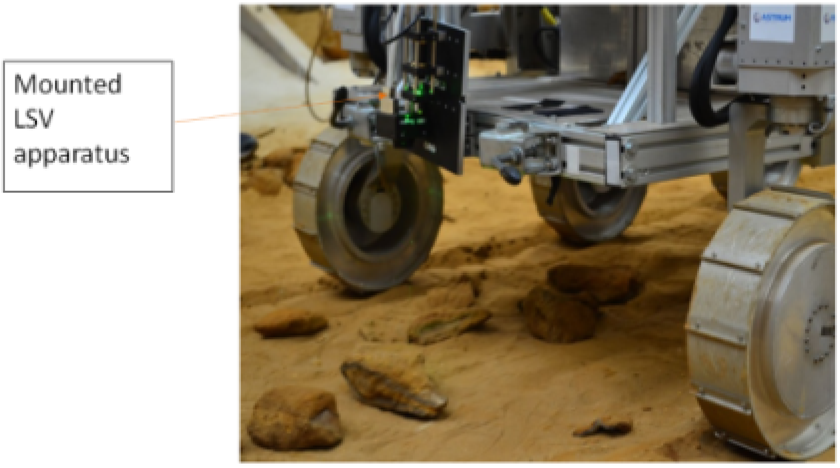


Figure 3.2: LSV in motion

3.3 Computer Setup

The computer setup for the project consisted of the following:

- Data transfer was done using a National Instruments NI-PCI-1433 frame grabber card in a PC
- AMD Phenom II 6-core processor
- 4GB of RAM.

The NI-PCI-1433 card is efficient for high data throughput for image application and worked well during the experiment trials at the laboratory. A laptop equipped with Windows XP was used to wirelessly access the embedded computer on the Rover. This is where the measurement data, Laser Speckle imagery was viewed and analysed. The laptop located on Bridget as shown in figure 3.3 was responsible for wheel encoder readings and other Bridget system functions. This laptop was not used in the measurement or encoder data collection analysis.

Figure 3.3: Rover Action

3.4 Bridget Rover

Bridget is a Rover built by Astrium used as a test bed for research or prototype purposes. Part of its build is made up of Power subsystem module and Control system. At this instance the power subsystem would be responsible for distributing power to diverse subsystems and modules on the vehicle. These powered modules could be payload, (LSV mounted at the front of the Rover), cameras and the computer system, on-board.

For the motion of the vehicle, Bridget relies on six wheels and these are independently controlled. These wheels are made of aluminium for weight, strength and budget purposes. The whole structure of the Rover is also made of aluminium for the mentioned reason. Please see figure 3.4 for the different subsystems and figure 3.5 as the rover prepares to take measurements:

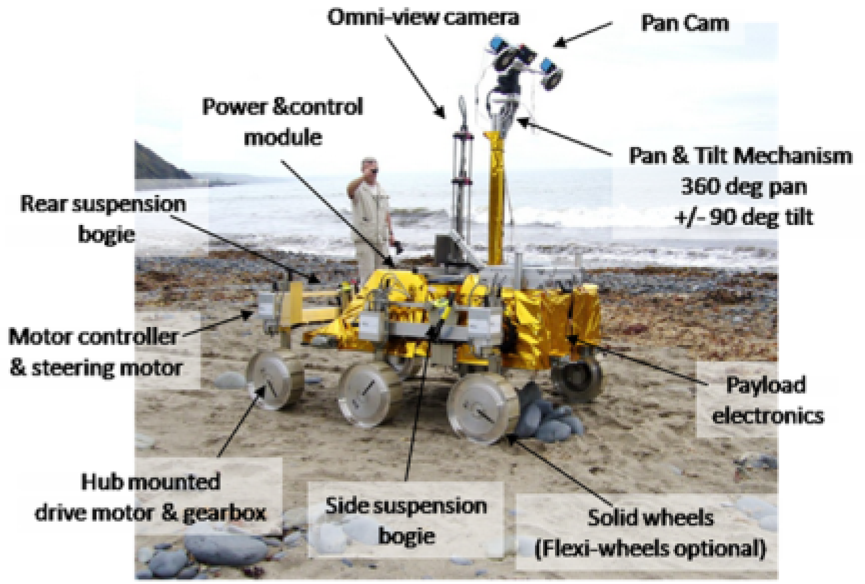


Figure 3.4: Bridget Overview

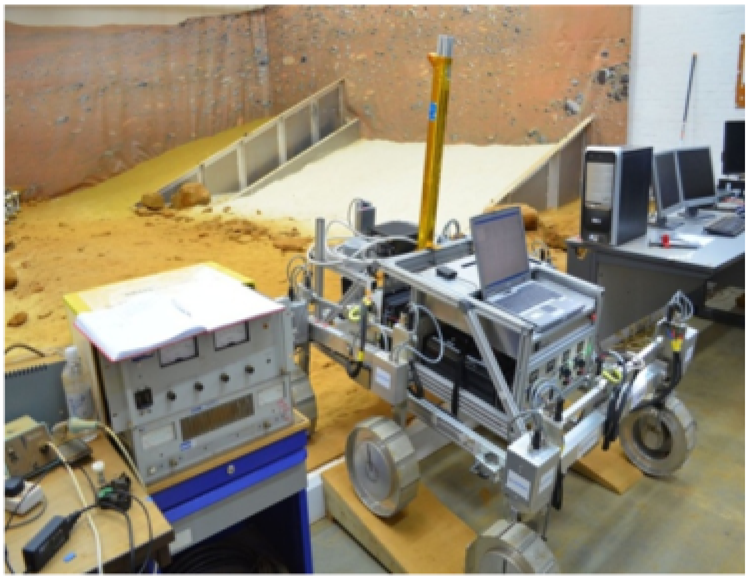


Figure 3.5: Bridget in motion

3.4.1 LSV Measurement Data

The measurement data was collected at Astrium using Bridget attached with the LSV system. The data is kept and maintained by the photonics engineering group and this section is mainly from the engineering photonics group. Please see figure 3.6 as Bridget is taking position measurements

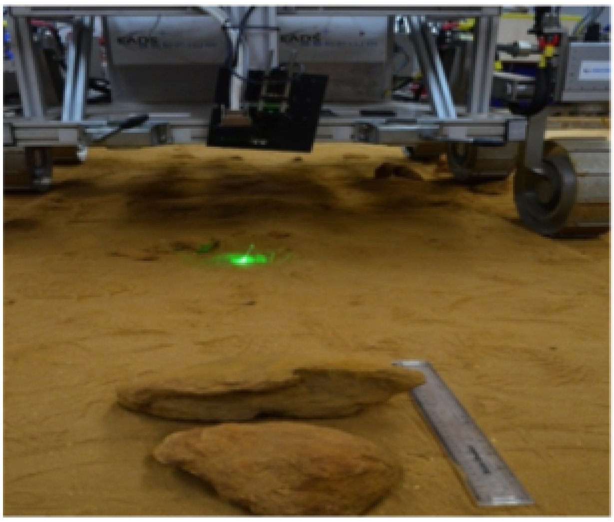


Figure 3.6: Rover Measurements

The laser speckle data collected onsite is shown in table 3.1. This is the partial data which was later simulated on MATLAB.

Table 3.1: Laser Speckle Data

dy	dx	Q
0.0576	-0.0609	5.476
0.0709	-0.0929	5.4455
0.042	-0.0575	5.4327
0.052	-0.0583	5.4496
0.036	-0.0445	5.4273
0.0464	-0.031	5.4226
0.0324	-0.04	5.4336
0.0466	-0.0186	5.3935
0.0206	-0.0058	5.4071
0.0268	0.02	5.3919
0.0165	0.0264	5.4284

The data from table 3.1 shows the displacement dy and, dx of the measured pixel shift between two frames. q is known as the correlation value and is the ratio of the correlation peak to mean level. So the translation measurement from accumulating images can be defined as the q-factor. Furthermore, the value of these accumulating images is found by integrating the translation measurement so that it is possible to produce the data for the plot. The motion of the Rover was simply moving forward (straight) and backward for approximately 4 metres. Later the Rover was made to turn and also drove it under sandy soil to get diverse measurements.

The next diagram (figure 3.7) is the presentation of the position measurement (x and y) converted from raw data collected by Bridget, MATLAB was used to produce the position measurement data.

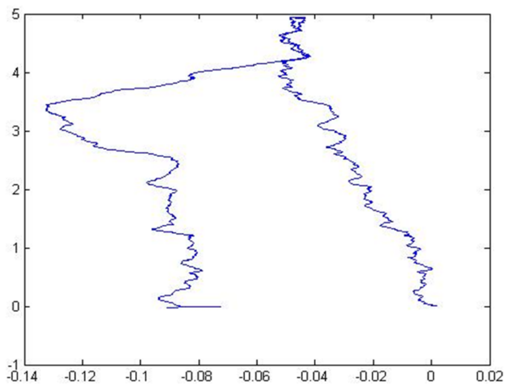


Figure 3.7: Bridget LSV Data

3.4.2 Field Trial Speckle Data

The speckle data was collected and later fed to a Kalman Filter using MATLAB. Vx and Vy were measured in m s-1 since these are of velocity components and these were plotted against time as shown in figure 3.8. Looking at the plotted data from the figure an estimation of the standard deviation is 3mm. The spikes on the figure can be ignored as they do not relate to any errors. The analysis of this data is summarised in the kalman filter section of this thesis.

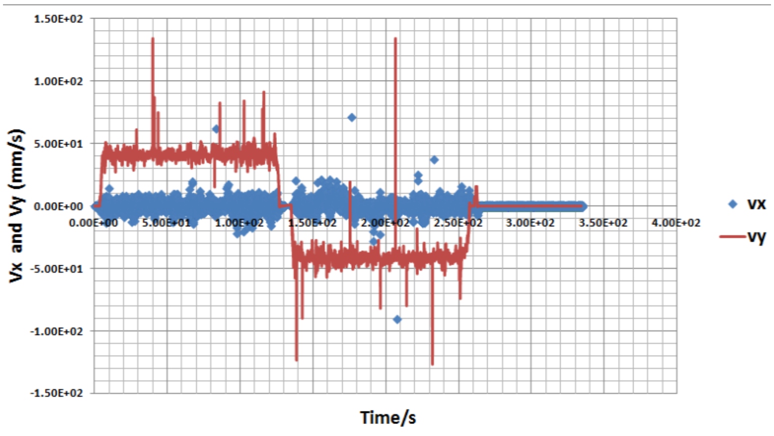


Figure 3.8: LSV Data from field trial

By standard deviation we simply take the values that are common in the data sample from the mean. Figure 3.9 highlights the laser speckle velocity component standard deviation.

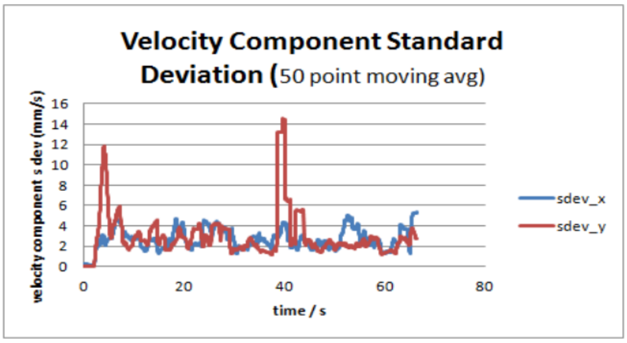


Figure 3.9: Velocity Standard Deviation

The following equations are used for the conversion process from raw data to position measurement:

$$lsv_{xpos} = LSVdata\,(:,2)\,.\frac{\left(\ 14.10^{-6}\ \right)}{1.985} \tag{3.8}$$
$$lsv_{ypos} = LSVdata\,(:,1)\,.\frac{\left(\ 14.10^{-6}\ \right)}{2.000} \tag{3.9}$$

LSV data is the raw data collected by the LSV system, lsv_{xpos} and lsv_{ypos} parameters are the LSV position measurement parameters in the x and y coordinate system (respectively) and is measured in meters.

For the need of not producing larger shifts due to errors on the image size good image overlap was implemented for each shift. Cross-correlation was responsible for the pixel shift.

Errors

- As there is full overlap at every shift the errors should be fairly constant with shift.
- The modelling suggests that the standard deviation in dy and dx is around 0.05 pixels with no camera noise added! i.e. errors dy +-3 , standard deviation = +-0.15 pixels.
- With camera noise added (random noise from a normal distribution, mean=3.0 standard deviation=0.5 counts should be approximately the level of camera noise for our camera) the standard deviation in dy,dx is around 0.07 pixels.i.e. errors dy +-3.standard deviation = +-0.21 pixels.

For working with velocity:

$$V_y = \frac{(d_y 14\mu m)}{S_c dt} \tag{3.10}$$
$$V_x = \frac{(d_x 14\mu m)}{S_c dt} \tag{3.11}$$

Where S_c are the scaling coefficients for the geometry used, dt is the inter frame time = 1/600.0 seconds.

3.5 Two Wheel Kinematic Simulation

With a differential drive robot it is necessary develop a simulation model and later implement sensor integration in order to measure and test the motion or the traversing of a robot in various situations. The environment might be non-obstacle and wheel friendly, unstable soft sand region (wheel slippage) and obstacles bound area which affects motion of the vehicle.

There are many sensors which can be integrated on a robot but for simplicity, focus on this section presents some basic equations to allow the motion of a simple planetary Rover to be simulated. The model can then be used to test various sensors.

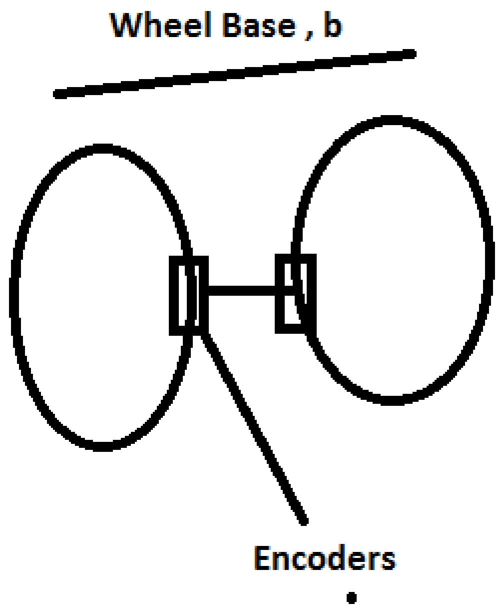


Figure 3.10: Two Wheels

Figure 3.10 outlines the two wheels of the robot: Since each wheel is independently controlled, we can control the turns i.e. turn left or turn right. Some of the sensors that might be implemented or considered are:

- Encoders for measuring the rotation of the wheels
- Position measurement (GNSS receivers)
- Speckle velocimetry to measure the forward speed

Sensors aid in the measurement of Rover motion in diverse circumstances. In the real world, for robotic applications, these sensors usually have noise associated with them. It is necessary to model these noise sources in order to make more improved measurement under simulation and implementation. In the case where error sources increase, utilizing sensor fusion techniques of some of these sensors allows us to make a much improved / better estimation for the trajectory of the Rover or position measurement.

Again MATLAB software was used to simulate the Rover for diverse environment. First an ideal simulation was performed for a Rover with no noise but using constant epsilon to model soft sand, see simulation results in chapter 5. The second simulation involved adding noise sources per specification of the encoder. (Noise immunity for HEDS-9000 encoder = 0.020in taken from Avago Technologies)[63]. The randn function was also used to normalize distribution and compute mean.

The next Rover simulation involved adding laser sensors to measure the velocity components, vx and vy against time. This process involved the placing of sensors in the robot coordinate frame. Initially a single laser sensor was simulated to detect soft sand in which the robot encoders did not know anything about.

A second laser sensor was placed in parallel to the first to see the effect of the measurement. To prevent additive noise to the sensors, the implementation of the sensor was placed considerably away from the encoder sensors. The assumption is that the robot is travelling straight on the Y direction without any turn and the sensors are fixed on the robot coordinate.Figure 3.11 shows the location of the laser speckle sensors

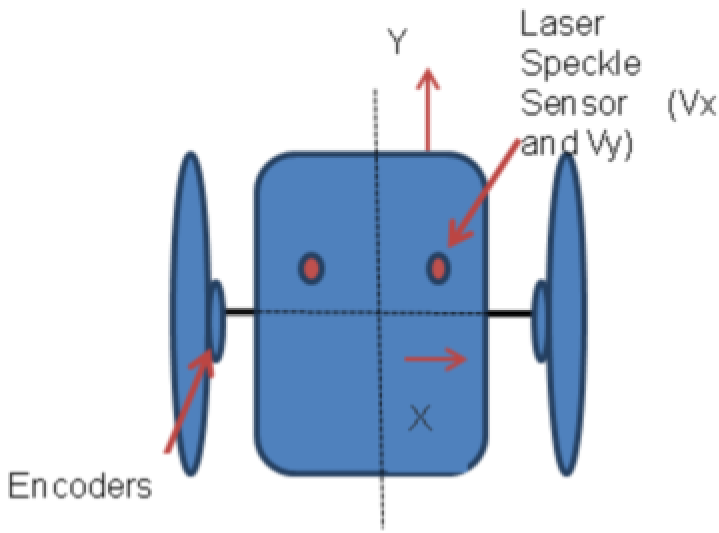


Figure 3.11: Laser Speckle Sensors

Approaches were initially taken to and comprehensively study / implement a simple two wheel robot for the purpose of understanding the kinematic model of the vehicle. We modelled the two wheel robot in hard ground, where the robot did not experience any obstacle and sandy region (loose soil). We analysed the trajectory in time steps and modelled the loose soil by assigning the constant of 0.5 to the kinematics equations for the two wheel robot. Wheel encoders in the model produced expected results when wheel speeds of the wheels were varied

3.5.1 Differential Drive Model

The process when the robot is in motion or is traversing is known as the locomotion of the vehicle and is also associated with the dynamics of the vehicle. The dynamics of the robot may include the following:

- Wheel speed
- Forces acting on robot
- Lower speed(RW) If right wheel slowed the robot turns to the direction of that wheel
- Lower speed (LW) If left wheel slowed the robot turns to the direction of that wheel
- If both wheels maintain same speed, the robots experience a straight trajectory.

Thus any model we wish to develop should include the system model and the above elements for understanding the motion associated with the vehicle. For a differential robot, the system model will have some parameters defined to aid in the development and understanding of a moving two-wheel robot. The variables of interest are the following:

- Radius of a wheel
- Speed of a left wheel (linear and angular)
- Speed of a right wheel(linear and angular)

Figure 3.12 shows the trajectory taken by the two wheel robot:

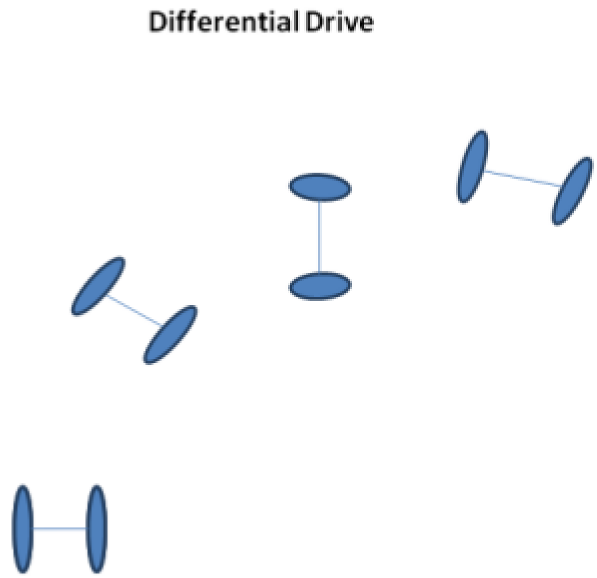


Figure 3.12: Trajectory for wheels

In the real world in relation to robot applications, these sensors usually have noise associated with them. It is necessary to model these noise sources in order to make more improved measurement under simulation and implementation. In the case where error sources increase, utilizing sensor fusion techniques of some of these sensors allows us to make a much improved / better estimation for the trajectory of the Rover or position measurement.

Again MATLAB was used to simulate the Rover for diverse environment. First an ideal simulation was performed for a Rover with no noise but using constant epsilon to model soft sand, see figure 5 above. Epsilon is a way of addressing wheel efficiences or modelling different soil types.In Chapter 5.4, the result analysis gives an overview of the constant involved in the traversing of the rover. The second simulation involved adding noise sources per specification of the encoder. (Noise immunity for encoder = 0.017 from National Instrument). The randn function was also used to normalize distribution and compute mean.

The next Rover simulation involved adding laser sensors to measure the velocity components, Vx and Vy against time. This process involved the placing of sensors in the robot coordinate frame. Initially a single laser sensor was simulated to detect soft sand in which the robot encoders did not know anything about. A second laser sensor was placed in parallel to the first to see the effect of the measurement. To prevent additive noise to the sensors, the implementation of the sensor was placed considerably away from the encoder sensors. The assumption is that the robot is travelling straight on the Y direction without any turn and the sensors are fixed on the robot coordinate frame.

3.5.2 Simulation Results

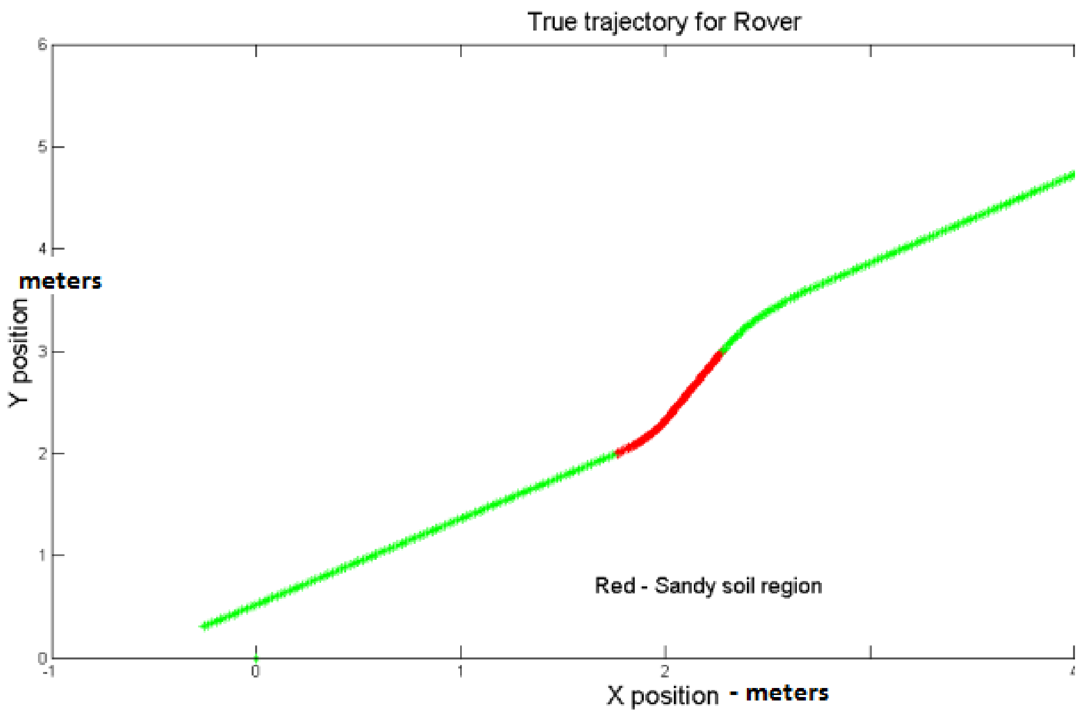


Figure 3.13: Trajectory of the Rover

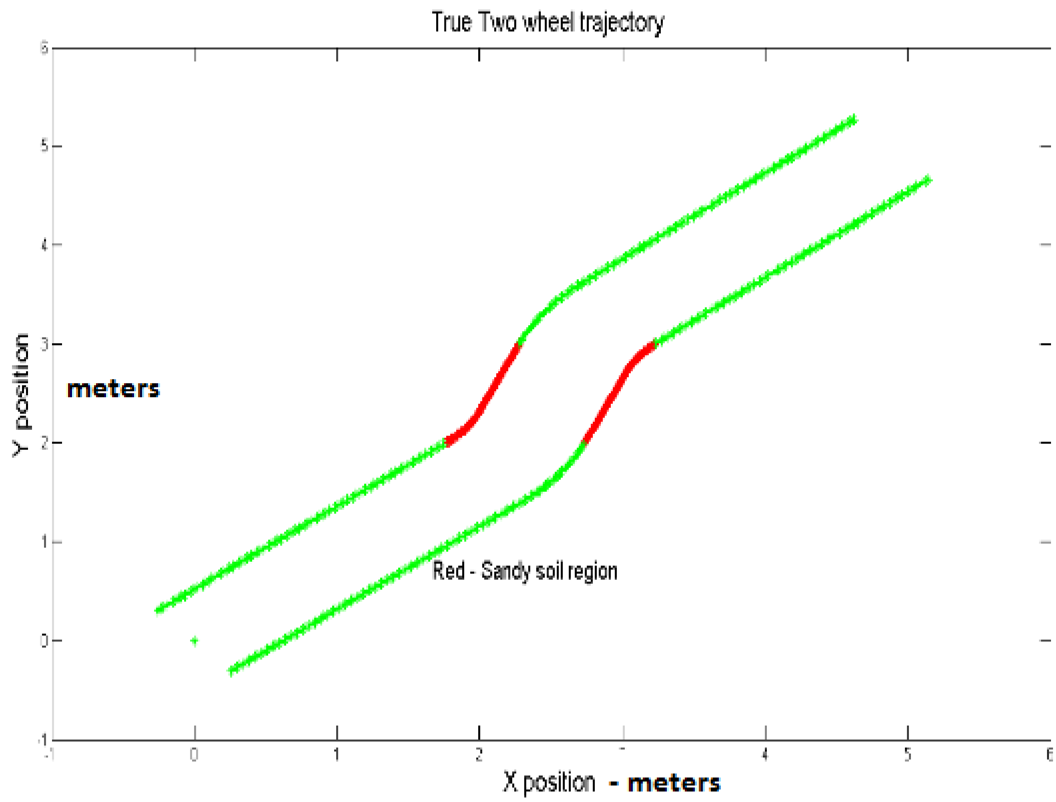


Figure 3.14: Two wheel trajectory

The simulation from MATLAB produced expected results. For ideal conditions, both wheels were influenced by the same angular velocity and the soil was modelled incorporating a constant value,ϵ to the Kinematic equations of motion. Wheel slippage is accounted for using a coefficient ϵ which is 1 if there is no slippage and 0 if the wheel turns without any forward motion. The wheel efficiencies can be a function of time or position, e.g. the efficiency could be set to a lower value such as 0.9 while that wheel is in some defined region to simulate the rover crossing soft sand. As can be seen from figure 3.13, the red outlines the sandy soil while the green outlines the hard ground. The first figure (figure 3.13) is a simple trajectory of the Rover while the second figure (figure 3.14) depicts the trajectory with two wheels. No noise sources have been added to the simulation. In summary we imagine the encoders are perfect and free from adverse external forces and errors.

The KF has produced interesting results in research particularly when simulating position estimation for robots. Near to accurate results are often performed by MATLAB and this enables researchers to understand the various sensors used in the sensor fusion process. Ruslan [59] has an example algorithm which uses the extended kalman filter for a two wheel robot. The outcome of this extended kalman filter is that the filter follows the true trajectory of the robot as opposed to the encoders often accumulating errors.For our simulation purposes we use the LSV data to produce the simulation and input these sensors to the Kalman Filter. The figure 3.15 below shows one of his simulation:

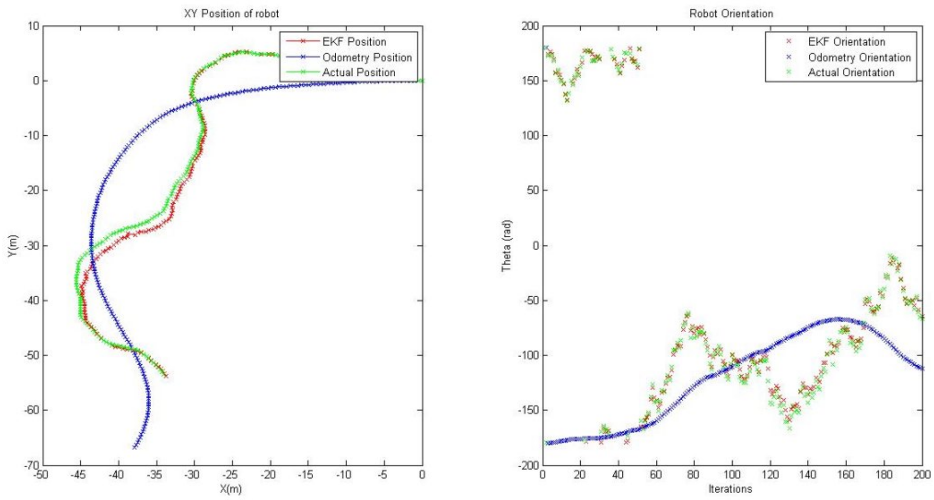


Figure 3.15: Simulation of a robot

Our simulation will differ in that we use a novel sensor (laser speckle velocimetry)but the outcome will be similar. This sensor will go on to detect loose soil therefore have the capability to collect reliable position measurements. Evaluating Ruslan simulations, wheel slippage is not taken into account and the model differs from the one contained in this thesis.

Advantages of Laser Speckle Velocimetry

- The Laser Speckle Velocimetry addresses most of the problems mentioned above, inefficient Odometry and other sensors.
- LSV produces accurate measurements in the presence of adverse terrain (sandy,

rough ground) using speckle imaging correlation.

- Image processing requirements are less than those associated with visual odometry.
- The use of laser speckle means that images can be produced in otherwise featureless terrain e.g. ice, which would cause problems for image correlation based sensors.
- Speckle velocimetry provides high sensitivity measurements, with the measurement resolution determined by the optical configuration and the pixel size.
- Laboratory tests have indicated errors of less than 1 for velocities up to 80 mms-1 translating into minimum errors.
- High speed information turnaround contributes to realtime image processing on a computer.
- Analysis on a sub pixel displacement contributes to the accuracy of the image and the average calculation of the particles from the image is also a major benefit on the image plane.

Chapter 4

Kalman Filter

The Kalman Filter Algorithm is a well-known recursive mathematical formula for estimation process calculations. In the implementation process of sensor fusion we use this algorithm which details some diverse variables of interest to aid us in the estimation problem. The effect of simulation of the inputs or data gathering and the sensor noise distribution (sensor modelling), is that the estimated output information is of greater benefit/useful than the initial input data. Thus, an adequate position estimate measurement should be deduced. An example of the estimation process is shown in figure 4.1

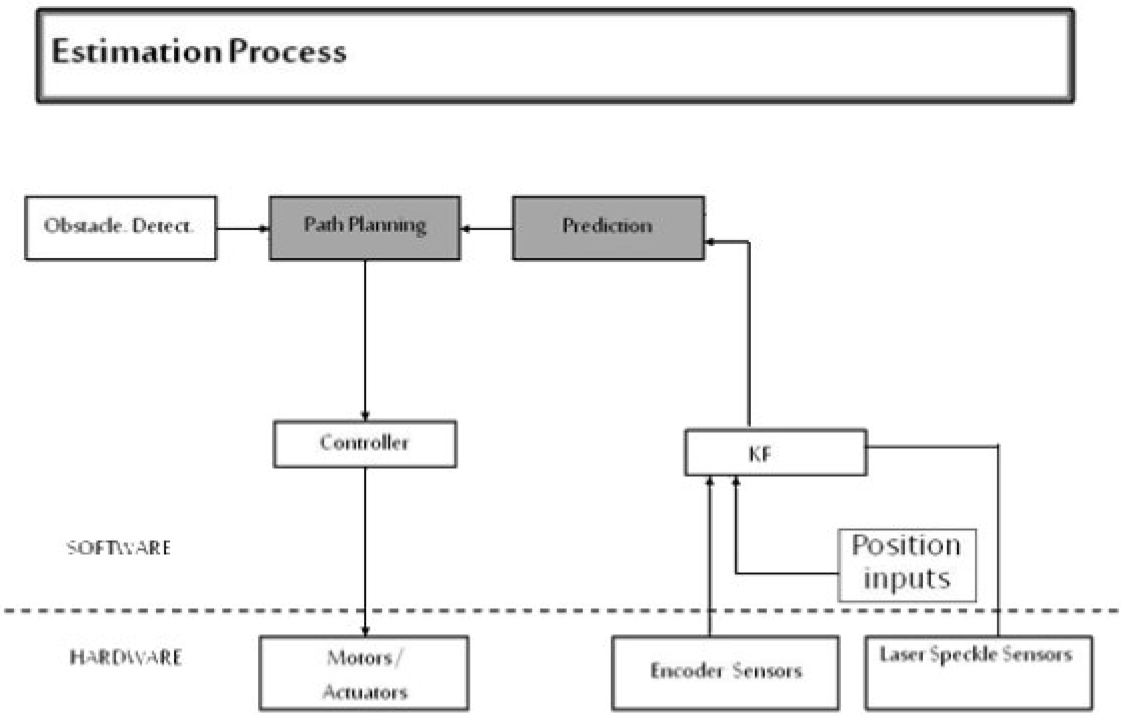


Figure 4.1: Estimation using KF

Interested readers who would like to learn more about this KF optimal estimate algorithm may read Chen [64], Pillai [65], and Ekstrand [66] The algorithm is commonly implemented on MATLAB software for ease of operation and calculation; other diverse software applications can also be used. For our initial acquaintance with the KF we simulate very simple kinematic problems; in our case its the two wheel differential drive model. Assumptions can be made for noise modelling for simulation sake.

4.1 Requirement

Data was collected with the LSV sufficiently but since the optics of the system were not perfect a desire to use an estimation algorithm was considered. The Kalman Filter seemed to be a good option for this.The KF will produce an improvement against the collected LSV and odometry data. More inputs to the KF may be considered for improving the position estimation for the rover. The next section lists some of the parameters that serve as inputs to a KF. Though the filter is simple in theory, it is the noise sources which is part of the algorithm which needs to be accounted for to get a reasonable estimation.

4.2 Algorithms

From implementation of the Kinematic model with Kalman Filter mentioned in Section 3, the initial parameters may consist of the following:

$$\bar{x} = (\dot{\theta}, x, y, \theta, V, \gamma_{LSx1}) \tag{4.1}$$

$\bar{x}$, is the input to the state vector of the Kalman Filter. Further equations maybe deduced for the $\dot{\theta}$ and, v, as follows:

$$V = (\omega_1 + \omega_2).a/2 \tag{4.2}$$
$$\dot{\theta}\ /2 = (\omega_1 + \omega_2) \tag{4.3}$$
$$\dot{\theta}\ /a/b = (\omega_1 + \omega_2) \tag{4.4}$$

For this model, the initial position and orientation of the Rover was set as [0, 0, 40], as a vector representing $[x, y, \theta]$. The angular wheel velocities, ω_1 and ω_2 were initially both set as 0.55 rad/s to test basic straight trajectory of the vehicle. Noise sources were later added to simulate errors for the traverse of the vehicle as follows:

$$\omega_{meas1} = \omega_{1True} + noiseAmp * randomNval; \tag{4.5}$$
$$\omega_{meas2} = \omega_{2True} + noiseAmp * randomNval; \tag{4.6}$$

Here ω_{meas1} represents the angular velocity in radians/sec under noise sources, noiseAmp and randomNval as mean zero and standard deviation of 1. Already two models have been deduced above, one with no noise sources as from equations (4.2) to (4.4) and the other model with noise distribution as in equations (4.5) to (4.6).The state matrix for the initial state space model is implemented taking into account the size of the variable state vector format. Since the equation can be written in this state-space form similar to equation (4.1): The state space equation can be reduced and rewritten as:

$$\bar{x} = A\,\bar{x} \tag{4.7}$$

where the control input variables are set to zero and only utilizing the state variable vector,x. We define the state matrix as follows as an example

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \partial t \\ 0 & 0 & 1 & \partial t & 1 & 0 \\ 0 & 0 & 0 & 1 & \partial t & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.8}$$

Here in equation (4.8), ∂t is the next time step when multiplied with the state vector variables. The next phase is to predict how much error using P which is automatically calculated based on equation (4.7), literature, [67], [68] defines this as the error covariance. The vector for the inputs would look like this:

$$\bar{x} = \begin{pmatrix} \dot{\theta} \\ x \\ y \\ \theta \\ V \\ \gamma_{LSx1} \end{pmatrix} \tag{4.9}$$

In equation 4.9, we have fused the Laser Speckle sensor, $(LSx1)$, forward velocity, γ and the pose information, x, y, θ for the Rover. Summary of the design and implementation for the KF is as follows: [64].

1. We Update the Gain to refine means square error

2. Read new measurement/values.

3. Update the State Estimate using new measurement

4. Use State Estimate,

5. Update the Error Covariance

6. Project ahead. Find state based on system information.

7. Project ahead.

4.3 Speckle Data and KF Simulation

As the field speckle data was collected from the field trial as mentioned in chapter 3.5. The next aim was to simulate this data on a kalman filter. So for our objectives one of the inputs would be denoted as $(LSx1)$, which will be fused with the rest of the inputs to the KF.

The following diagram summarises the simulation / estimation method for testing new sensor / filter models for a planetary rover. With the Speckle Velocimetry sensor (SV) we use the data from the SV experiments (a) to set a realistic level for the sensor noise in the simulation and (b) to set realistic sensor measurement variances (matrix R) to run the Kalman filter on the simulated SV data in the estimator.

To compare the data we subtract one x coordinate from the other (true and estimated) and the do the same for the y coordinates and plot the differences. For this simple filter, we expect a random walk in these differences if only the SV data are used, but there should be no bias. If only the wheel encoders are used for the sensor in the simulator and the estimator then the true and estimated trajectories will diverge whenever there is wheel slip.

For the noise model we make use of the Matlab function normsinv(rand()) to generate a random variable with a Gaussian distribution, zero mean and standard deviation of 1. Multiplying this by the desired noise standard deviation gives us a simulation of realistic measurement noise.

The modified equations from above are :

Given state vector

$$\bar{x}= \begin{pmatrix} x \\ y \\ \theta \\ \dot{\theta} \\ V \end{pmatrix} \tag{4.10}$$

and measurement vector

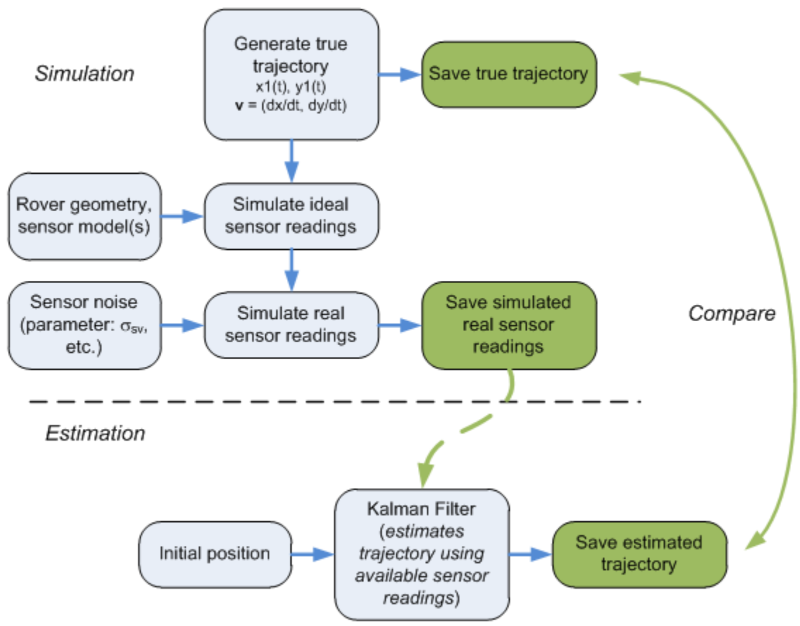


Figure 4.2: Overview of simulation methodology to test rover sensor models and their effects on trajectory estimation

$$y = \begin{pmatrix} vx_1 \\ vy_1 \\ vx_2 \\ vy_2 \end{pmatrix} \tag{4.11}$$

the prediction (M) and measurement model (K) can be written. For sensor positions in rover body axes (p,q) = (p1, q1) and (p2,q2), the measurement model is

$$vx_1 = -q_1\,\dot{\theta} \tag{4.12}$$
$$vy_1 = V + p_1\,\dot{\theta} \tag{4.13}$$
$$vx_2 = -q_2\,\dot{\theta} \tag{4.14}$$
$$vy_2 = V + p_2\,\dot{\theta} \tag{4.15}$$

So

$$K = \begin{bmatrix} 0 & 0 & 0 & -q_1 & 0 \\ 0 & 0 & 0 & p_1 & 0 \\ 0 & 0 & 0 & -q_2 & 0 \\ 0 & 0 & 0 & p_2 & 0 \end{bmatrix} \tag{4.16}$$

The measurement uncertainties are:

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.17}$$

The prediction model is

$$x_{n+1} = x_n + VCos\theta \tag{4.18}$$
$$y_{n+1} = y_n + VSin\theta \tag{4.19}$$
$$\theta_{n+1} = \theta_n + \dot{\theta}\ \partial t \tag{4.20}$$
$$\dot{\theta}_{n+1} = \dot{\theta}_n \tag{4.21}$$
$$V_{n+1} = V_n \tag{4.22}$$

Since the terms sin and cos are included these equations are non-linear and the prediction model is (strictly) therefore a function rather than the linear matrix equation. The extended Kalman filter equations are needed, and these functions are linearised for the calculation of the covariances.

$$M = \begin{bmatrix} 1 & 0 & -VSin\theta & 0 & Cos\theta \\ 0 & 1 & VCos\theta & 0 & Sin\theta \\ 0 & 0 & 1 & \partial t & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.23}$$

The prediction uncertainties also need to be quantified

$$S_{err} = \begin{bmatrix} \partial_x^2 & 0 & 0 & 0 & 0 \\ 0 & \partial_y^2 & 0 & 0 & 0 \\ 0 & 0 & \partial_\theta^2 & 0 & 0 \\ 0 & 0 & 0 & \partial^2\,\dot{\theta} & 0 \\ 0 & 0 & 0 & 0 & \partial_V^2 \end{bmatrix} \tag{4.24}$$

Where x etc. is the prediction uncertainty in each of the state vector components. With these definitions the Kalman filter (an extended Kalman filter using the non-linear model only for the prediction) can be implemented.

4.4 Validation

It is a well-known phenomenon that measurements from sensors come with associated errors or noise and for this reason should not be trusted for accurate measurement on face value. Sensor modelling is a crucial step to understanding uncertainty for a sensory Rover and this may produce efficient measurement for the sensor involved. Noise sources can be analyzed, specific variables or parameters for the noise model can be deduced and acted upon and learning these variables and extending to real data can be achieved. Also adding noise sources to the noise variables on simulation of KF gives us a clear indication of what a model looks like under the influence of noise. A comparison of expected measurement and the true measurement can give us reliable measurement results.

Using a common sensor responsible for measuring tilt or acceleration (accelerometer) as an example. This sensor can measure forces along the x, y, and z direction. Since we know the vibration sensitivity associated with the accelerometer, the dynamics of the sensor will need to be modeled and taken into account during implementation. For the two-wheel Rover, the sensor may have use in the LSV system since we are interested in the vehicle height variation measurement and errors for the LSV. Most literature for sensor fusion of the robot has the accelerometer implemented for the purposes of improving estimation process. The idea behind sensor fusion is to make improvement and noise reduction on sensor systems.

4.4.1 Data Fusion Analysis

The standard deviation was used from the collected LSV data to help in the effect of the outcome of the KF simulation. Varying the standard deviation produced expected results i.e the larger the errors the more the trajectory was affected.

The performance of Kalman filtering has been tested on the basis of two different dynamical models, assuming either a motion with constant velocity or with constant acceleration. The former is expected to better predict the trajectory when the motion is along a straight line, while the latter should work better in case of a winding path.

Validation from using the Kalman filter based on the scenario suggests a trajectory closer to a straight line since a higher level of regularity is imposed. The position errors which were present in the model are reduced after a time period and these are compared to the initial errors which were input to the filter algorithm, validating the whole process of position estimation.

Another important process is the innovation that can be classed as a measure of the uncertainty estimates and observation. So testing the estimator is best done through the innovation process. So the process of validating the performance of a Kalman Filter could be classed as innovation.

Most literatures have the exteroceptive sensors and proprioceptive sensors modelled for robot application.[64],[69],[70], [71], [72]. The LSV is an interesting system, for it consists of laser source and detector optics (camera). In summary the setup may be considered as having an external sensor (laser) and a visual sensor (camera).

4.5 Results

The implementation of the Kalman Filter algorithm for the LSV and the encoder is written in MATLAB in Appendix B.

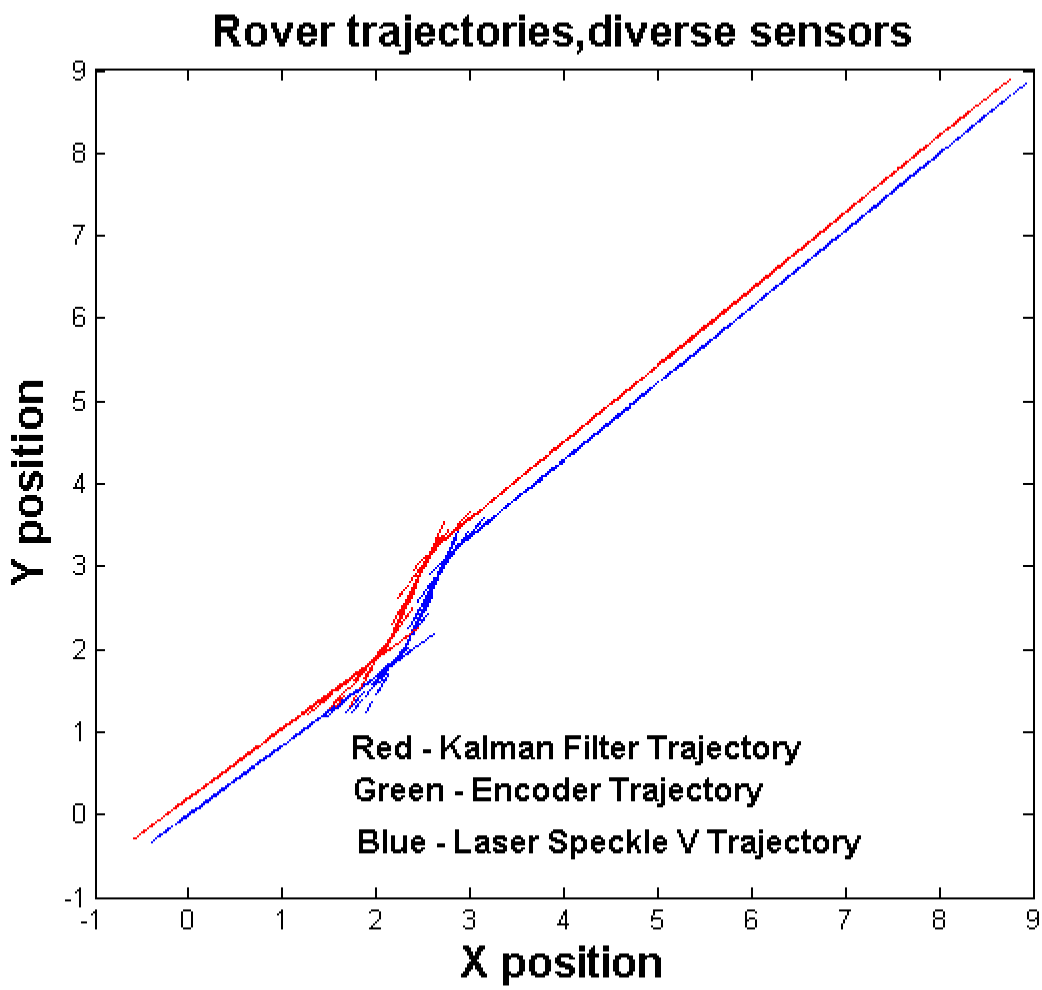


Figure 4.3: Trajectory 1 for Rover

The Kalman Filter gives a good estimate of the trajectory. The simulation was run

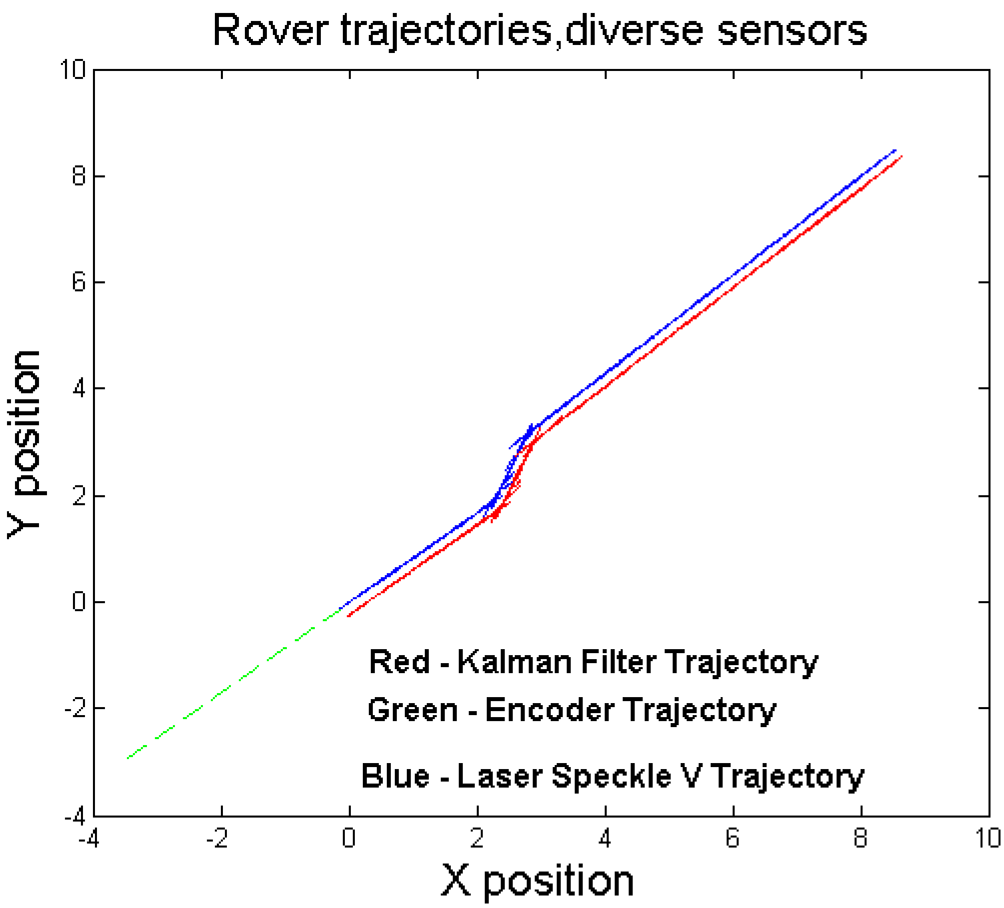


Figure 4.4: Trajectory 2 for Rover

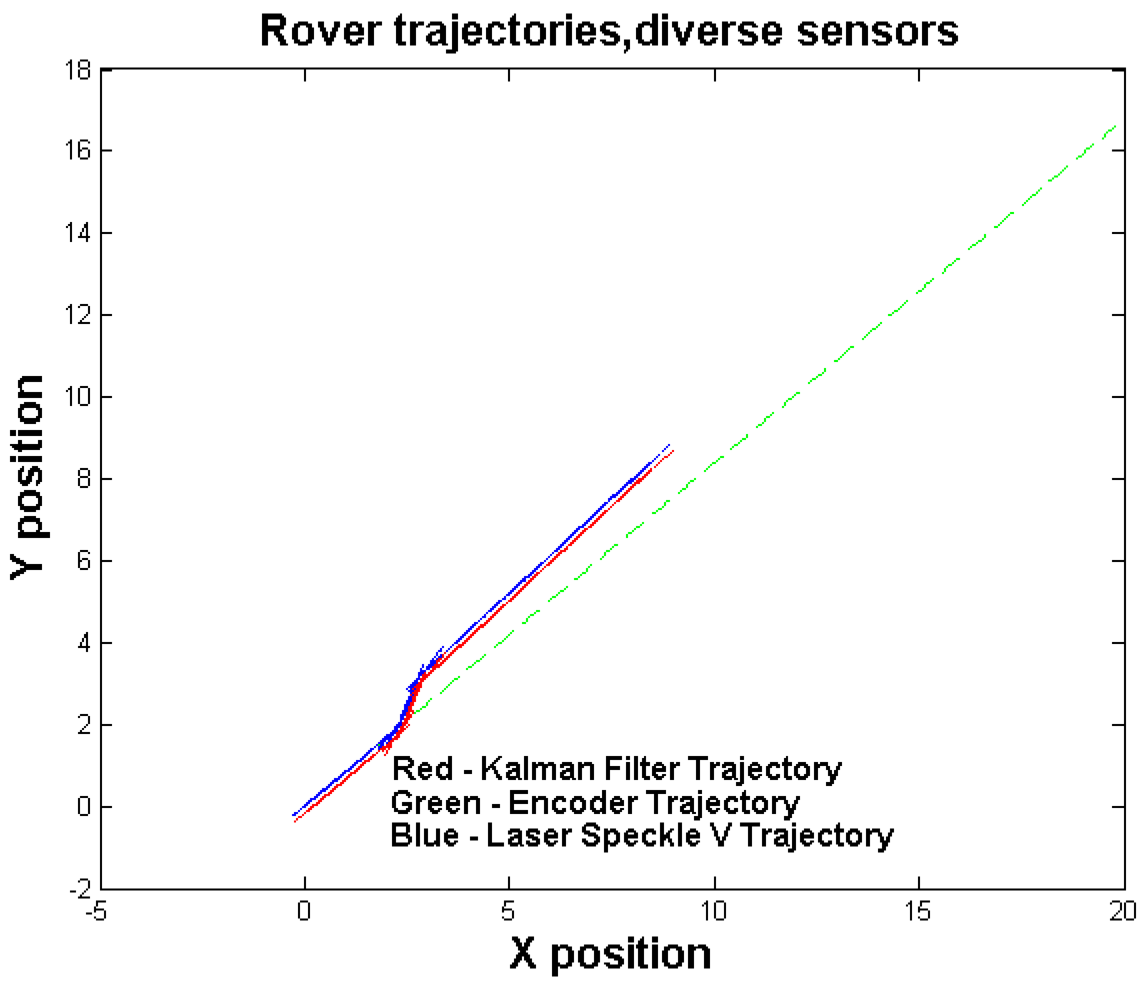


Figure 4.5: Trajectory 3 for Rover

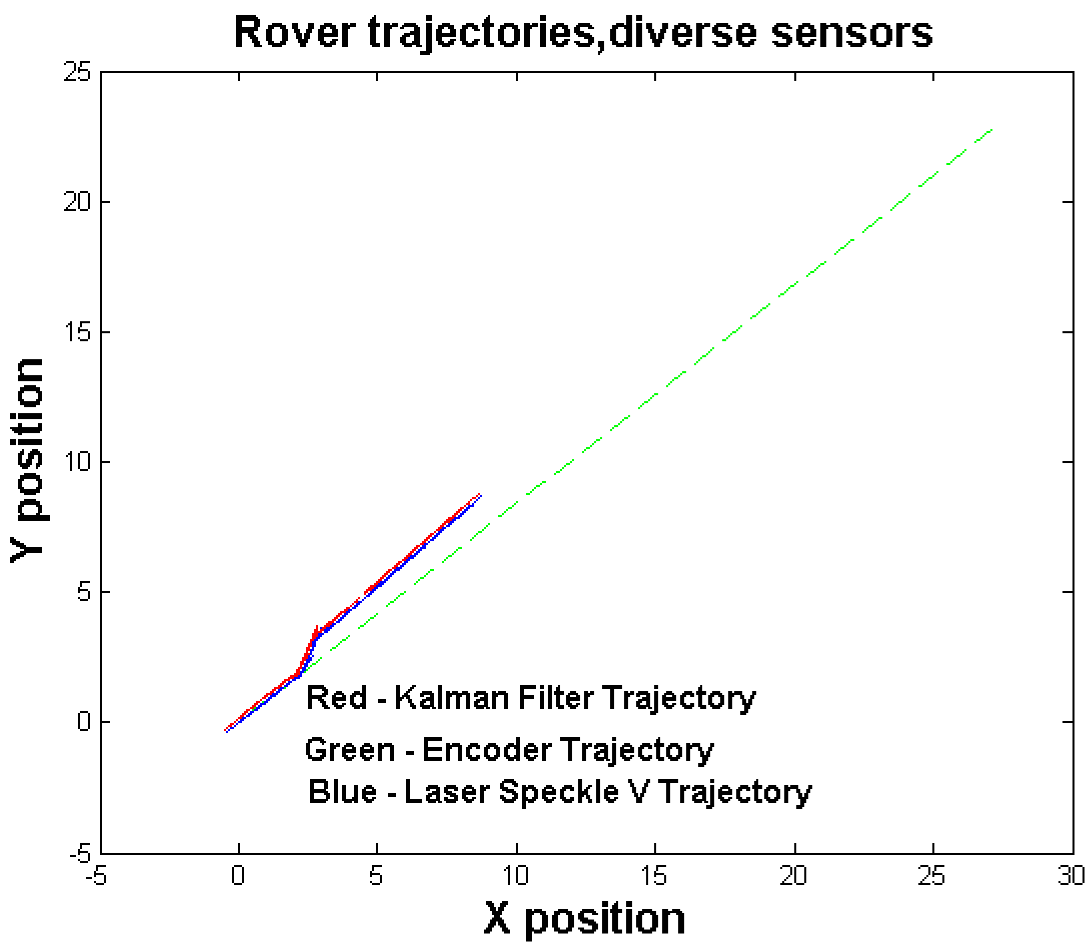


Figure 4.6: Trajectory 4 for Rover

several times on MATLAB which affected the filtered trajectory.The randn function was used and the errors were taken from the LSV data which was collected from the Optics engineering group involved in laser speckle odometry research, an initial standard deviation of 3mm /s was used to test this along the simulations done on MATLAB.

For Trajectory 1 we notice the encoders do not produce the measurements whereas we do see the outcome for the speckle velocimetry trajectory which is the true locomotion of the rover. The Kalman filter estimate is offset to the LSV trajectory.

For Trajectory 2 the encoders trajectory is on a different direction compared to the kalman filter and LSV trajectories proving that the encoder readings can not be trusted.

For Trajectory 3 the encoders trajectory believe the robot is traversing on a straight path when it true sense the robot experiences wheel slippage and the direction is diverted as seen from the y axis between 2 and 3 and thus the kalman filter and LSV trajectories are on a parallel path compared with the encoder trajectory.

So measurements in the KF algorithm, a noise distribution of 0.03 was introduced and these were $sigma_x$ for the x-direction and $sigma_y$ for the y direction. As we ran more simulation different values were added to get different results. In summary the KF trajectory always followed the LSV trajectory.

The two sensors used for the simulation were the LSV and the encoder. Only a single LSV sensor was used and therefore measured vx_1 and vy_1. An accelerometer was added to see the effect during the fusion process but did not influence the outcome of the KF. The following diagram figure 4.7 illustrates the process of the fusion in block chart.

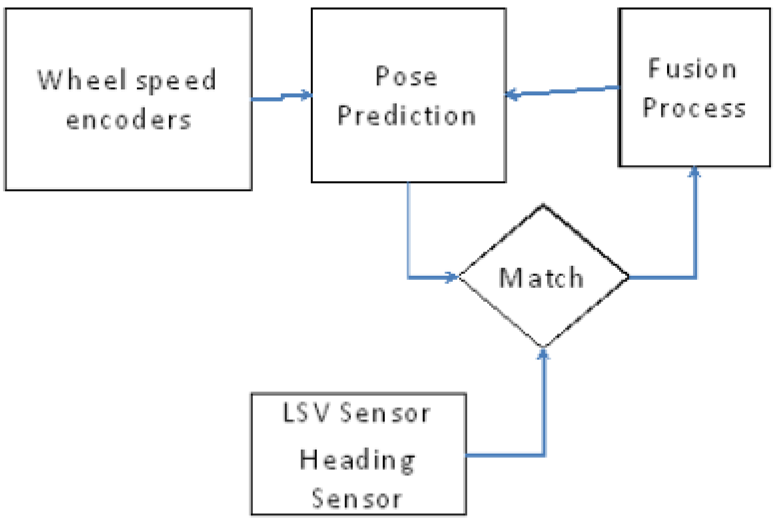


Figure 4.7: Sensor fusion for the Rover

Chapter 5

Results - Two Wheel Simulations

5.1 Perfect Encoders Simulation(w/o Noise)

Figure 5.1: Rover with no encoder noise

For the simulation from figure 5.1 the two wheel robot produced a trajectory dependent on the initial angle of 40 degrees. Again no noise was injected to the simulation and thus the motion of the robot appears to go on a straight line. The encoders at this stage are acting perfectly, free from the influence of noise. Meanwhile for the velocities, (figure 5.2) the LSV optical sensors and encoders where

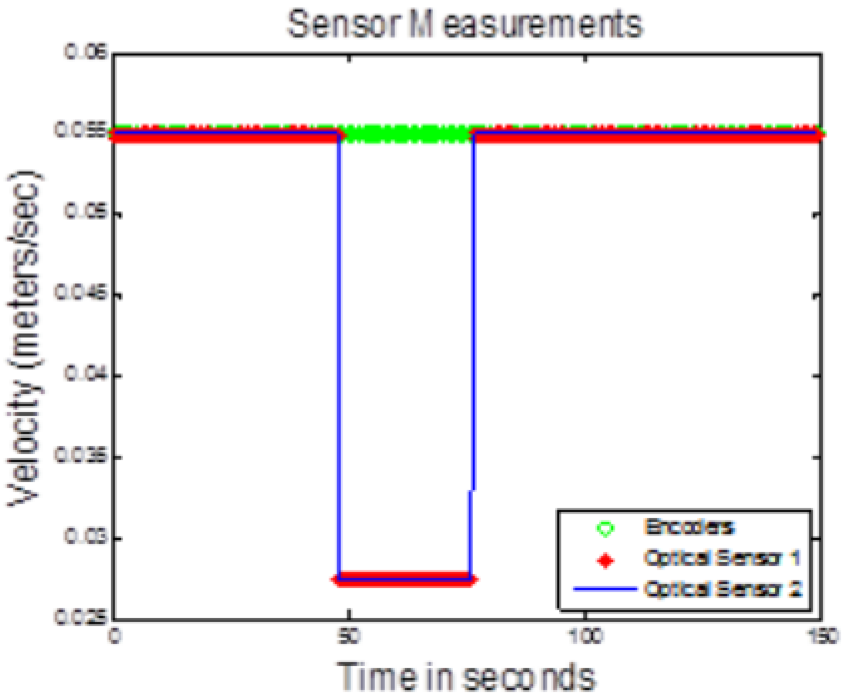


Figure 5.2: Rover with encoders and optical sensors

simulated and produced expected result. The encoders(green line) were unable to detect any wheel slippage and the reading is constant at 0.055m/s, while the optical sensors (LSV sensors) do detect the wheel slippage and gives a reading of 0.0275m/s at the sandy region depicted from figure 5.2.

So in summary, LSV sensors do a good job of detecting slippage while the encoder reading cannot be trusted should the wheel experience obstacles and slippage.

5.2 Noised Encoders Simulation

Figure 5.3: Rover with noisy encoders

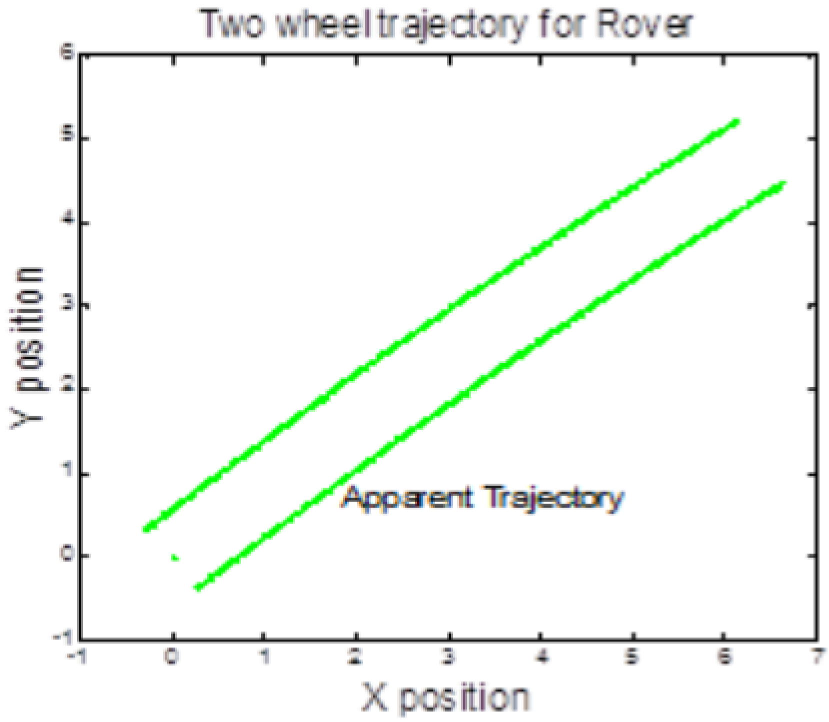


Figure 5.4: Rover with noisy encoders

For this part of the simulation we added noise to the encoder sensors figure 5.3 to see the what effect it will have on trajectory. The above diagram figure 5.4 shows a slight deviation in trajectory depending on the amount of noise amplitude added. Large quantities of noise amplitude was added to the simulation and this altered the trajectory of the two wheel robot and in the simulation the robot appeared to wobble in figure 5.4.

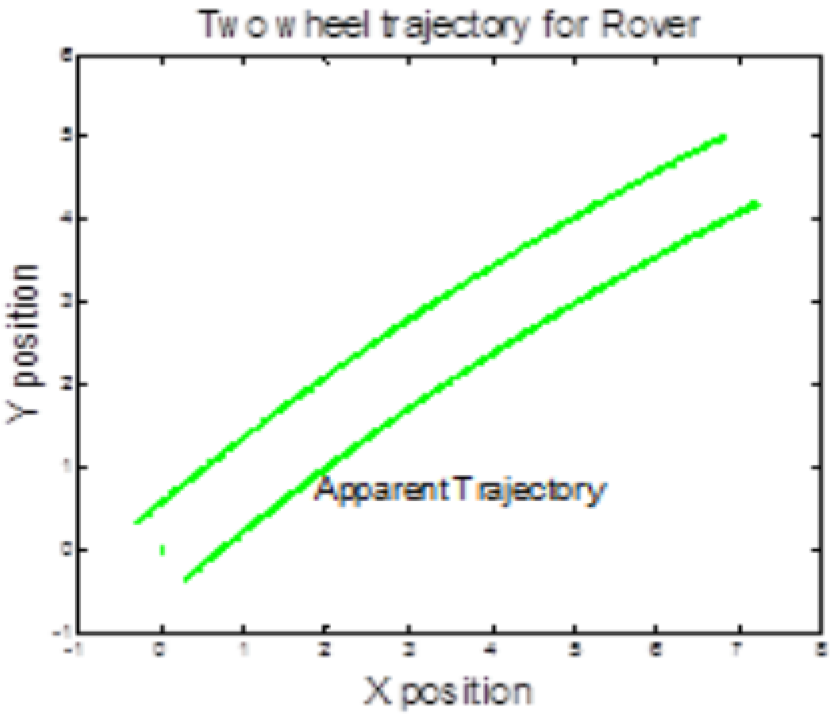


Figure 5.5: Rover with noisy encoders

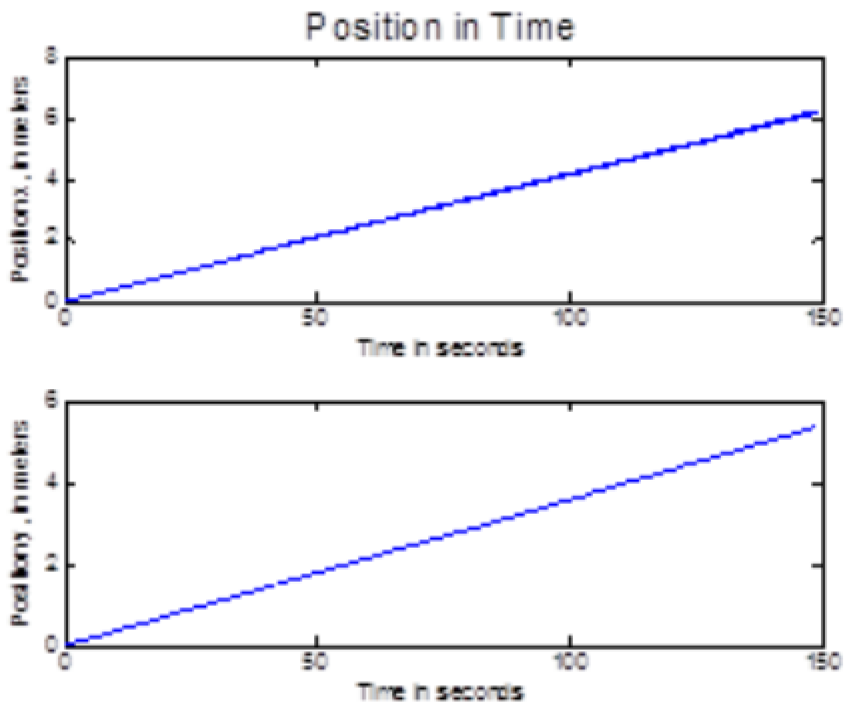


Figure 5.6: Rover position

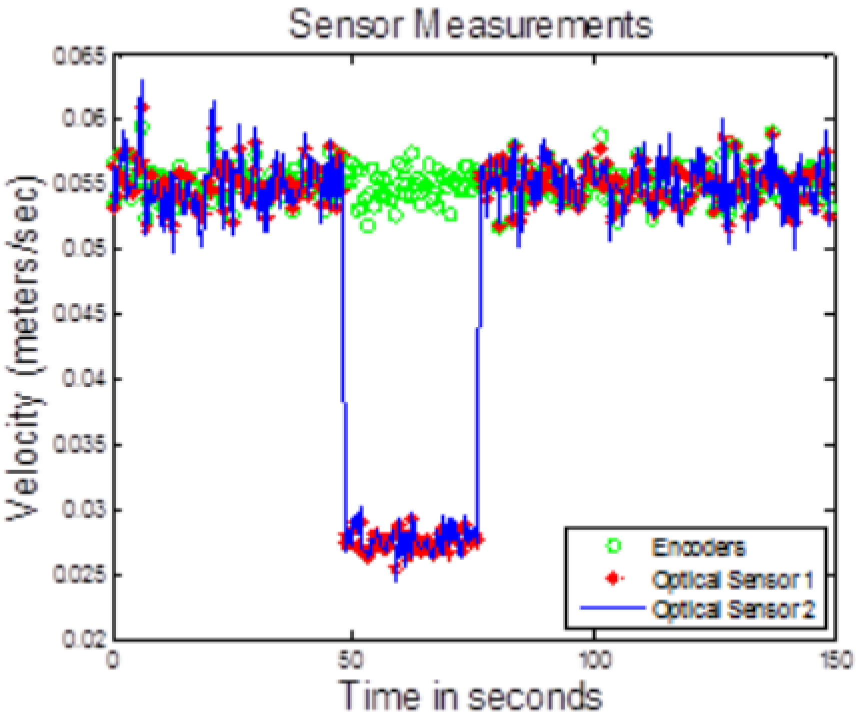


Figure 5.7: Rover with noisy sensors

The three diagrams from figure 5.5 to figure 5.7 were also influenced by noise during simulation. Again, in figure 5.5 the noise amplitude was added to the encoder sensors, figure 5.6 shows the position in time (in seconds) and figure 5.7 shows the velocity measurements. Though noise was added to the optical sensors and encoders, the noise did not alter the detection of wheel slippage using LSV sensors, as well as the encoders was not able to detect slippage. The result was still similar to figure 5.2, which had no noise influence.

Simulations were reiterated and produced a variance of results for the encoders due to the randn() on MATLAB and different trajectory and motion was seen.

5.3 LSV sensor simulation results

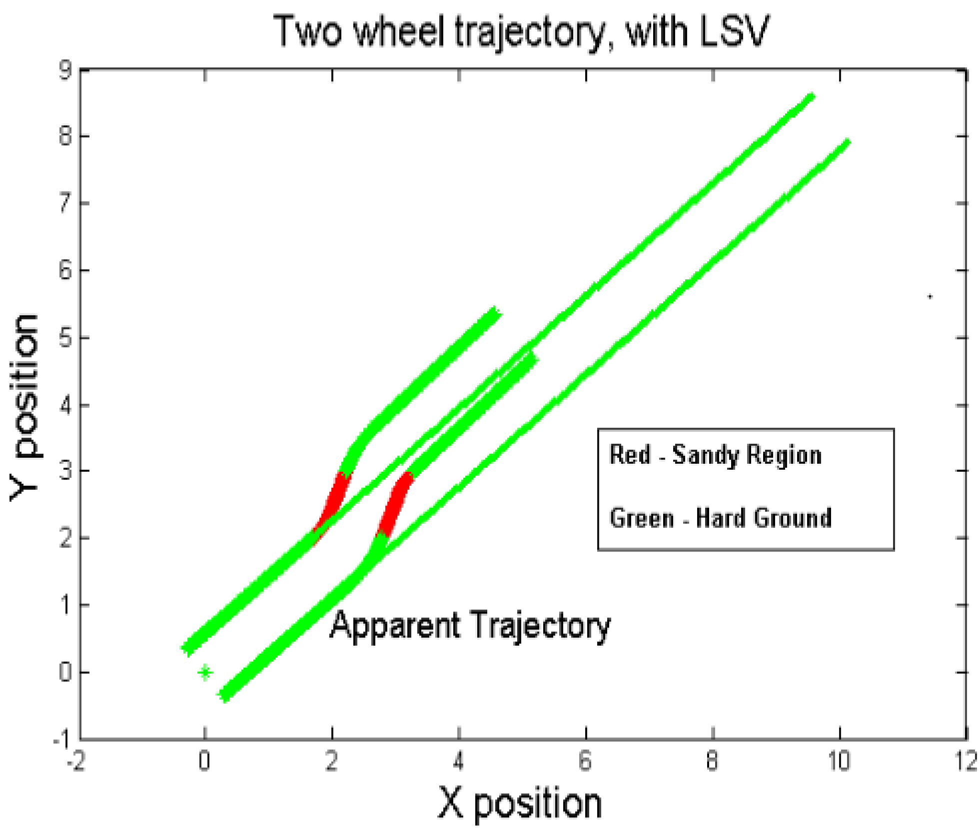
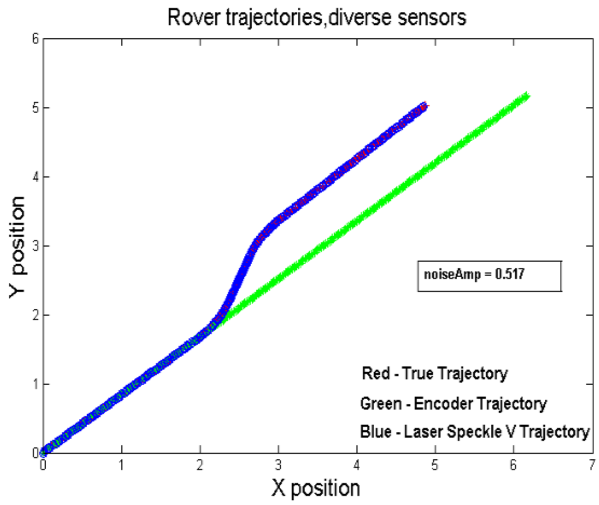
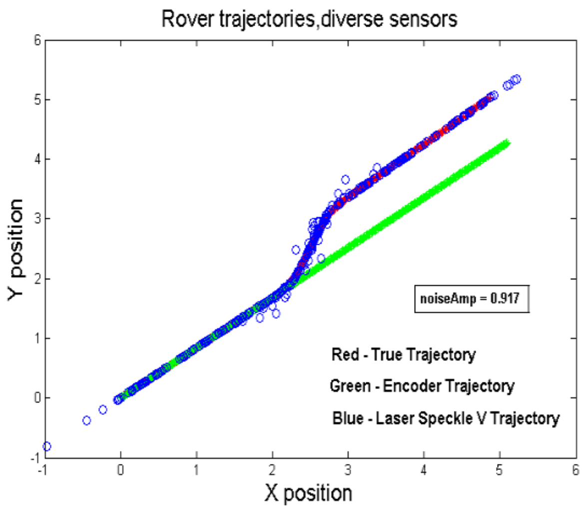


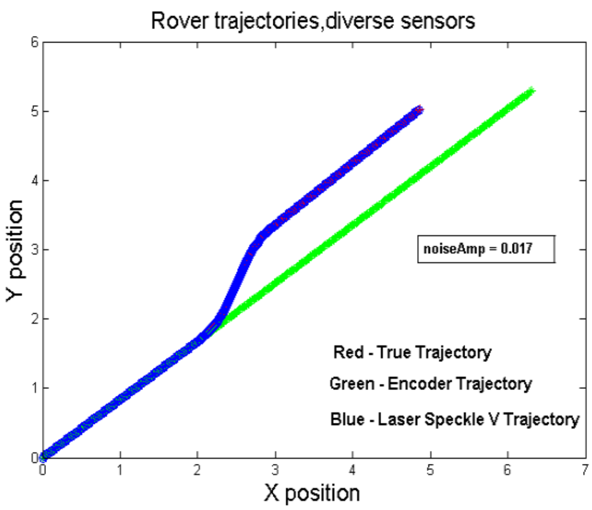
Figure 5.8: Encoder and LSV sensors



(a) LSV Trajectory 1



(b) LSV Trajectory 2



(c) LSV Trajectory 3

Figure 5.9: LSV trajectory noised

For this part of the simulation, since gaining knowledge of the encoders and LSV sensors, we simulated the position estimation of the two-wheel robot. First, in respect to figure 5.9(a) a noisy LSV trajectory, noisy encoder, and true trajectory was simulated. Though the LSV was influenced by noise errors, it followed the true trajectory of which we imagined as a perfect model for a two-wheel robot.

Again in respect to the early discussion concerning the encoders unable to detect slippage, the encoders assumed the robot was going straight. Diagrams from figure 5.9 are simply an iteration of figure 5.8 and getting pretty much the same results with a small deviation between the LSV trajectory and the true trajectory due to the zero mean and standard deviation of one. By iteration, we run the simulation several times and allow the random() from MATLAB to impact the simulation result. Also, noiseAmp reflected on the diagrams is simulated noise which was originally set to 0.017 inches.(readers may see the noiseAmp value from reference Appendix C) This number was varied to the reflected diagrams to show the effect it has on the trajectories.

5.4 Result Analysis

By performing simulations we address various challenges associated with the navigation of the rover. Results from the simulation prove navigation is improved through the use of a kalman filter when data from different sensors are fused together. The LSV data simulated enables further study of optic data and gives a good idea of how speckle imaging optics relates to the motion of the rover.

The influence of errors developed from either surface roughness, rocks, sandy terrain and other obstacles was simply modelled as a constant epsilon , ϵ and assumed at specific values. For the simulations above we simply assumed the sandy soil at less than 1 and set the value for the hard ground at 1. So in theory this is what the values of soil types may look like:

- Soil A - 1 (hard ground /no slippage)
- Soil B - 0.8
- Soil C - 0.6
- Soil D - 0.5
- Soil E - 0.4

$$V = (\omega_1 \epsilon_1 + \omega_2 \epsilon_2).a/2 \tag{5.1}$$

$$\dot{\theta}\ /2 = (\omega_1 \epsilon_1 + \omega_2 \epsilon_2) \tag{5.2}$$

$$\dot{\theta}\ /a/b = (\omega_1 \epsilon_1 + \omega_2 \epsilon_2) \tag{5.3}$$

The equation 5.1 is the forward velocity of the mobile platform, if any of the ϵ values are modelled under the soil values that will affect the speed of the rover. Wheel slippage is accounted for using a coefficient ϵ which is 1 if there is no slippage and 0 if the wheel turns without any forward motion. The wheel efficiences can be a function of time or position, e.g. the efficiency could be set to a lower value such as 0.9 while that wheel is in some defined region to simulate the rover crossing soft sand.

5.4.1 MATLAB Analysis

So using Soil E as an example, the soil is more loose and sandy than Soil B. The rover navigating in this region would easily divert direction because of the loose soil. For the MATLAB code presented in Appendix C eps1 and eps2 are the constants for the two wheels as they enter different soil types. It only takes only a single wheel to divert the trajectory if it is in the sandy soil region.

The performance of the MATLAB code for the two wheel drive (Appendix D)is dependant on a number of factors:

- Wheel size
- Wheel axle size/length
- Wheel speed
- Calculation of velocity
- Calculation of orientation

The manipulation of the above parameters easily affects the outcome of the MATLAB simulation and thus affects the performance. Using the wheel speed as an example and reference, if any of the wheels from the rover are not the same speed then the robot will turn and not continue to travel in a straight line or trajectory. This is the case with simulation figure 5.5. Therefore, for a straight trajectory the two wheels will need to have equal wheel speeds and assume that no obstacles will hinder the trajectory along the way. Below is a snippet of the MATLAB code presented in Appendix D.

```
omega1 = speedwheels(i-1,1); % Speed of wheel1
omega2 = speedwheels(i-1,2); % Speed of wheel2
```

The above code shows the omega variables that hold the speeds for both wheels.

Chapter 6

Discussion

Modelling the System

In previous sections we have implemented and simulated the two-wheel Rover with associated sensors including using the KF. The modelling was not explained in detail and for this section we attempt to deduce the ideas behind the system modelling particularly addressing the uncertainty of the locomotion of the vehicle. Figure 6.1 is a diagram showing the hierarchy of modelling a system.

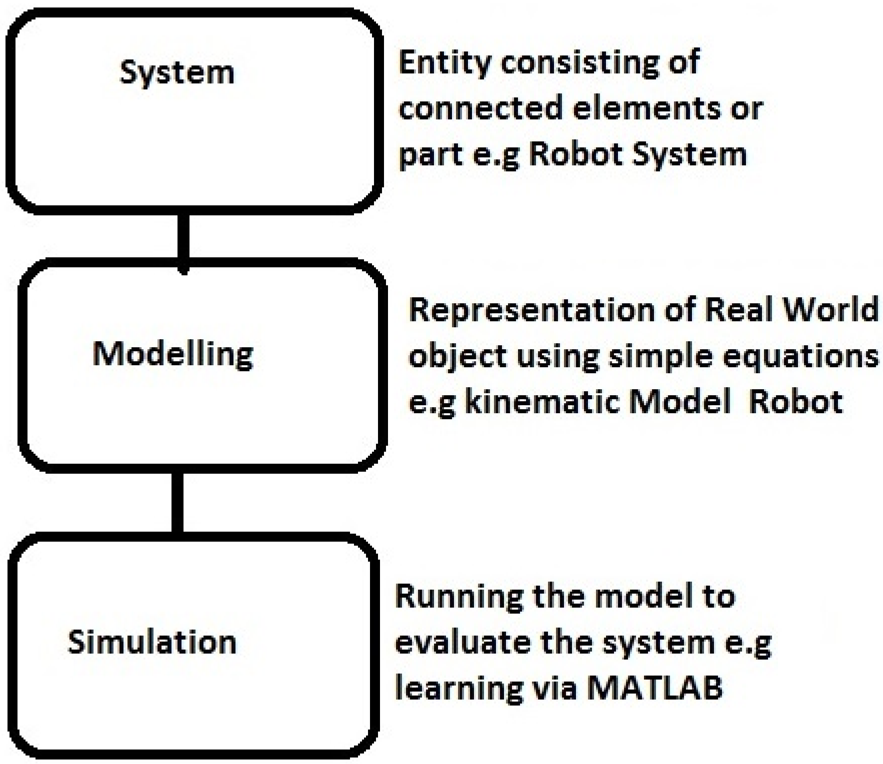


Figure 6.1: Rover Modelling

From a top level hierarchy, the system with its interconnected entities needs to be simplified down to a level where it can be understood. In our case the two-wheel Rover, will need the dynamics thereof to be learned and simplified. So part of the analysis may include kinematics or locomotion of the vehicle, the sensing capability and understanding of the environment. Furthermore to our endeavour we model the two-wheel Rover such that the position measurement may be realized.

6.1 Measurement Models

For a robot to make meaningful measurements the wheel encoder sensors are usually implemented to measure the wheel speed. At this instance the mobility of the vehicle is aided with motors. The encoders and the dynamics of the vehicle will give measurement in time. This is a common measurement model usually implemented for a traversing robot.

Though the measurements are made by the encoders, they often are corrupted by errors which might be attributed to (i)unequal wheel sizes of the rover (ii) incorrect axle size and these can be either system or non-system contained. Through the calculation of the kinematic equations the axle and wheels radius affect the trajectory and performance of the rover. Therefore these measurements will need to be rectified to give us some reliable sensor readings. Appendix D is a MATLAB simulation that gives information about the wheel radius and axle, if we change the defined values (wheel radius and wheel axel length) then that affects the motion of the rover.

6.2 Ideal Case Model

Initial modelling consisted of developing a kinematic model which produced a truth trajectory of the vehicle. This involved setting the wheels to the same velocities and not adding any external disturbances or noise. We imagine the system dynamics of the two-wheel robot is functionally perfect. Noise and errors will be accounted once we have a full understanding under the ideal case. In MATLAB (see code under Appendix) we simply set the velocity to 0.55 m/s for both wheels. Also the sandy soil or loose sand was modelled using a constant, c = 0 for soft sand.

The modelling process for a traversing rover is based on the assumption that the wheels are in motion on a steady ground(hard ground). It is also assumed that position, pose and orientation is known before hand. Since it is known that wheel slippage can occur while the rover is in motion, we define this slippage with a constant less than 1 and in this case we simply chose 0.5 for simulation practice.Summary of constant coefficient, c, used:

- No Wheel Slippage , c = 1, Hard ground
- Wheel Slippage , c = 0, Soft sand

So when the wheels of the rover are crossing the soft sand region from hard ground, c could be 0.9.The constant, c, can be considered as wheel efficiency. So in our ideal case scenario we would hope there are no slippage and the wheels continue to move in a forward direction without any deviation. So the forward velocity from equation 5.1 would be expanded to the following:

$$V = (c_1\omega_1 + c_2\omega_2).a/2 \tag{6.1}$$

The wheel radius is a and the distance between the two wheels or robot wheelbase is b. The wheel efficiencies can be a function of time or position, e.g. the efficiency could be set to a lower value such as 0.9 while that wheel is in some defined region to simulate the rover crossing soft sand. The forward speed v depends on the average wheel speed (and any wheel slippage), and its components along the x and y axes give the speed components.

If any of the wheels was in motion on soft sand the velocity of the robot would be halved. The constant value was assumed for simulation purposes, while in the real world soil properties would need to be accounted for. Simulation results produce a trajectory of the vehicle going in a straight line depending on the orientation of the vehicle and slight change of trajectory influenced by the sandy soil. Probably a good reason for using the ideal case is that the model can be used as a reference with noisy diverse sensors for the Rover and then position estimation of the Rover can be deduced.

6.3 System Error Model

Martinelli[40] gives an account of two error sources for the navigation of robot, systematic and non-systematic errors. The robot is equipped with laser range finder, an exteroceptive sensor and encoders, an proprioceptive sensor. He develops equations for the purpose of estimating errors using augmenting KF processing. Though his implementation involves the extended KF, in our case for the application of systematic errors, the odometry error model can be tailored for the provision of LSV / odometry project. In this case, for developing the system error model we use the linear KF.

Meanwhile Bradley [73] discusses the alternative way of mitigating errors from odometric sensors. One of the proposed solutions is driving the robot in such a way that the error measurement from encoders do not cancel out with errors from the dynamic system. The other method for dealing with odometric errors is linearization of odometry calibration. The linearization of the odometry calibration has produced positive results and knowledge of this information enables the characterisation for the noise model for the KF [74]. Since mean square errors have been enormously eliminated (within 0.1 from true reading), this further enables the sensor fusion process with ease for the odometric process. Errors associated with LSV also need a model similar to the odometry calibration, to produce consistent results after sensor fusion. Error sources are likely to come from the laser optics and detector optics. These errors can be classed as system errors since they are system contained and can be referenced with non system errors derived from the environment.

Wang Junli[75] proposes the Expectation Maximization Method for the purpose of recognising noise coefficients in laser speckle images. In his paper he derives the image model using equations involving noise speckle image intensity. Interested readers may refer to his discussion for the equations involved for modelling speckle images. The ability with his approach to discriminate noise from white noise and the signal makes it favourable for modelling under the KF process. The other advantage to the image modelling technique is the speckle statistics learning is simplified and provides an opportunity to detect and filter noise properties.

For evaluating the process model, the state vector inputs will need information such as position and velocity of the object. Velocity information can be gathered through the measurement process.

6.4 Measurement Noise Model

Sensors which come with noise are defined in the measurement noise model. Using the KF as a basis, R is the matrix term for the measurement noise.

In dealing with the displacement noise (relative), we automatically assume that these are affected by a process which is random by nature and thereby also affecting the noise vector outcome. This noise vector may be defined as having a zero mean and is Gaussian. The noise is usually independently associated. One way of producing the measurement noise matrices is through the estimation process of an object. Usually the measurement noise matrices define the data corrupted by noise from the sensed object through the use of measurements. The noise source could be in the form of vibrations, humidity, temperatures and others, depending in some cases on application.

The process of validating sensor data is an important one especially for the use of performance and efficiency of the sensors invovled. In regards to temperature and humidity and as an example application Castello [76] in his paper, addresses the use of filters (e.g KF) for the purposes of improving sensor data for building technologies.In this instance inaccurate data is rectified through modelling analysis. Other potential applications is in agriculture and weather forecasting [77].

For our robotic application, the process for validating sensor data would be similar to Castello's. Sensors might be affected by inaccurate data due to defects that originally come with the sensors. In a planet like Mars temperatures are extreme and this may affect the functionality of the sensors on a planetary rover. Modelling sensors for such conditions would enable the KF to output near satisfactory results. This data over a time period would be modelled and observed.

The usual set-up of implementing the noise model is by developing the outcome of a random number generator, and comparing this with sensor readings by performing diverse experiments. It is easy to tailor the matrices from the truth model.

The measurement error covariance matrix is a useful term for the interpretation of errors in corrupted data. There are diverse algorithms in research which aid the limitation of accumulation of errors or noise. Zhao[78] gives an account of a robust algorithm for the purpose of measuring noise properties. The KF does a good job of converging sensor data by filtering of noise. It may not converge initially but will take some time to achieve the goal of dealing with the noise associated with the sensor. Figure 6.2 is an example of measurement noise convergence data.

Sensor noise (on the measured speckle pattern shift in pixels) will depend upon a larger number of factors including:

Image size - The size of the images used in computing the cross-correlation will influence the accuracy the peak can be located to. More pixels (larger images) will lead to less influence in camera noise in the correlation peak (a smoother correlation peak) and hence a better accuracy. However this is at the expense of processing speed.

Speckle size The average size of the speckles on the detector will vary due to beam spot size and detector working distance. Different speckle sizes will lead to different correlation peak widths and hence different accuracies in determining the peak centre and shift.

Speckle blur due to Rover velocity and image exposure time When the rover is moving quickly the speckles can appear to be blurred in the direction of motion due to the non-zero camera integration time. This increases the effective speckle size and hence influences the accuracy to which the peak can be found.

Camera noise The level of random noise on each camera pixel will influence the accuracy to which the peak can be found, higher noise will give a noisier correlation, however typically we are averaging over 1000+ pixels so this noise is not as critical as it may seem, until you get to very small images sizes. Camera pixel noise is also complicated for this camera as it is a CMOS sensor each pixel has individual noise characteristics (in contrast to a CCD camera where all pixels share a readout circuit.). A typical pixel will have a mean background level of approximately 3 counts and standard deviation of 0.44 pixels.[79]

As can be see the actual level of noise will depend upon the data set and how it was processed, but I think it is safe to assume a noise standard deviation of 0.1 pixels or less in the pixel shift values. This has been determined by some modelling of speckle patterns that I has been performed.

This would translate to a velocity error of 0.7 mm/s using standard error propagation for a pixel size of 14m:

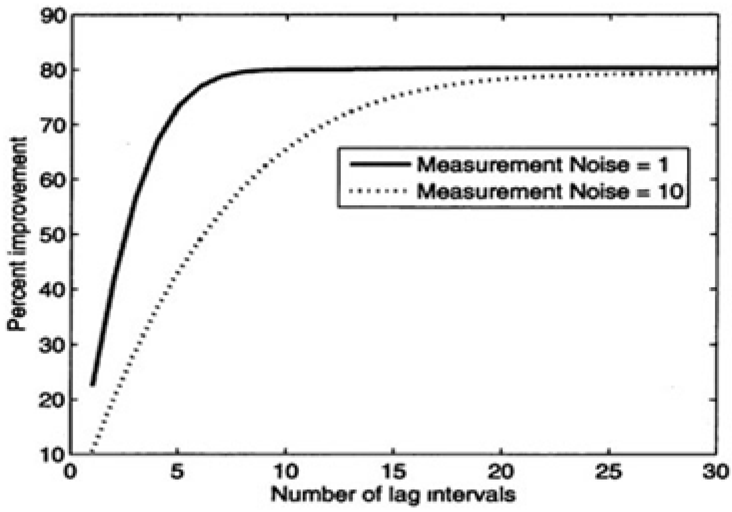


Figure 6.2: Measurement noise estimation, [5]

$$
\begin{aligned}
V_x &= \Delta x * 14\mu m)/(Cxx * \Delta T) && (6.2)\\
\sigma\Delta_x^2 &= (\delta V/\delta\Delta x)^2 * \sigma\Delta_x^2 && (6.3)
\end{aligned}
$$

Hence

$$\sigma v_x = (14\mu m/Cxx * \Delta T) * \sigma\Delta_x \quad (6.4)$$

Where $\sigma\Delta_x$ is the standard deviation in the measured pixel shift, σv_x is the standard deviation in the velocity, Cxx is the scaling co-efficient determined from the sensor geometry (2) and T is the inter-frame time (0.001s @ 1000fps).

Noise was added from a normal distribution with a standard deviation of 0.44. [79]

6.5 Discussion of Results

Two wheel robot simulation produces results which are expected under diverse influences or circumstances.

The simulation criterion was to control the mobile platform so that the robot can maintain it's forward direction at a configuration which depends on different parameters of the rover. These parameters were wheel efficiences, soil dynamics, sensors, and encoders for measurement. The MATLAB algorithm for the two wheel is designed to give an understanding of the steering of the rover. For the initial traverse the rover appears to travel in a forward direction as long its in hard ground, but soon as the wheels are under soft sand the rover steers.

Noise certainly has an influence in the simulation and avoiding this, can be achieved through the understanding of the noise model, this is further implemented in the KF. An improvement of the trajectory is achieved through the successful implementation of the KF and the associated parameters of the filter.

Using LSV sensors, we have derived a method that can accurately provide intermediate position estimates suggesting that if more sensors are used particularly with the KF, a better position estimate is achieved.

The results of the simulation did not take into account the placement of sensors in the robot. Calculations for placing or the location of the LSV or encoders suggest that the trajectory would be affected positively or negatively. So introducing or implementing diverse sensors can be more beneficial than simply implementing a simple kinematic drive for the rover.For avoiding minor or major errors a good estimate for orientation and positioning for the rover can always be achieved through the implementation of sensor fusion.

Chapter 7

Data Management

7.1 Introduction

Data management is a subject relevant to this research since there is valid data of interest coming out of sensors and the rover itself including computing system responsible for these. The research concerns the fusion of diverse sensors and data from these need to be validated to meet the objective of achieving data fusion. Therefore, data management can play a major role in this process. A need for robust and efficient sensor fusion can be attributed through Data Management for Sensor Fusion.Proposed summary of data of interest:

- LSV data
- Locomotion data
- Rover Clock measurement
- Communication data
- Visual Camera Data
- Computer information

Since there is an interest to proactively manage the operation of the Rover from ground segment, a unique product from Computer Associates (CA) has the ability to add value to the space operation activities. CA Unicenter NSM is an Enterprise Management product from CA; a company based Islandia, New York. This author has extensive experience working with this product, gained over 6 years while working for several IT services companies in the UK, USA and South Africa. The following observations regarding CA Unicenter NSM are derived directly from this authors experiences. CA claims that its IT Management e-business suite is robust and can offer e-business management regardless of IT platform. [60] Through the implementation of agent software and management, IT infrastructure can be comprehensively managed regardless of platform and hardware. It is possible to extend the functionality to space exploration activities ranging from ground segment infrastructure and space segment management. Some platforms of interest that can be managed:

- Windows platform (All versions)
- UNIX (Most versions)
- Linux (All versions)
- AS/400
- Netware
- Networks(TCP/IP,SNA,IPX/SPX, DECnet)
- Mainframes

The above platforms are not all necessarily part of the space operation infrastructure but a summary or idea what can be managed utilizing Unicenter.

7.1.1 Unicenter Architecture

The architecture for the product is diverse in structure consisting of major components that enable the monitoring and management of IT infrastructure. The architecture consists of simple to advanced components that enable visualisation of the infrastructure.[60] These tools usually are catered for business executives and technical people, (system administrators, technicians and engineers). The table below outlines some of the components found in the Unicenter architecture:

Table 7.1: Unicenter Architecture

Component	Description
WorldView	Visualisation of your IT using GUIs
Common Object Repository	Main database which holds Unicenter software objects.
Manager	Responsible for the reporting of agents.
Agents	Software residing at the monitored resource.

7.2 Agent Technology Background

Agent Technology is an interesting software management platform, because it is in real-time the software reports back to the manager should anything change from the managed resource. The power in this software is the ability to extensively monitor networks and systems coupled with visualization tools to configure activities in the infrastructure.(ca, 2013) Agent Technology should work well in a demanding spacecraft operation environment where data received from the spacecraft is constantly changing or when telecommanding to a spacecraft. Based on the scalability of Unicenter NSM and its diverse component structure, it is the authors assumption that agent technology can be deployed to manage planetary Rover telemetric data and have the capability to automate anomalies and procedures which are in place.

So a question could be asked? What are the specific resources are we interested in monitoring or managing for space exploration? Due to the scalability of the Unicenter Agents, they can be tasked to monitor close to all software resources associated with spacecrafts / robots, hardware and also ground segment related activities. These agents will reside on the monitored resource and real-time information will be viewed using 2D/3D visualisation tools e.g. node view or agent view. Since part of our interest is planetary Rovers and data fusion, we can implement the agents to manage the data fusion process if we have developed robust software that handles the LSV system and other sensors. Figure 7.1 is a flow diagram depicting the possible inter-relationship setup between sensors, fusion software and unicenter agents.

It is possible for the LSV system to be monitored by the agents based on past work experience for a client who needed the CCTV camera monitored for performance, outages and other metrics of interests. Since the LSV has a camera, this should be trivial implementation on board the Rover system. An example of a 2D visualisation of the monitored resources is shown in Figure 7.2:

7.3 Unicenter Managers

The Managers manage the agents residing on a monitored resource. The managers are usually installed on servers separate from the agents. Though in some instances it is possible to install agent and manager on the same machine provided we want to monitor that same server. These collect the information from the agents and send that information to the common object repository (COR), a central database. Real-time events from the entire infrastructure can be viewed from the event management console (a component of the manager) and specific messages and processes can be acted upon or automated. This feature may be of benefit mainly to the GS infrastructure but probably not for the space segment since there is a time delay to consider.

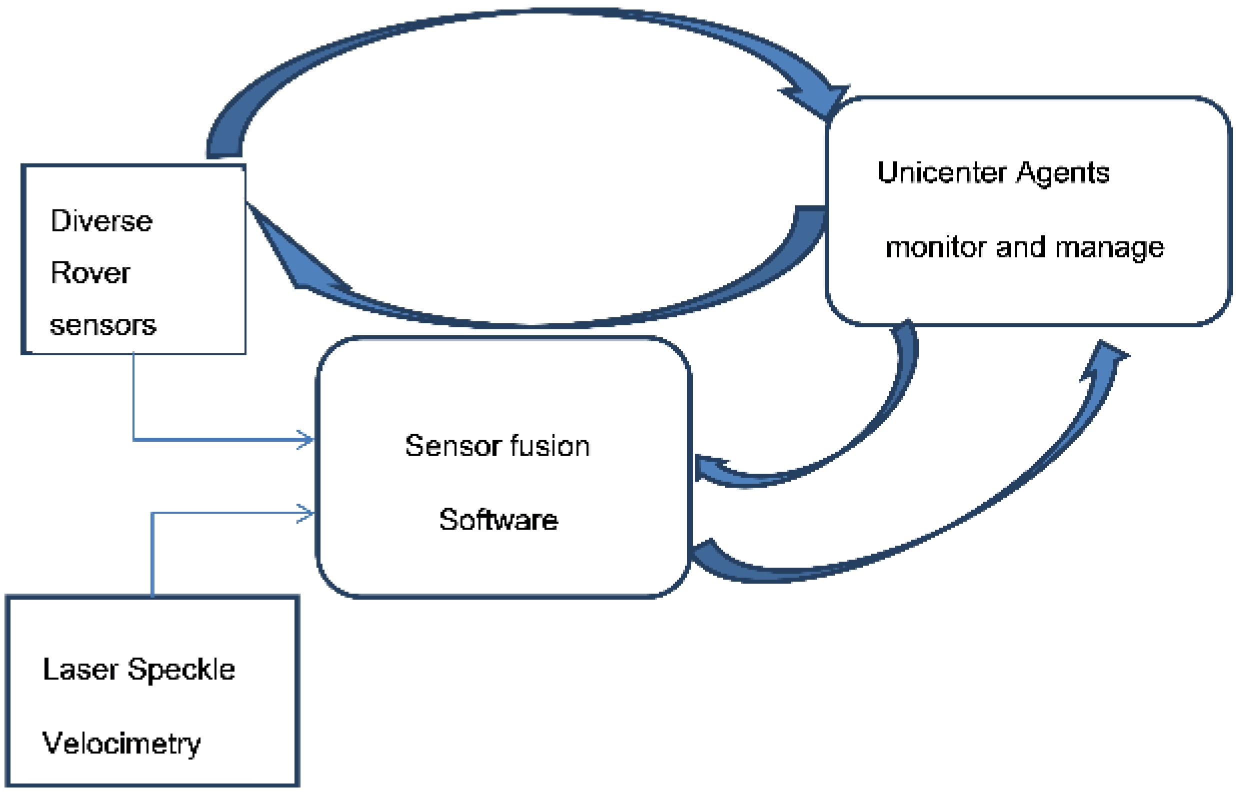


Figure 7.1: Unicenter and sensors diagram,

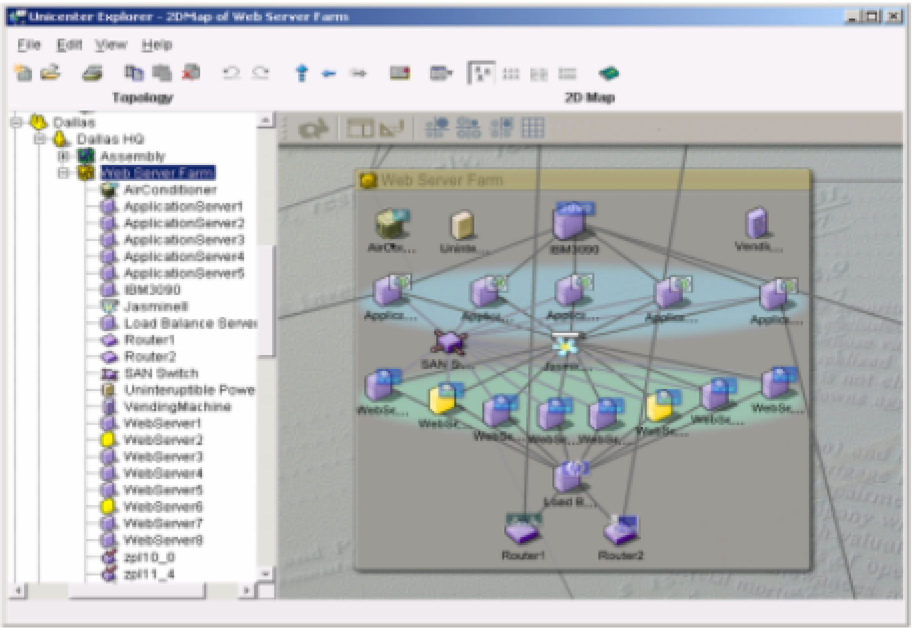


Figure 7.2: 2D Map Visualisation Example

7.3.1 Space Exploration Infrastructure

Unicenter and Space Exploration (Ground Segment / IT Infrastructure) Unicenter can bring significant benefits to existing space infrastructure. Since the product works well in IT business world for diverse industry, it should also work well for the space industry. There are advantages and disadvantages associated with the use of ca product.

Advantages

- Simple Implementation
- Lots of diverse software solution/options
- Good for producing reports
- Real time monitoring for Ground Segment

Disadvantages

- Complex training
- Demanding software package
- Expensive initial Investment
- Strong knowledge of diverse hardware and software (Networking, Databases,OS)

7.3.2 Unicenter Ground Segment Infrastructure

Unicenter can be integrated with the ground segment infrastructure to enable effective monitoring of diverse applications and network operations. The value Unicenter brings to the Ground Segment is:

- Scalability
- Automated tasks
- Performance management
- Visualization of Ground Segment activities

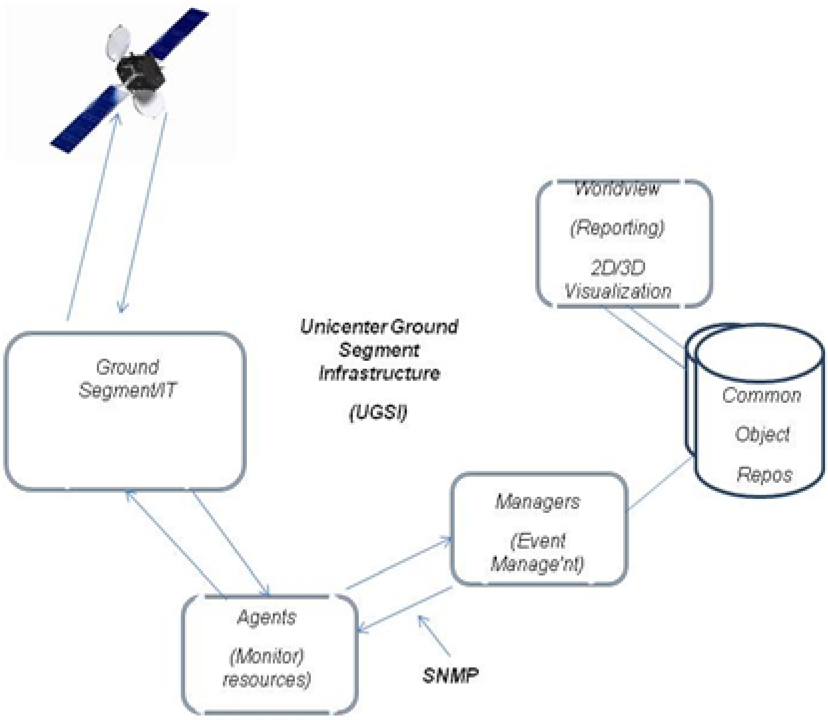


Figure 7.3: Unicenter Ground Segment

Monitoring and management of the IT infrastructure and ground segment is handled by a combination of scripting and software in place. This is in response to an engineers account from a small micro satellite manufacture company located in Surrey, UK. A lot of the basic monitoring, such as general network connectivity and nominal ground segment hardware operation is achieved through a series of scripts written in a scripting language that was developed in-house specifically for spacecraft operations. For the management of the network and application in the space operations environment the space company is relying on OpManager software. This tool as described from the companies website, manageEngine.com.Zoho [80] claims to offer comprehensive end to end monitoring for a host of diverse network devices and applications. This is in response to the growth and complexity of the ground segment and the need to effectively manage the entire infrastructure. The OpManager software seems to be similar to Unicenter NSM and other CA technologies software.

7.3.3 Unicenter Extended Features

In addition, CA claims that the product has the ability to integrate with a family of other optional software modules, including third party software, to further enhance the management / reliability of the IT infrastructure. The products that support or extend this operation or claim are the following:

- Unicenter Autosys job scheduler and workload management
- Unicenter Bright Store storage and backup for the entire infrastructure
- Unicenter Service Desk helpdesk tickets, opening and closing calls
- Unicenter Access Control - security control in the infrastructure

The above products have previously been implemented by the author for diverse clients and have proved to add value in driving costs down, efficiently managing the IT infrastructure, and help generating critical reporting information about the activities that affect the IT infrastructure. Though the optional modules are many, only a few can be relevant to the space exploration endeavour. For the management of our ground segment activities, as an example, we might have use for implementing Unicenter Brightstore for effectively managing telemetric data that is kept on a spacecraft database, and other space related data information found in diverse database systems in the GS infrastructure. Having a GS infrastructure protected by Unicenter access control software could add value through the implementation of policies that enable prevention of attacks to the infrastructure from the outside world to within the infrastructure.

There seem to be numerous advantages for using Unicenter as a tool for space segment and ground related activities, especially automated processes but the issue with the product is licensing. This may further need extensive investigation for the GS. A proof of concept is probably needed to see whether such a product can effectively drive down costs and works well in mission control activities. Currently in the industry most space mission control organisations / companies use open source software, some of the reasons being to enable development of software, to cut down costs and the ease of updating software. The extended features of Unicenter may be costly for the initial implementation but the value on investment is effective from the deployed product outcome.

7.3.4 Unicenter Sensor Monitoring

On board the Rover the vehicle is equipped with the following sensors:

- Wheel encoders
- Laser Speckle Velocimetry
- Accelerometers
- Visual Camera

The above sensors as well as other sensors onboard would need to be monitored and managed robustly for achieving the goal of traversing along the Martian surface. There are different opportunities for dealing with on-board sensors and the data they collect. The ground segment has a good infrastructure for sending instructions via telecommand as well as receiving via telemetry. Spacecraft Operation team has systems in place for monitoring data collected from the spacecraft or Rover vehicle. Unicenter is an optional product which can be used to effectively monitor, collect data and manage the on-board systems on a planetary Rover. One idea which can be implemented on-board a Rover is the installation of Unicenter agent software. CA gives a good account of the versatility of their agent software monitoring and managing diverse platforms and software applications. Since Curiosity recently had upgrade sensor software installed for the purpose of taking pictures while it traverses on the Martian terrain NASA (2011), installing Unicenter software for the planetary Rover should also work seamlessly provided the right requirements are fulfilled. The software will reside in the Rover since agent software needs to reside on the monitored resource. The only concern shall come from the downlink and uplink data acquisition.

As long the sensors are associated with on-board software and provide measurement data, the agent software has the ability to perform comprehensive data management and monitoring. Unicenter uses SNMP trapping for polling the agents to report to the managers should any activity change from the monitored resource. For a distant Mars Rover equipped with agent technology, the uplink and download would be analogous to SNMP trapping, and the ground segment infrastructure would have Unicenter Manager implemented for the reporting of agents.

Sensors on a planetary Rover can fail for a variety of reasons, for example poor manufacture, and errors in the design, random errors probably influenced by the motion of the rover (in an adverse terrain) and poor testing can all contribute to the lifespan of sensors. So, management and efficient monitoring is important for the functionality of the sensors.Also prevention of sensor failure is key to the locomotion of the rover.

7.4 Summary for technologies for Space Exploration

The ground segments vary with mission requirements and technologies are currently available to monitor and manage such environments. Some technologies which might be considered for the ground segments and operate similar are:

- CA Unicenter Computer Associates
- Microsoft SMS - Microsoft
- HP Openview Hewlett Packard
- OpManager Manageengine

The above toolsets work well in demanding IT infrastructures. The author has only worked with CA Unicenter from Computer Associates, and from his previous architecture and implementation experience, the product adds value, and drive down costs in relation to the IT environment being managed.

The space / satellite industry is a niche market. There are several satellite companies in UK and Europe who provide services for the end user/scientific user. Some of these are :

- Inmarsat
- Telespazio Vega
- SES
- SSTL
- Astrium Services
- Comsat
- Intelsat
- ESA

The above clients or customers vary in service offerings to the end user and that also determines the size of the IT infrastructure or ground segment involved. If the infrastructure is large, then we are assume the diverse software and hardware are of complex nature but if the IT infrastructure is small then the environment consist of few hardware/device items and software. SSTL, as an example looks after few satellites compared to Inmarsat who look after many satellites.

The enterprise management software toolsets from different vendors mentioned earlier in this section usually are for large IT infrastructure environment since heterogeneous infrastructure would have become complex and enterprise solution software is needed for the management of systems and devices. So clients like ESA, Intelsat, SES and others with large infrastructure in place might have a good use for enterprise management software (CA Unicenter, HP Openview, and SMS) but SSTL and other smaller ones might benefit a little. The question which toolset will do a good job for space exploration will depend on the features the clients would be interested (i.e technologies he/she would like to manage) in and as well as functionality. On face value the products from different vendors appear similar in functionality though there should be a difference in the feature and some functionality offerings. A proof of concept may be needed to test the enterprise tools onsite to validate their offerings and deduce the impact to ground segments/IT infrastructure.

7.5 Data Management Summary

Data management involves a lot of topics particularly relevant to Planetary Rover. The architecture of the infrastructure of sensor fusion, algorithms involved in the computation of the moving robot and information monitoring are vital topics that can enable an efficient and secure operation of a traversing Planetary Rover. These can be achieved through some Data Management software as mentioned in this chapter.

Chapter 8

Conclusion

The combination subject of Kalman Filters and LSV is a novel one which has been presented in chapter 3 and the simulation results on chapter 4. The advantages of using sensor fusion have been noted particularly improving the position estimation process. The disadvantage lies in the sensitivity of the Laser Speckle system, especially when the Rover is on idle mode, it still makes measurements and this becomes a major task to model errors associated with this sensitivity.

This preliminary kinematic model simulation in MATLAB enabled more developed approach to work with other sensors, e.g. Chapter 3 has detailed the implementation of the LSV and odometry which has produced satisfactory results in relation to simulations. The LSV sensor does a good job of following the true trajectory while the encoder does not follow the truth. The written MATLAB algorithm had the capability to facilitate noise distribution on the trajectory of the Rover. So noise in the dynamics of the vehicle showed that the position of the robot eventually turning depends on the noise levels.

The KF Model was introduced in the simulation to observe the effect of the algorithm in the localization process. For initial purposes the noise levels were simply assumed for the simulation sake. We fused the LSV sensor and wheel encoder sensor to make a comparison and have the KF produce the estimated trajectory. The results were what we expected; KF trajectory followed the LSV trajectory close as seen on simulation graphics on chapter 4. Fusion with a good initial pose estimate, the position estimation algorithm produces enhanced measurements and results, even on sandy environment. This proves the robustness of the sensor fusion technique.

8.1 Position of the Rover

The position of the Rover in time has been discussed and simulated. We considered the Rover position under noisy conditions and in a perfect state. Lessons from the kinematic model gave us the opportunity to understand the different parameters that influence the motion of the two wheel Rover. If we vary the speed of the wheels, the Rover turns because of less speed on the affected wheel. Placing LSV sensors on the Rover enabled the measure of vx and vy and enabled the plot of these velocities in time. The position estimation process using the LSV sensor enabled us to understand the benefits of measuring position in time. Some discussions about the benefits of using LSV and encoders for the positioning estimation problem were outlined from early chapters. Furthermore, the influence of noise on the sensors to the positioning problem was simulated to better understand the sensor modelling for the Rover. One area which was not exploited because of time constraints was the modelling and influence of using the exteroceptive sensors for the localisation problem and sensor fusion process. Many literatures suggest good position estimation outcomes for the sensor fusion outcome.

8.2 Summary of Kalman Filter

The study and implementation of the Kalman Filters has proved to be invaluable for this research. The algorithm was central to the sensor fusion process as well as preliminary tests for simulating simple models through the aid of data sets. Some of these data sets which were part of the sensor fusion implementation were, LSV, encoders and accelerometers. It was also shown how the noise amplitude found in the measurements impacts the estimated process.

The algorithm definition was outlined in chapters 2 and 3, with its various elements. Assumptions we used for uncertainty and noise correlation during the mathematical operation of the algorithm. For the computation process of the KF, we added white, zero-mean Gaussian noise to see the effects the KF has on the measurements. The equations which make up the algorithm were clearly defined for the purpose of understanding the simulations which were performed with MATLAB.

So for the two wheel Rover simulation or localization we have seen the performance of the algorithm and the usefulness when using LSV information as data input. While varying the noise parameters, from the measurement and system model we could see the effect of uncertainty in the outcome of the algorithm computation process. The computation results enable us to learn more about various parameters involved during the calculation process. Uncertainty is a subject which still needs to be further explored based on the models developed. Accuracy of the developed models was based on the assumptions made for noise modelling. It was a good concept to test noise as Gaussian distribution for fulfilling the system modelling and the measurement modelling, however though the noise distribution on the LSV measurement was uniform, there were other noise sources which were not zero mean and which needed to be modelled respectively. This was along the X and Y direction of the vehicle as well as the surface height of the LSV and Rover.

8.3 Possible Future Work

Investigating the rotation of the vehicle, varying the surface height as the vehicle is moving and utilizing the Kalman Filter gives us the opportunity to further our research and capitalize on the measurement made thereof.

More work is needed to see the effect of different type of soils e.g. Martian soil, pebbles or rocks. When all these are extensively modelled, the results in conjunction with LSV measurements may improve accuracy for the odometer application and this would confirm a robust LSV system in place. An opportunity also exists for dealing with obstacles as the Rover is traversing using the LSV.

The noise issue for Rovers may have been modelled in a simple manner by assuming Gaussian for our process noise measurement and measurement noise modelling but in the real world the noise levels are complex because it is uncertain in most cases whether in some systems errors are independent or non independent. So the issue of studying noise modelling in detail is an option to better our understanding of the modelling technique associated with system modelling of the Rover. For the Rover and its measurement system (LSV) we might look at the functionality of the LSV optics and detector equipments in detail to further our understanding about the errors associated thereof. What are the effects of sensors in time variation and noise?

Summary of future work and opportunity includes the following:

- More analysis of diverse sensors, particularly concerning defects and noise, and model this for the rover application. The encoders need to be studied further to correct their outcome as an odometry reader.In some instances the odometry would produce high errors and simply relying on the kalman filter may not always be the best option and optimizing every sensor is an option.
- Assess the LSV and rover robustness for different environments. The Antarctica rover is an example of this. Also researching the performance of sensors in different weather patterns.
- Learning and contrasting LSV technology and the fusion of sensors trends with current robotic applications in terms of performance, reliability and robustness
- Using the Kalman Filter for data fusion has been the basis for this thesis, what happens when more sensors are integrated to the data fusion process?On theory this should give us an optimal estimate and there is an opportunity to learn the behaviour of diverse sensors for the KF implementation. An opportunity also exists for the use of alternative filters and gathering data to see the benefits of such implementation. Simply using the KF might not be the best option.

Last but not least, the Kalman Filter should aid in the fusion process. Other types of sensors may be integrated to see the effect and their limit for estimating processes.

Is the implementation of sensor fusion (including LSV) on a planetary Rover power efficient? Or computationally expensive?meaning that complicated algorithms will need to be developed and therefore using more computing resources:

- CPU, a faster processor is needed for faster response time or data fetching mechanism.
- Memory, The random access memory (RAM) would need to be large for processing images produced by the algorithms.
- Storage, a critical part in computing for maintaining image archives. The harddrive would need to be large.
- Data Architecture, the type of data bus structure used,and interfaces.
- Redundancy Systems, failover system design and fault tolerance.

More models and simulations would answer these questions.

References

[1] Wikiversity. *Rover Mission and Analysis*, 2011 (accessed June 13, 2014). http://en.wikiversity.org/wiki/Rover_mission_analysis_and_design.

[2] NASA. *Five Things About NASA's Mars Curiosity Rover - 09.16.10*, 2014 (accessed November 24,2014). http://www.nasa.gov/images/content/482643main_msl20100916-full.jpg/.

[3] Leonard David. *How Wheel Damage Affects Mars Rover Curiosity's Mission*, 2014 (accessed July 9, 2014). http://www.space.com/26472-mars-rover-curiosity-wheel-damage.html.

[4] et al Chao Jiang ; Hongyan Zhang ; Jia Wang ; Yaru Wang ; Heng He. Dedicated hardware processor and corresponding system-on-chip design for real-time laser speckle imaging. *J. Biomed. Opt. 16(11), 116008 (October 31, 2011)*, 2011.

[5] Dan Smith. *Fixed Lag Smoothing*, 2014 (accessed June 20, 2014). http://www.globalspec.com/reference/19861/160210/chapter-9-3-fixed-lag-smoothing.

[6] R. Welch, D. Limonadi, and R. Manning. Systems engineering the curiosity rover: A retrospective. In *System of Systems Engineering (SoSE), 2013 8th International Conference*, 2013.

[7] J. Maki, D. Thiessen, A. Pourangi, P. Kobzeff, T. Litwin, L. Scherr, S. Elliott, A. Dingizian, and M. Maimone. The mars science laboratory engineering cameras. *Space Science Reviews*, 2012.

[8] M. Bajracharya, M. W. Maimone, and D. Helmick. Autonomy for mars rovers: Past, present, and future. 2008.

[9] A. A. Aliverdiev, M. A. Caponero, C. Moriconi, P. A. Fichera, and G. Sagratella. Laser speckle velocimeter for a robotized vehicle. In *2nd International Conference on Advanced Optoelectronics and Lasers, CAOL 2005*, 2008.

[10] S. Kilston and E. Friedman. Space-how far we have come, how far there is to go. *Proceedings of the IEEE*, 2000.

[11] W. L. Freedman. How old is the universe? measuring the expansion rate with the hubble space telescope. In *Aerospace Applications Conference, 1996.*

[12] B. Hamza, A. Nebylov, and G. Yatsevitch. Original solutions for localization and navigation on the surface of mars planet. In *Aerospace Conference, 2010 IEEE.*

[13] A. Coustenis, M. Hirtzig, G. Bampasidis, A. Solomonidou, E. Bratsolis, K. Kyriakopoulos, X. Moussas, and P. Preka-Papadema. Exploring the satellites of the outer planets with in situ elements. In *2nd International Conference on Space Technology, ICST 2011.*

[14] A. Helfrick. Undergraduate spacecraft design projects. *Aerospace and Electronic Systems Magazine, IEEE*, 2006.

[15] E. R. Taylor. Evaluation of multidisciplinary design optimization techniques as applied to spacecraft design. In *Aerospace Conference Proceedings, 2000 IEEE.*

[16] G. Paar, L. Waugh, D. P. Barnes, T. Pajdla, M. Woods, H. R. Graff, Y. Gaog, K. Willnerh, J. P. Mulleri, and R. Li. Integrated field testing of planetary robotics vision processing - the provisg campaign in tenerife 2011. In *Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*, 2012.

[17] A. Lambert, P. Furgale, T. D. Barfoot, and J. Enright. Field testing of visual odometry aided by a sun sensor and inclinometer. *Journal of Field Robotics*, 2012.

[18] L. Matthies, M. Maimone, A. Johnson, Y. Cheng, R. Willson, C. Villalpando, S. Goldberg, A. Huertas, A. Stein, and A. Angelova. Computer vision on mars. *International Journal of Computer Vision*, 2007.

[19] C. F. Olson, L. H. Matthies, J. R. Wright, R. Li, and K. Di. Visual terrain mapping for mars exploration. *Computer Vision and Image Understanding*, 2007.

[20] J. Hidalgo, P. Poulakis, J. Khler, J. Del-Cerro, and A. Barrientos. Improving planetary rover attitude estimation via mems sensor characterization. 2012.

[21] C. H. Tong, T. D. Barfoot, and E. Dupuis. Three-dimensional slam for mapping planetary work site environments. *Journal of Field Robotics*, 2012.

[22] N. Azer, A. T. Erdem, A. . Ercan, and . E. Erdem. Inertial sensor fusion for 3d camera tracking. In *2012 20th Signal Processing and Communications Applications Conference, SIU 2012.*

[23] X. Ruan, Y. Gao, H. Song, J. Yu, and D. Gong. Robot attitude estimation based on kalman filter and application to balance control. *Advanced Science Letters*, 2012.

[24] Xiaokang Song, Yuechao Wang, Zhenwei Wu, Chunguang Bu, and Yong Chang. Kinematics-based velocity estimation of lunar rovers. In *Robotics and Biomimetics, 2007. ROBIO 2007.*

[25] D. Nistr, O. Naroditsky, and J. Bergen. Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 2006.

[26] Changbae Jung and Woojin Chung. Accurate calibration of two wheel differential mobile robots by using experimental heading errors. In *Robotics and Automation (ICRA), 2012 IEEE.*

[27] M. Kam, X. Zhu, and P. Kalata. Sensor fusion for mobile robot navigation. *Proceedings of the IEEE*, 1997.

[28] A. Wickramasooriya, G. Hamilan, L. S. I. Jayawardena, W. M. D. Wijemanne, and S. R. Munasinghe. Characteristics of sonar range sensor srf05. In *Information and Automation for Sustainability, 2008. ICIAFS 2008.*

[29] M. F. Misnan, N. H. M. Arshad, R. L. A. Shauri, N. A. Razak, N. M. Thamrin, and S. F. Mahmud. Dynamic height measurement using sonar sensor based on single dimension kalman filter in embedded controller. In *Control System, Computing and Engineering (ICCSCE) 2012,*.

[30] Sungjin Jo, Hyukdoo Choi, and Euntai Kim. Ceiling vision based slam approach using sensor fusion of sonar sensor and monocular camera. In *Control, Automation and Systems (ICCAS), 2012.*

[31] S. Thompson and S. Kagami. Stereo vision and sonar sensor based view registration for 2.5 dimensional map generation. In *Intelligent Robots and Systems, 2004. (IROS 2004).*

[32] JPL/NASA. *Mars Science Laboratory/Curiosity*, 2014 (accessed June 13, 2014). `http://www.jpl.nasa.gov/news/fact_sheets/mars-science-laboratory.pdf`.

[33] R. A. Lindemann, D. B. Bickler, B. D. Harrington, G. M. Ortiz, and C. J. Voorhees. Mars exploration rover mobility development - mechanical mobility hardware design, development, and testing. *IEEE Robotics and Automation Magazine*, 2006.

[34] T. M. Howard, A. Morfopoulos, J. Morrison, Y. Kuwata, C. Villalpando, L. Matthies, and M. McHenry. Enabling continuous planetary rover navigation through fpga stereo and visual odometry. In *IEEE Aerospace Conference*, 2012.

[35] S. Farritor, H. Hacot, and S. Dubowsky. Physics-based planning for planetary exploration. In *IEEE International Conference on Robotics and Automation*, 1998.

[36] J. Hilland, A. Bhanji, and P. Estabrook. Mars exploration rover telecommunications subsystem. In Pistora J., Postava K., Hrabovsky M., and Rawat B.S., editors, *Microwave and Optical Technology*, 2004.

[37] Oliver Woodman. An introduction to inertial navigation, 2007.

[38] A. Adam, E. Rivlin, and H. Rotstein. Fusion of fixation and odometry for vehicle navigation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 1999.

[39] F. Chenavier and J. L. Crowley. Position estimation for a mobile robot using vision and odometry. In *IEEE International Conference on Robotics and Automation*, 1992.

[40] A. Martinelli. Improving the precision on multi robot localization by using a series of filters hierarchically distributed. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007.*

[41] D. Francis, T. O. H. Charrett, L. Waugh, and R. P. Tatam. Objective speckle velocimetry for autonomous vehicle odometry. *Applied Optics*, 2012.

[42] V. J. Delgado, P. R. Kurka, and E. Cardozo. Visual odometry in mobile robots. In *2011 IEEE 9th Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control, LARC 2011.*

[43] S. B. Goldberg, M. W. Maimone, and L. Matthies. Stereo vision and rover navigation software for planetary exploration. In *Proceedings of the IEEE Aerospace Conference*, 2002.

[44] S. Se, H. K Ng, P. Jasiobedzki, and T. J Moyung. Vision based modeling and localization for planetary exploration rovers. In *55th International Astronautical Congress*, 2004.

[45] N. Patel, R. Slade, and J. Clemmet. The exomars rover locomotion subsystem. *Journal of Terramechanics*, 2010.

[46] S. Zhang. Recent progresses on real-time 3d shape measurement using digital fringe projection techniques. *Optics and Lasers in Engineering*, 2010.

[47] S. Chen, Y. Li, and N. M. Kwok. Active vision in robotic systems: A survey of recent developments. *International Journal of Robotics Research*, 2011.

[48] Y. Yachide, M. Ikeda, and K. Asada. High-speed 3-d measurement system using smart image sensor and fpga based 3-d engine. In *Electronics, Circuits and Systems, 2006. ICECS '06.*

[49] Toshimitsu Asakura Yoshihisa Aizu. *Spatial Filtering Velocimetry: Fundamentals and Applications.* 2006.

[50] Z. Jan, A. Olwal, and R. Raskar. Specklesense: Fast, precise, low-cost and compact motion sensing using laser speckle. 2011.

[51] J. Brillaud and F. Lagattu. Limits and possibilities of laser speckle and white-light image-correlation methods: Theory and experiments. *Applied Optics*, 2002.

[52] Li-Wei Fong. Multisensor data fusion with estimated weights. In *Industrial Electronics, 2006 IEEE International Symposium*, 2006.

[53] H. J. Kushner and W. J. Runggaldier. Filtering and control for wide bandwidth noise driven systems. 1987.

[54] Y. Ersoy and M. Efe. Fused and adaptive modified wave estimators for navigation systems. 2006.

[55] Jihua Zhu, Nanning Zheng, Zejian Yuan, Qiang Zhang, Xuetao Zhang, and Yongjian He. A slam algorithm based on the central difference kalman filter. 2009.

[56] Joseph J. LaViola Jr. Double exponential smoothing: An alternative to kalman filter-based predictive tracking. 2003.

[57] M. S. Grewal and A. P. Andrews. Applications of kalman filtering in aerospace 1960 to the present. *IEEE Control Systems Magazine*, 2010.

[58] Bilgin Esme. Kalman filters for dummies, 2009.

[59] Ruslan Masinjila. *Extended Kalman Filter for two wheel*, 2014 (accessed November 24, 2014). `https://github.com/ruslanmasinjila/ekf_simulator`.

[60] Computer Associates Technologies. *Infrastructure Management*, 2014 (accessed June 13, 2014). `http://www.ca.com/us/products/infrastructure-management.aspx`.

[61] T. O. H. Charrett, L. Waugh, and R. P. Tatam. Speckle velocimetry for high accuracy odometry for a mars exploration rover. *Measurement Science and Technology*, 2010.

[62] S. Lee and J. B Song. Robust mobile robot localization using optical flow sensors and encoders. In *IEEE International Conference on Robotics and Automation*, 2004.

[63] Avago Technologies. *Two Channel Optical Incremental Encoder Modules*, 2014 (accessed July 21, 2014). `http://www.avagotech.com/docs/AV02-1867EN`.

[64] S. Y. Chen. Kalman filter for robot vision: A survey. *IEEE Transactions on Industrial Electronics*, 2012.

[65] S. K. Pillai. Techniques in model error estimation by means of linear kalman filtering. 1996.

[66] B. Ekstrand. Some aspects on filter design for target tracking. *Journal of Control Science and Engineering*, 2012.

[67] James P. Underwood, Andrew Hill, Thierry Peynot, and Steven J. Scheding. Error modeling and calibration of exteroceptive sensors for accurate mapping applications. *Journal of Field Robotics*, 2010.

[68] D. C. Herath, K. R. S. Kodagoda, and G. Dissanayake. Modeling errors in small baseline stereo for slam. In *9th International Conference on Control, Automation, Robotics and Vision*, 2006.

[69] K. S. Chong and L. Kleeman. Accurate odometry and error modelling for a mobile robot. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation.*

[70] M. M. Trivedi, M. A. Abidi, R. O. Eason, and R. C. Gonzalez. Developing robotic systems with multiple sensors. *IEEE Transactions on Systems, Man and Cybernetics*, 1990.

[71] H. H. S. Liu and G. K. H Pang. Accelerometer for mobile robot positioning. *IEEE Transactions on Industry Applications*, 2001.

[72] T. T. Q. Bui, A. Widyotriatmo, Nguyen Le Hoa, and Hong Keum-Shik. Sonar-based collision avoidance and probabilistic motion model for mobile robot navigation. 2009.

[73] David Bradley. Odometry: Calibration and error modeling, 2006.

[74] A. Kelly. Fast and easy systematic and stochastic odometry calibration. In *Intelligent Robots and Systems, 2004. (IROS 2004).*

[75] Wang Junli, Yin Fuchang, and Song Zhengxun. Laser speckle images research based on wavelet-domain hidden markov models. 2011.

[76] C.C. Castello, J.R. New, and M.K. Smith. Autonomous correction of sensor data applied to building technologies using filtering methods. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, 2013.

[77] Manolis Anadranistakis, Kostas Lagouvardos, Vassiliki Kotroni, and Helias Elefteriadis. Correcting temperature and humidity forecasts using kalman filtering: potential for agricultural protection in northern greece. *Atmospheric Research*, 2004.

[78] H. Zhao and Z. Wang. Motion measurement using inertial sensors, ultrasonic sensors, and magnetometers with extended kalman filter for data fusion. *IEEE Sensors Journal*, 2012.

[79] 2014 Thomas Charrott. Personal communication, cranfield university, 2014.

[80] Zoho Corporation. Network monitoring software, 2013 (accessed July 14, 2014). `http://www.manageengine.com/network-monitoring/?gclid=CKCthb7yg7cCFUzHtAodawIAgA`.

Appendix A

List of Publications

Simulation of a two wheel robot using Laser Speckle Velocimetry and Odometry, Anthony Masuku and co-author Dr Stephen Hobbs accepted by Journal of Unmanned Systems Technology

Appendix B

Kalman Filter, LSV and Encoder trajectories

```
%The trajectories for Kalman Filter, encoder
%readiings and laser speckle readings

%Initialize unchanging variables

clear;
clc;
a = 0.1;  %Wheel radius
b = 0.9;  %Wheel axle length

dt = 0.8; %Time step
numsteps =300;

Ymax = 3;
Ymin = 2;

load LSVdata.txt % LSV Data for single sensor(Vx and Vy) collected from Optics grou


% Sort out x and y coordinates for LSVdata collected
lsv_dx = LSVdata(:,2);
lsv_dy = LSVdata(:,1);

lsv_vx = (LSVdata(:,2)* (14 * 10^-6))/(1.985 * 0.0017);
lsv_vy = (LSVdata(:,1)* (14 * 10^-6))/(2.000 * 0.0017);

lsvxpos =(LSVdata(:,2)* (14 * 10^-6)/ 1.985);
lsvypos =(LSVdata(:,1)* (14 * 10^-6)/2.000);
```

```
position = LSVdata';

posxLSV = cumsum(lsvxpos,1);
posyLSV = cumsum(lsvypos,1);

posicion = [posxLSV,posyLSV]';


% Plotting LSV data raw

plot(LSVdata(:,2));
plot(LSVdata(:,1));

% Errors from the LSV data collected from Optics Group
sgx = 0.08; % in m s-1
sgy = 0.07; % in m s-1


%Initialize changing variables

speedwheels =  repmat([0.55,0.55],numsteps,1);

pose = zeros(length(speedwheels),4);

eps1=zeros(numsteps,1);
eps2=zeros(numsteps,1);

degPerRad = 180/pi;   %Conversion degrees to radian
RadPerDeg = 1/degPerRad;

start = [0,0,0,40*RadPerDeg]; % Start position and initial pose of the rover, degre

pose(1,:)= start;



% The beginning of the for loop to
% Nval

for i = 2:1:(numsteps)

t =  pose(i-1,1);
x = (pose(i-1,2));
y = (pose(i-1,3));
theta = pose(i-1,4);
```

```
  omega1True = speedwheels(i-1,1); % Speed of wheel1
  omega2True = speedwheels(i-1,2); % Speed of wheel2


% Position coordinates for wheel1 and wheel2
x1 = x - (b/2) * sin(theta);
y1 = y + (b/2) * cos(theta);
x2 = x + (b/2) * sin(theta);
y2 = y - (b/2) * cos(theta);

% Testing the sandy region by setting upper and lower limits
if (Ymin < y1)&& (y1 < Ymax)

     eps1(i-1)= 0.5; % Set 0.5 as a constant for sandy soil
 else
     eps1(i-1) = 1;  % Set 1 as a constant for hard ground

end

if (Ymin < y2)&&(y2 < Ymax)

     eps2(i-1)= 0.5;

 else
     eps2(i-1) = 1;
 end

epsvalues(i,:) = [eps1(i-1),eps2(i-1)]; %Store epsilon values in a vector



velTru = ( omega1True*eps1(i-1) + omega2True*eps2(i-1))*0.5*a; %Calculate forward v
thetTru = -( omega1True*eps1(i-1) - omega2True*eps2(i-1))*(a/b);%Calculate orientat


pose(i,1) = pose(i-1,1) +  dt; %increment in time
pose(i,2) = pose(i-1,2) + (dt * velTru) * cos(pose(i-1,4)); % Calculate in timestep
pose(i,3) = pose(i-1,3) + (dt * velTru) * sin(pose(i-1,4)); % Calculate in timestep
pose(i,4) = pose(i-1,4) + (dt * thetTru);% Calculate in timesteps,orientation

wheel1(i,:)= [x1,y1]; % Save wheel1 coordinates values
wheel2(i,:)= [x2,y2]; % Save wheel2 coordinates values

M2{1} = wheel1(:,1);  % Save wheel1 x data to cell array
M2{2} = wheel1(:,2);  % Save wheel1 y data to cell array
```

```
M3{1} = wheel2(:,1);  % Save wheel2 x data to cell array
M3{2} = wheel2(:,2);  % Save wheel2 y data to cell array

posXtru = [pose(:,2)];
posYtru= [pose(:,3)];

end




noiseAmp = 0.517; %rad/s, Taken from National Instruments in relation to encoder ro



%Laser Speckle sensor readings measured from vTru and wTru reading


%Position LSV sensor from center of axle

senLsv_x1 = 0.3 * b;
senLsv_y1 = 0.2;
senLsv_x2 = 0.5 * b;
senLsv_y2 = 0.3;
senLoc1 = [senLsv_x1,senLsv_y1];
senLoc2 = [senLsv_x2,senLsv_y2];


% Calculate v and w using vx and vy measurements

randn(numsteps,1);

randomNval = [randn(numsteps,1)];



for j =2:1:(numsteps)

t =    pose(j-1,1);
xSen = pose(j-1,2);
ySen = pose(j-1,3);
theta = pose(j-1,4);
```

```
%Position coordinates for wheel1 and wheel2
x1Pos = xSen - (b/2) * sin(theta);
y1Pos = ySen + (b/2) * cos(theta);
x2Pos = xSen + (b/2) * sin(theta);
y2Pos = ySen - (b/2) * cos(theta);

% Testing the sandy region by setting upper and lower limits
if (Ymin < y1Pos)& (y1 < Ymax)

     epsi1(j-1)= 0.5; % Set 0.5 as a constant for sandy soil
 else
     epsi1(j-1) = 1;  % Set 1 as a constant for hard ground

end

if (Ymin < y2Pos)& (y2 < Ymax)

     epsi2(j-1)= 0.5;

 else
     epsi2(j-1) = 1;
 end



% Measuring Laser speckle,vx and vy with noise disturbances using
% true velocity

vx1Meas = velTru - thetTru * senLsv_x1 + noiseAmp * randomNval(j);
vy1Meas = senLsv_y1 * thetTru + noiseAmp * randomNval(j);
vx2Meas = velTru - thetTru * senLsv_x2 + noiseAmp * randomNval(j) ;
vy2Meas = senLsv_y2 * thetTru + noiseAmp * randomNval(j);

w = (vx1Meas - vx2Meas)/(senLsv_x2 - senLsv_x1) ; %Calculate orientation, theta

v = (vx1Meas + vx2Meas)/ 2 + (vx1Meas - vx2Meas)/(senLsv_x2 - senLsv_x1) * (senLsv_


pose(j,1) = pose(j-1,1)+ dt; %increment in time
poseSpeckle(j,2) = pose(j-1,2)+ (dt * v) * cos(pose(j-1,4)); % Calculate in timeste
poseSpeckle(j,3) = pose(j-1,3)+ (dt * v) * sin(pose(j-1,4)); % Calculate in timeste
poseSpeckle(j,4) = pose(j-1,4) +(dt * w); % Calculate in timesteps , orientation

wheels1(j,:)= [x1Pos,y1Pos]; % Save wheel1 coordinates values
wheels2(j,:)= [x2Pos,y2Pos]; % Save wheel2 coordinates values
```

```
M6{1} = wheels1(:,1);  % Save wheel1 x data to cell array
M6{2} = wheels1(:,2);  % Save wheel1 y data to cell array

M7{1} = wheels2(:,1);  % Save wheel2 x data to cell array
M7{2} = wheels2(:,2);  % Save wheel2 y data to cell array


t =  pose(j-1,1);
xSensor(j) = [poseSpeckle(j-1,2)];
ySensor(j) = [poseSpeckle(j-1,3)];
thetaSensor(j) = [poseSpeckle(j-1,4)];


lsvYpos = [ySensor];
lsvXpos =  [xSensor];

end



%Encoder readings measured from true readings (omega1True and omega2True)

%Initialize changing variables

w_meas1 = omega1True + noiseAmp * randomNval; %rad/s , Add noise and disturbances t

w_meas2 = omega2True + noiseAmp * randomNval;

wheelspeed = [w_meas1,w_meas2];





for ii = 2:1:(numsteps)


t =  pose(ii-1,1);
xEs = (pose(ii-1,2));
yEs = (pose(ii-1,3));
theta = pose(ii-1,4);

  omega1Est = wheelspeed(i,1); % Speed of wheel1
  omega2Est = wheelspeed(i,2); % Speed of wheel2
```

```
% Position coordinates for wheel1 and wheel2
x1E = xEs - (b/2) * sin(theta);
y1E = yEs + (b/2) * cos(theta);
x2E = xEs + (b/2) * sin(theta);
y2E = yEs - (b/2) * cos(theta);


vE(ii-1) = ( omega1Est + omega2Est)*0.5*a; %Calculate forward velocity
wE(ii-1)= -( omega1Est - omega2Est)*(a/b);%Calculate orientation, theta


pose(ii,1) = pose(ii-1,1)+ dt; %increment in time
pose(ii,4) = pose(ii-1,4) + (dt * wE(ii-1)); % Calculate in timesteps , x positon
pose(ii,2) = pose(ii-1,2)+ (dt * vE(ii-1)) * cos(pose(ii-1,4)); % Calculate in time
pose(ii,3) = pose(ii-1,3)+ (dt * vE(ii-1)) * sin(pose(ii-1,4)); % Calculate in time



wheel1Est(ii,:)= [x1E,y1E]; % Save wheel1 coordinates values
wheel2Est(ii,:)= [x2E,y2E]; % Save wheel2 coordinates values

M4{1} = wheel1Est(:,1);  % Save wheel1 x data to cell array
M4{2} = wheel1Est(:,2);  % Save wheel1 y data to cell array

M5{1} = wheel2Est(:,1);  % Save wheel2 x data to cell array
M5{2} = wheel2Est(:,2);  % Save wheel2 y data to cell array

posXest(ii) = [pose(ii,2)];
posYest(ii) = [pose(ii,3)];
end


%Plot the results

%  plot(posXtru,posYtru,'r--')
%  hold on
  plot(posXest,posYest,'g--')
 hold on
  plot(lsvXpos,lsvYpos,'b--')


 xlabel('X position','FontSize',16);
ylabel('Y position','FontSize',16);
title('Rover trajectories,diverse sensors','FontSize',16);
text(2*pi/4,sin(2.5*pi/4),...
            '   Red - Kalman Filter Trajectory',...
```

```
                'FontSize',12)
            text(2*pi/4,sin(3.2*pi/4),...
             '   Green - Encoder Trajectory',...
                'FontSize',12)
            text(2*pi/4,sin(3.8*pi/4),...
             '   Blue - Laser Speckle V Trajectory',...
                'FontSize',12)



% apply kalman filtering:


R=[sgx^2   0
   0       sgy^2]; % covariance matrix for noise in x and y

H=[1 0 0 0 0;
   0 0 1 0 0]; % observation matrix

M=[ 1  dt 0  v * sin(0)  cos(1)
    0  1  0  v * cos(1)  sin(0)
    0  0  1  dt 0
    0  0  0  1  dt
    0  0  0  0  1  ]; % state transition matrix

P=eye(5); % initial guess for P-matrix

I=eye(5); % unit matrix

Q = eye(5);
Q(1,1)= 1000;
Q(2,2)= 1000;
Q(3,3)= 1000;
Q(4,4)= 100;
Q(5,5)= 100;




x_est = [lsvXpos(3)
         posXest(3)
         lsvYpos(3)
         posYest(3)
         0];            % x_est=[lsvXpos,posXest,lsvYpos,posYest,ay]'

 p_est = zeros(5, 5);
```

```
x_hat = zeros(5,300);
p_hat = [];


r = [lsvXpos+sgx*randn; lsvYpos+sgy*randn];%Measurement with LSV data



for i=1:300
   x_prd = M * x_est;    % Predicted state and covariance
   p_prd = M * p_est * M' + Q;

   S = (H * p_prd') * H' + R;    % Estimation
   B = H * p_prd';
   klm_gain = (S \ B)';

   x_est = x_prd + klm_gain * (r(:,i) - H * x_prd); % Estimated state and covarianc
   p_est = p_prd - klm_gain * H * p_prd;

   x_hat(:,i) = x_est;
   p_hat = [p_hat, p_est];

end


hold on
plot(x_hat(1,:)',x_hat(3,:)','r--')
```

Appendix C

Noisy optical sensors and position measurements

```
% Two wheel rover simulation. We test the wheels
% whether they are on soft sand or hard
% ground using epsilon as constant.
% 2D Compass simulation, Noisy Optical Sensor
% measurements simulation, Position measurements
% simulation

%Initialize unchanging variables

clear;
clc;
a = 0.1;  %Wheel radius
b = 0.8;  %Wheel axle length

%Position LSV sensor from center of axle
senLSV_X = 0.3 * b;
senLSV_Y = 0.2;
senLoc = [senLSV_X,senLSV_Y];

%Second LSV sensor placement from center of axle

senLSV_X2 = 0.7 * b;
senLSV_Y2 = 0.2;
senLoc2 = [senLSV_X2,senLSV_Y2];


dt = 0.5; %Time step
numsteps =300;
```

```
Ymax = 3; % Upper Boundary of Sand Region
Ymin = 2; % Lower Boundary of Sand Region



w_true1 = 0.55; % True Speed of the left wheel rover
w_true2 = 0.55; % True Speed of the right wheel rover

noiseAmp = 0.017; %Taken from National Instruments in relation to encoder rotation



%Initialize changing variables

w_meas1 = w_true1 + noiseAmp * randn(numsteps,1); % Add noise and disturbances to m

w_meas2 = w_true2 + noiseAmp * randn(numsteps,1);


% speedwheels =  repmat([w_meas1,w_meas2],numsteps,1);

speedwheels = [w_meas1,w_meas2];


pose = zeros(length(speedwheels),4);

eps1=zeros(numsteps,1);
eps2=zeros(numsteps,1);

degPerRad = 180/pi;   %Conversion degrees to radian
RadPerDeg = 1/degPerRad;

start = [0,0,0,40*RadPerDeg]; % Start position and initial pose of the rover, degre

pose(1,:)= start;



% The beginning of the for loop to

for i = 2:1:(numsteps)


t =  pose(i-1,1);
x = (pose(i-1,2));
y = (pose(i-1,3));
```

```
theta = pose(i-1,4);

  omega1 = speedwheels(i,1); % Speed of wheel1
  omega2 = speedwheels(i,2); % Speed of wheel2


% Position coordinates for wheel1 and wheel2
x1 = x - (b/2) * sin(theta);
y1 = y + (b/2) * cos(theta);
x2 = x + (b/2) * sin(theta);
y2 = y - (b/2) * cos(theta);


% Testing the sandy region by setting upper and lower limits
if (Ymin < y1)& (y1 < Ymax)

     eps1(i-1)= 0.5; % Set 0.5 as a constant for sandy soil

else
     eps1(i-1) = 1;  % Set 1 as a constant for hard ground

end

if (Ymin < y2)& (y2 < Ymax)

     eps2(i-1)= 0.5;

 else
     eps2(i-1) = 1;
 end

epsvalues(i,:) = [eps1(i-1),eps2(i-1)]; %Store epsilon values in a vector


v(i-1) = ( omega1 + omega2)*0.5*a; %Calculate forward velocity
w(i-1)= -( omega1 - omega2)*(a/b);%Calculate orientation, theta

vx(i-1) = v(i-1)*eps1(i-1) +  senLSV_X * w(i-1)*eps1(i-1); % horizontal velocity
vy(i-1) = senLSV_Y * w(i-1)*eps1(i-1); % vertical velocity

vx2(i-1) = v(i-1)*eps1(i-1) +  senLSV_X2 * w(i-1)*eps1(i-1); % horizontal velocity
vy2(i-1) = senLSV_Y2 * w(i-1)*eps1(i-1); % vertical velocity

% heading(i-1)=atan(y/x)*180/pi; %Heading/Compass calculation

X_posSens = senLSV_X ;
```

```
Y_posSens = senLSV_Y ;


pose(i,1) = pose(i-1,1)+ dt; %increment in time
pose(i,4) = pose(i-1,4) + (dt * w(i-1)); % Calculate in timesteps , x positon
pose(i,2) = pose(i-1,2)+ (dt * v(i-1)) * cos(pose(i-1,4)); % Calculate in timesteps
pose(i,3) = pose(i-1,3)+ (dt * v(i-1)) * sin(pose(i-1,4)); % Calculate in timesteps



wheel1(i,:)= [x1,y1]; % Save wheel1 coordinates values
wheel2(i,:)= [x2,y2]; % Save wheel2 coordinates values

M2{1} = wheel1(:,1);  % Save wheel1 x data to cell array
M2{2} = wheel1(:,2);  % Save wheel1 y data to cell array

M3{1} = wheel2(:,1);  % Save wheel2 x data to cell array
M3{2} = wheel2(:,2);  % Save wheel2 y data to cell array

% Save x, y, t values
 sav(i)=  [t];
 sav1(i)= [y];
 sav2(i)= [x];

%Sensor Velocities
vValues =  [v(:)];
vxValues = [vx(:)];
vx2Values =[vx2(:)];
vyValues = [vy(:)];
vy2Values= [vy2(:)];
time(i-1)= [t];

heading(i-1)=atan(sav1/sav2)*180/pi; %Heading/Compass calculation


end

%Plot the results

figure(1);
plot(M2{1,1},M2{1,2},'g.',M3{1,1},M3{1,2},'g.');
xlabel('X position','FontSize',16);
ylabel('Y position','FontSize',16);
title('Two wheel trajectory for Rover','FontSize',16);
text(2*pi/4,sin(3*pi/4),...
            '   Apparent Trajectory',...
```

```
     'FontSize',14)

figure(2);
subplot(2,1,1);
plot(sav,sav2);
xlabel('Time in seconds');ylabel('Position x, in meters');
title('Position in Time','FontSize',16);
hold on
subplot(2,1,2);
plot(sav,sav1);
xlabel('Time in seconds');ylabel('Position y, in meters');
hold on
hold off

figure(3);
plot(time,vValues,'gO');
hold on
plot(time,vxValues,'r*')
hold on
plot(time,vx2Values,'b-')
hold on
xlabel('Time in seconds','FontSize',16);
ylabel('Velocity (meters/sec)','FontSize',16);
title('Sensor Measurements','FontSize',16);
text(2*pi/4,sin(3*pi/4),...
            'Noisy measurements',...
     'FontSize',14)
hleg1 = legend('Encoders','Optical Sensor 1 ','Optical Sensor 2',...
               'Location','SouthEast');

figure(4)
plot(heading);
xlabel('Time in seconds','FontSize',16);
ylabel('Heading (deg)','FontSize',16);
title('2D Compass','FontSize',16);
% compass(pose(i,3));

figure(5);
plot(time,vValues,'gO');
hold on
plot(time,vyValues,'r*')
hold on
plot(time,vy2Values,'b-')
hold on
xlabel('Time in seconds','FontSize',16);
ylabel('Velocity (meters/sec)','FontSize',16);
```

```
title('Sensor Measurements','FontSize',16);
text(2*pi/4,sin(3*pi/4),...
            'Noisy measurements',...
     'FontSize',14)
hleg1 = legend('Encoders','Optical Sensor 1 ','Optical Sensor 2',...
               'Location','SouthEast');
```

Appendix D

Two wheel trajectory simulation

```
% We test the wheels
% whether they are on soft sand or hard
% ground using epsilon as constant. The red
% in the figures is the soft sand and green
% is the hard ground


%Initialize unchanging variables

clear;
clc;
a = 0.1;  %Wheel radius
b = 0.8;  %Wheel axle length

dt = 0.5; %Time step
numsteps =300;

Ymax = 3;
Ymin = 2;



% speedwheel1 = 0.55  %Omega one
% speedwheel2 = 0.55  %Omega two

%Initialize changing variables

speedwheels =  repmat([0.55,0.55],numsteps,1);

pose = zeros(length(speedwheels),4);
```

```
eps1=zeros(numsteps,1);
eps2=zeros(numsteps,1);

degPerRad = 180/pi;   %Conversion degrees to radian
RadPerDeg = 1/degPerRad;

start = [0,0,0,40*RadPerDeg]; % Start position and initial pose of the rover, degre

pose(1,:)= start;



% The beginning of the for loop to
% Nval

for i = 2:1:(numsteps)

t =  pose(i-1,1);
x = (pose(i-1,2));
y = (pose(i-1,3));
theta = pose(i-1,4);

  omega1 = speedwheels(i-1,1); % Speed of wheel1
  omega2 = speedwheels(i-1,2); % Speed of wheel2


% Position coordinates for wheel1 and wheel2
x1 = x - (b/2) * sin(theta);
y1 = y + (b/2) * cos(theta);
x2 = x + (b/2) * sin(theta);
y2 = y - (b/2) * cos(theta);

% Testing the sandy region by setting upper and lower limits
if (Ymin < y1)& (y1 < Ymax)

     eps1(i-1)= 0.5; % Set 0.5 as a constant for sandy soil
 else
     eps1(i-1) = 1;  % Set 1 as a constant for hard ground

end

if (Ymin < y2)& (y2 < Ymax)

     eps2(i-1)= 0.5;

 else
```

```
     eps2(i-1) = 1;
 end

epsvalues(i,:) = [eps1(i-1),eps2(i-1)]; %Store epsilon values in a vector



v = ( omega1*eps1(i-1) + omega2*eps2(i-1))*0.5*a; %Calculate forward velocity
w = -( omega1*eps1(i-1) - omega2*eps2(i-1))*(a/b);%Calculate orientation, theta



pose(i,1) = pose(i-1,1)+ dt; %increment in time
pose(i,4) = pose(i-1,4) + (dt * w); % Calculate in timesteps , x positon
pose(i,2) = pose(i-1,2)+ (dt * v) * cos(pose(i-1,4)); % Calculate in timesteps,y po
pose(i,3) = pose(i-1,3)+ (dt * v) * sin(pose(i-1,4)); % Calculate in timesteps,orie


wheel1(i,:)= [x1,y1]; % Save wheel1 coordinates values
wheel2(i,:)= [x2,y2]; % Save wheel2 coordinates values

M2{1} = wheel1(:,1);  % Save wheel1 x data to cell array
M2{2} = wheel1(:,2);  % Save wheel1 y data to cell array

M3{1} = wheel2(:,1);  % Save wheel2 x data to cell array
M3{2} = wheel2(:,2);  % Save wheel2 y data to cell array

epsvalues(i,:) = [eps1(i-1),eps2(i-1)];


d = [pose(:,2)];
e = [pose(:,3)];


 sav(i)= [t];
 sav1(i)= [y];
 sav2(i)= [x];
end


%Plot the results

figure(1);
xlabel('X position','FontSize',16);
ylabel('Y position','FontSize',16);
title('Two wheel trajectory for Rover','FontSize',16);
```

```
text(2*pi/4,sin(3*pi/4),...
            'Red - Sand Region = 0.5',...
     'FontSize',14)
plottycolor;

hold on

figure(2);
xlabel('X position','FontSize',16);
ylabel('Y position','FontSize',16);
title('Trajectory for Rover','FontSize',16);
text(2*pi/4,sin(3*pi/4),...
            'Red - Sand Region = 0.5',...
     'FontSize',14)
plotcolor;
```

Appendix E

Partial data for Laser Speckle Velocimetry

```
time vx vy sdev_x sdev_y
0.000 0.127 4.20E-02 3.00 1.00 0.248 0.052
0.030 -0.254 4.20E-02 -6.00 1.00 0.248 0.052
0.060 0.296 -8.40E-02 7.00 -2.00 0.248 0.051
0.100 -0.212 -4.20E-02 -5.00 -1.00 0.245 0.050
0.130 -0.212 -4.20E-02 -5.00 -1.00 0.244 0.049
0.160 0.254 4.20E-02 6.00 1.00 0.248 0.049
0.200 -0.381 4.20E-02 -9.00 1.00 0.246 0.051
0.240 0.254 0.00E+00 6.00 0.00 0.240 0.050
0.270 -0.127 0.00E+00 -3.00 0.00 0.238 0.051
0.300 -0.296 -4.20E-02 -7.00 -1.00 0.240 0.051
0.330 0.381 0.00E+00 9.00 0.00 0.239 0.052
0.360 -0.339 4.20E-02 -8.00 1.00 0.232 0.052
0.390 0.296 -4.20E-02 7.00 -1.00 0.228 0.052
0.420 -0.042 0.00E+00 -1.00 0.00 0.225 0.052
0.470 -0.042 -4.20E-02 -1.00 -1.00 0.226 0.052
0.500 0.339 4.20E-02 8.00 1.00 0.226 0.052
0.530 -0.339 4.20E-02 -8.00 1.00 0.220 0.052
```