This electronic thesis or dissertation has been downloaded from Explore Bristol Research, http://research-information.bristol.ac.uk

Author:
Malinowski, Mateusz T

Title:
Slip-aware localisation for wheeled vehicles

General rights
Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

Take down policy
Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact collections-metadata@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

Slip-Aware Localisation for Wheeled Vehicles

By

Mateusz Tomasz Malinowski

Department of Aerospace Engineering
University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Doctor of Philosophy in the Faculty of Engineering.

January 2023

Word count: Approx. thirty-nine thousand five hundred

ABSTRACT

Wheeled robots are subject to slip which may cause errors in position estimation if not correctly observed. Wheel slip is typically considered only by a path-following controller. An offline-calibrated slip value can also compensate for wheel odometry (WO). This thesis proposes a new method of fusing visual odometry (VO) and WO with integrated slip estimation using the Extended Kalman Filter. The approach handles the correlation between the slip and the rover's position estimation and occasional errors in any VO or WO measurements. Furthermore, it is possible to tune the model to emphasise WO (e.g. when no slip is expected and thus reduces the number of VO measurements) or rely more on VO (high slippage variability).

An adaptive filter is introduced to tune both process and measurement noises automatically based on available measurements. It enables the assessment of noise statistics at each filter's iteration, leading to the design of a reactive VO scheduling algorithm. The solution does not rely on prior knowledge of the environment and offers a route to conserve computational resources while maintaining good navigational accuracy.

The localisation system can be further improved if the wheel slip is known in advance. This prior information can be exploited by investigating how the wheel slip predictions, derived, for example, from forward-facing vision or motor current, can be fused within the model using different slip prediction schemes. The solutions provide improved localisation accuracy and hint at the intriguing possibility of slip-based SLAM.

The above research scope was reduced to one dimension to lower the complexity and focus on the principles of new algorithms. However, to advance some elements described in this thesis, the adaptive filter with integrated slip estimation and reactive VO scheduling is expanded into two dimensions. The results demonstrate improved navigational performance but highlight the need for more work in the future.

DEDICATION AND ACKNOWLEDGEMENTS

Firstly, I would like to thank my academic supervisor, Prof. Arthur Richards, for his support and guidance. When I started this project, I thought that having several years of experience would enable me to tackle all problems quickly. I could not have been more wrong! It was all due to discussions we had, perhaps fierce on some occasions, that I found most beneficial and which allowed me fully understand the purpose of a PhD. Advancing human knowledge is one thing, but the new mindset I found is what I most appreciate. Thank you!

I would not have the opportunity to pursue this PhD if not for SCISYS UK Ltd. and the former Director of Space UK division, Dr Spencer Ziegler, who made it happen - I am truly grateful. Furthermore, I am thankful to CGI IT UK Ltd., who, after acquiring SCISYS, let me finish my PhD. Many things may change with a new employer, but I was fortunate to continue my research without interruptions.

Combining work with studies is not always easy, and I found it challenging. To maintain my work-life balance, I needed every tiny pinch of flexibility where possible, and luckily I was always able to get that from my Line Manager and Industrial Supervisor, Dr Mark Woods. Thanks, Mark, for helping me balance this thin line with a relaxed atmosphere at work, support for my PhD, and growing friendship.

My eternal thanks to my fantastic parents, Maria and Andrzej. You shaped me into who I am today. You let me explore life when I was young to understand what I wanted to do. I am proud, lucky, and perpetually grateful for having you, and I know you feel the same. And many thanks to my sister, Agnieszka. You were my role model, and I always looked up to you. I would not have begun a PhD or experienced living abroad if I had not witnessed your trips and the bags of experience you were always coming with. You are precisely the person I needed and still need as a sister. Thank you.

All in all, I would have failed miserably so many times on so many levels if I did not have constant and daily support from my beloved wife, Dominika. You were the one who had to listen to me when I needed to express my doubts, frustrations, and fears, but also when I was hyper-enthusiastic when I managed to get something working and was talking over and over like a parrot. The pandemic did not help either, as you were the only one at that time I could talk to face to face. You were there for me, you are still, and I know I can count on you to be by my side in the future. Thank you for your unconditional love. And I am so grateful for all your effort during the pregnancy and birth. Kostek, you were just born when I was finishing this thesis. I am elated and proud to have you, and I hope that you will read this thesis one day so that we may discuss it together.

Finally, I would like to thank all who were, are, and will be part of the 6^{th} Sea Scout Group from Rybnik, Poland, which I was privileged to lead at some point. Being a scout helped shape me by motivating me to pursue challenges, and this PhD was one of the biggest challenges I had to date.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE: ....

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS AND ACRONYM

1D	One-Dimensional
2D	Two-Dimensional
3D	Three-Dimensional
AEKF	Adaptive Extended Kalman Filter
ALMF	Adaptive Limited Memory Filter
CNN	Convolutional Neural Network
CNSA	China National Space Administration
CPU	Central Processing Unit
DARPA	Defence Advanced Research Projects Agency
DGPS	Differential Global Positioning System
EKF	Extended Kalman Filter
ESA	European Space Agency
FPGA	Field Programmable Gate Arrays
GNC	Guidance, Navigation, and Control
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GPU	Graphics Processing Unit
HiRISE	High Resolution Imaging Science Experiment
ICP	Iterative Closest Point
IMU	Inertial Measurement Unit
LiDAR	Light Detection And Ranging
MER	Mars Exploration Rover

MSL	Mars Science Laboratory
NASA	National Aeronautics and Space Administration
POSE	Position and Orientation State Estimate
RANSAC	Random Sample Consensus
RMSE	Root Mean Square Error
RTK	Real-Time Kinematic
SFR	Sample Fetch Rover
SLAM	Simultaneous Localisation And Mapping
SVM	Support Vector Machine
TIO	Thermal-Inertial Odometry
ToF	Time-of-Fligh
UKF	Unscented Kalman Filter
VIO	Visual-Inertial Odometry
VO	Visual Odometry
VPU	Vision Processing Unit
WO	Wheel Odometry

NOMENCLATURE

$\mathbb{1}$	Identity matrix	
B_i	Width-wise coordinate of the $i'th$ wheel in the local frame	m
F	State transition Jacobian matrix	
G	Process noise Jacobian matrix	
H	Observation Jacobian matrix	
I	Moment of inertia	kg m m
K	Kalman filter gain	
M	Total mass of the vehicle	kg
M_i	Mass at $i'th$ wheel	kg
N	Window size in ALMF algorithm or gain in Sage-Husa algorithm	
N_Q	Window size or gain for adapting process noise	
N_{VO}	Window size or gain for adapting VO measurement noise	
N_{WO}	Window size or gain for adapting WO measurement noise	
P	Covariance of the state estimate x	
P_{xx}	Covariance of x position estimate	m m
P_{yy}	Covariance of y position estimate	m m
Q	Covariance matrix of process noise w	
R	Covariance matrix of measurement noise n	
R_{VO}	Covariance matrix of VO measurement noise n_{VO}	
R_{WO}	Covariance matrix of WO measurement noise n_{WO}	

R_ϕ	Radius of the Ackermann turn	m
S	Covariance matrix of the measurement residual r	
T_0	Time constant of the low-pass filter	s
$T_{VO_{MAX}}$	Maximum allowed VO measurement period	s
T_{VO}	VO measurement period, i.e. the time between two consecutive measurements	s
Λ	Slip profile consisting of multiple profile point Λ_i	
Λ_i	A profile point from Λ that consists of a position value p_i and wheel slip λ_i associated with that point	
Ω	Angular velocity of wheels (rate)	rad / s
Ω_i	Angular velocity of $i'th$ wheel	rad / s
Θ	Accumulated wheels angle	rad
λ	Wheel slip on the longitudinal axis	-
λ_ψ	Angular wheel slip in the local frame	rad
λ_x	Longitudinal wheel slip in the local frame	-
λ_y	Lateral wheel slip in the local frame	-
ω	Angular velocity of the vehicle (motion in a plane)	rad / s
ϕ_i	Steering angle of the $i'th$ wheel	rad
ψ	Orientation of the vehicle in the global frame	rad
ρ	Wheel radius	m
σ	Standard deviation of normal distribution	
σ_{VO}	Standard deviation of VO measurement noise n_{VO}	
σ_{WO}	Standard deviation of WO measurement noise n_{WO}	
σ_α	Standard deviation of process noise w_α	m / s / s
σ_β	Standard deviation of process noise w_β	-
σ_{λ_ψ}	Standard deviation of process noise w_{λ_ψ}	rad
$\sigma^2_{\lambda_x\lambda_y}$	Covariance between w_{λ_x} and w_{λ_y}	-

σ_{λ_x}	Standard deviation of process noise w_{λ_x}	-
σ_{λ_y}	Standard deviation of process noise w_{λ_y}	-
σ_{λ}	Standard deviation of n_{λ} measurement noise	-
c	Steering curvature	1 / m
d^2	Squared Mahalanobis distance	
f	State transition function	
h	Observation function	
k	Time instant in the discrete model that is a multiple of Δt	s
l_A	Distance from the centre of a vehicle to its front	m
l_i	Length-wise coordinate of the $i'th$ wheel in the local frame	m
l_{MAX}	The far distance in front of the robot for slip profile prediction	m
l_{MIN}	The near distance in front of the robot for slip profile prediction	m
m	Augmented state that stores the pose of the last VO measurement	
n	Measurement noise	
n_{VO}	VO measurement noise	
n_{WO}	WO measurement noise	
n_{λ}	Measurement noise of λ measurements	-
p	Current longitudinal position	m
p_x	Position of the vehicle on X axis in the global frame	m
p_y	Position of the vehicle on Y axis in the global frame	m
r	Measurement residual	
Δt	Duration of a single discrete time step	s
u	Control input	
u_c	Commanded steering curvature	1 / m
u_v	Commanded linear velocity on X axis in the local frame	m / s

u_Ω	Commanded wheels angular velocity	rad / s
u_λ	Wheel slip on the longitudinal axis treated as pseudo-control input	-
u_ϕ	Steering angle of an ideal steering wheel in bicycle model	rad
v	Linear velocity on the longitudinal axis	m / s
v_x	Linear velocity of the vehicle on X axis in the local frame	m / s
v_y	Linear velocity of the vehicle on Y axis in the local frame	m / s
w	Process noise	
w_c	Process noise that captures uncertainty of input u_v	1 / m
w_v	Process noise that captures uncertainty of input u_v	m / s
w_α	Process noise component acting as wheels angular acceleration	m / s / s
w_β	Process noise component driving wheel slip estimation	-
w_γ	Process noise component related to the uncertainty of pseudo slip profile Λ	-
w_{λ_ψ}	Process noise component driving angular wheel slip estimation	rad
w_{λ_x}	Process noise component driving longitudinal wheel slip estimation	-
w_{λ_y}	Process noise component driving lateral wheel slip estimation	-
x	State of the system	
z	Observation / measurement	
z'_{VO}	Accumulated VO measurement	
$z_{VO_{acc}}$	Accumulated predicted relative VO measurement	
z_{VO}	Relative VO measurement (difference in pose between two stereo image pairs)	
z_{WO}	WO measurement	
z_λ	Measurement of the longitudinal wheel slip λ	-
z_{ϕ_i}	Measurement of $i'th$ wheel steering angle	rad
$\bar{a}$	Mean value of a	
$\hat{a}$	*a posteriori* estimate of a	
$\tilde{a}$	*a priori* estimate of a	

CHAPTER 1

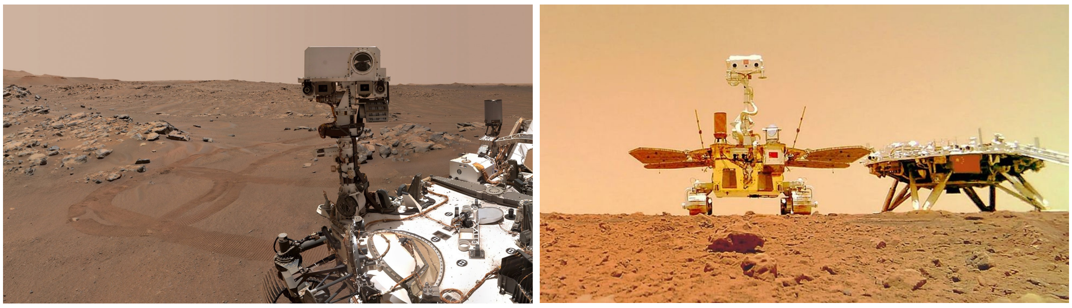
INTRODUCTION

This thesis proposes a novel method of reducing the frequency of processing compute-intense Visual Odometry (VO) without sacrificing navigational accuracy. It is mainly achieved by integrating wheel slip within the sensor fusion algorithm and adapting process and measurement noises to determine when the subsequent VO measurement is required. The thesis also discusses, to some extent, how predicted wheel slip could be used to improve localisation accuracy further. Most work is done in one dimension with an initial approach of addressing planar motion.

1.1 Motivation

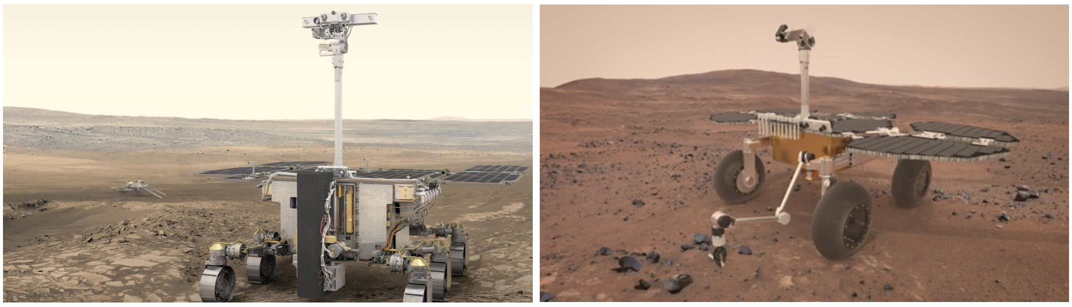
It is an exciting decade for planetary exploration. With the National Aeronautics and Space Administration's (NASA) Perseverance [1] (Fig. 1.1(a)) and China National Space Administration's (CNSA) Zhurong [2] (Fig. 1.1(b)) landing on Mars in 2021, we could see the highest surge in planetary science since Mars Science Laboratory (MSL) Curiosity. Even though the European Space Agency's (ESA) ExoMars Rosalind Franklin rover (Fig. 1.1(c)), initially scheduled to launch in 2022, has been suspended due to Russia's invasion of Ukraine [3], there are still plans to launch ESA's Sample Fetch Rover (SFR) (Fig. 1.1(d)), due in 2026, as part of Mars Sample Return (MSR) mission that aims to bring a sample of Mars back to Earth [4]. And not to mention returning to the Moon with the Artemis program in 2024 [5], which may require a whole fleet of rovers.

SFR may become an exceptionally challenging mission, as it will have to collect all samples dropped by Perseverance in a limited time due to the global dust storm season. To achieve this, SFR will need to drive long distances at higher speeds than previous missions (~200 m per sol – a Martian day) [6]. It can become even more challenging for SFR, as it will be equipped only with solar panels, which may constrain its operation to only a few hours around noon.



(a) Perseverance, ©NASA/JPL-Caltech/MSSS

(b) Zhurong, ©CNSA



(c) ExoMars (Visualisation), ©ESA/ATG Medialab

(d) SFR (Visualisation), ©ESA/ATG Medialab

Figure 1.1: Example of Mars rovers

To make most of the time around the noon, SFR's nominal speed is expected to be 0.0667 m/s (early indication in [7]) which would allow covering the necessary daily distance in about 50 minutes. For comparison, a typical speed of MSL Curiosity with hazard avoidance and VO is only 0.0056 m/s compared to 0.039 m/s for a blind drive. The difference comes from the requirement of Curiosity to stop for VO [8]. Reasons for these stops are not specified, but one may guess that i) acquiring images while stationary reduces possible blur coming from vibrations, and ii) there are constraints on processing resources. Interestingly, the longest distance travelled by a Mars rover belongs to Perseverance - 319.79 m [9], however, at the moment, it is unknown whether it was a blind drive or with all autonomy functions similar to those of Curiosity. One has to be careful when performing any type of drive on Mars. Sandy terrain poses a real risk to rovers and their wheels. They may get stuck in soft sand as was the case for MER Spirit, or get damaged during traverse as it was for Curiosity. The latter relates to wheel slippage and even though engineers work on optimising wheel design to reduce the slippage effect, in general, it cannot be avoided. Interestingly, the ExoMars rover will be equipped with a unique wheel-walking mechanism that offers walking capability should a wheel got stuck in the sand. Other rovers, however, do not possess this feature and therefore investigating ways to reduce slippage is one of the motives for this research.

There is a primary risk that the required distances will not be covered, especially if more energy is required to deviate from the preferred path. Secondly, it leaves little time to analyse the surrounding environment making energy and resource management key for this mission. Even though the environment would have been assessed already by Perseverance and scientists prior to the SFR launch, the rover may approach the terrain differently, giving a different perspective for images. Although not its primary function, the additional scientific output would be very much welcomed. This, however, is a challenge because there is little free time available on the co-processor for any other processing when driving to collect samples. But what if VO were done less often? Could it allow for other algorithms to take advantage of freed computational resources? And what would be the impact of fewer VO measurements?

This thesis investigates a fusion of VO and Wheel Odometry (WO) using Extended Kalman Filter (EKF) to propose a novel method to reduce the number of VO measurements by analysing process noise and measurement noise statistics that enable slip-based estimation. The problem is analysed in one dimension with some elements expanded into two dimensions. By one dimension it is assumed a longitudinal/heading direction, i.e. movement in all directions is allowed but only the total distance travelled and longitudinal wheel slip are analysed. The research question to answer is: **Can VO frequency be reduced without sacrificing navigational accuracy?**

It is best illustrated by Fig. 1.2. Insufficient VO measurements in a high-slip area lead

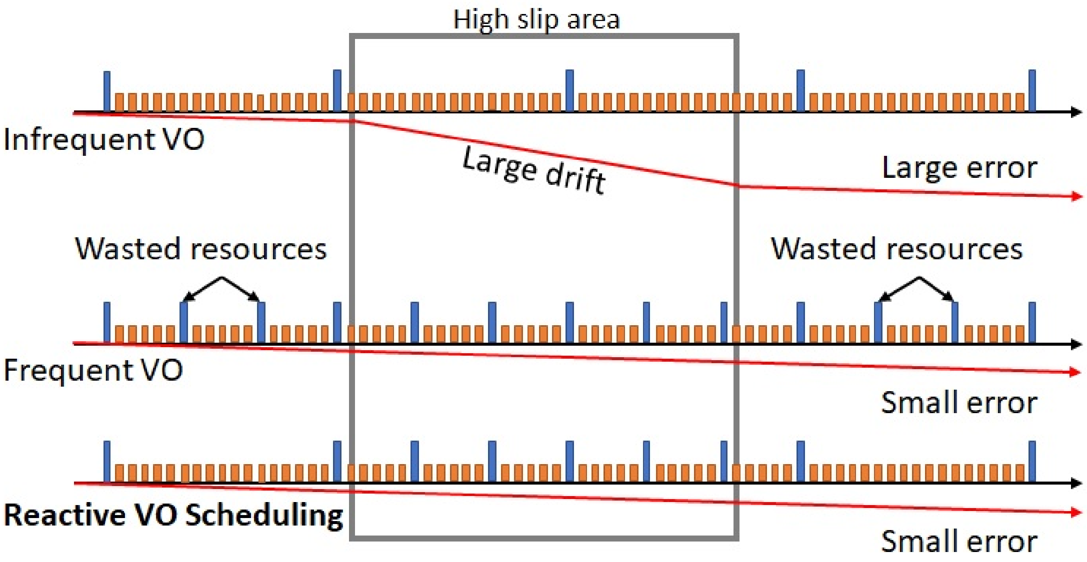
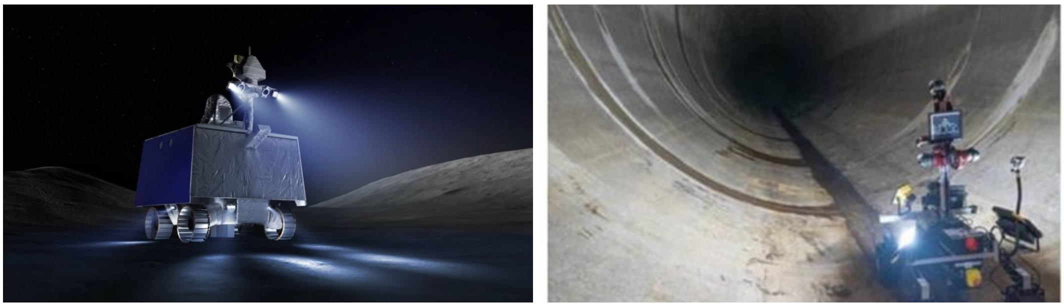


Figure 1.2: Pictograph illustrating the hypothetical impact of reduced VO frequency. Blue bars indicate VO measurements, and orange bars WO. When performing infrequent VO measurements, wheel slippage in the high slip area may not be correctly estimated leading to large drift and increased error. On the other hand, frequent VO, even though offers a small error, wastes computational resources when driving in a low slip area. An optimal solution would perform infrequent VO measurements in low slip areas and frequent measurements in the high slip area, benefiting from the advantages of the above examples but not sharing their disadvantages. This hypothesis assumes a good VO algorithm.



(a) Viper (Visualisation), ©NASA/Daniel Rutter (b) SOLO, ©CGI IT UK

Figure 1.3: Example of rovers with lights

to increased drift, however, too many VO measurements, when driving on solid terrain, are considered as wasted resources. The aim is to use VO only where necessary, thus reducing resources used while maintaining good navigational accuracy. For clarity, VO provides a relative motion estimation and cannot correct WO drift after the fact if not used during the traverse.

Reduced VO frequency may indirectly reduce the power usage of a rover when driving on the Moon in permanently-shadowed areas (Fig. 1.3(a)) or in an underground tunnel (Fig. 1.3(b)), the rover will need to carry lights to illuminate the environment for cameras. Even when lights are turned on for the duration of camera exposure, a reduced number of flashes may save enough energy to make several kilometres-long inspections feasible.

1.2 Odometry and Wheel Slip

In the robotic world, typically, there are two types of sensors: proprioceptive and exteroceptive. The former type relates to the robot's internal state (e.g. motors encoders, inertial sensors, battery voltage), whereas the latter type allows estimating the external state (the environment), for example, using cameras or Light Detection and Ranging (LiDAR) sensors. Some researchers also define a third type, interoceptive, which relates to internal engineering quantities used for a system's health monitoring. The general understanding is that proprioceptive sensors offer fast measurements which do not cost much energy, however, they are subject to drift and the environment's variability. On the other hand, exteroceptive sensors can offer measurements that take external states into account at the cost of more computational complexity.

An example of proprioceptive localisation technique in wheeled robots is WO that offers a cheap estimate of the rover's position by counting how far wheels have turned. However, because wheels may slip, WO usually reports further distance than the actual. In the case of Mars, wheel slip can contribute to over 10% of error in position estimation [10]. Although there is a chance for a vehicle to drive further than the WO reports (sliding), it has more impact on cars, which drive at faster velocities than rovers, when they have wheels locked during braking. Typically, WO is accompanied by the Inertial Measurement Unit (IMU). These units are popular among robotics researchers, however, they are subject to drifts. It means that another technique is usually required to correct the IMU. Because in this research we investigate how a wheel slip is estimated and can be used to improve localisation, and because we have an access to an accurate VO algorithm, having an IMU was not deemed necessary. Its benefit in the one-dimensional analysis would be limited yet it would add the complexity of estimating biases. Due to the latter, it would not be enough to use only IMU with WO, it would still require VO or another exteroceptive localisation technique.

When it comes to exteroceptive sensors, LiDARs gain increasing popularity among researchers and engineers due to their simplicity and robustness in low-light conditions. However, space rovers often rely on visual information for navigation, mainly because LiDAR and Time-of-Flight (ToF) cameras are not currently space-rated for Mars exploration and are considered too power and mass inefficient for such missions where both of these are at a premium. The most popular solution is to use VO, which extracts essential information from images in the form of features. These are pixels or clusters of pixels that remain similar across several frames, as long as they are in the camera's field of view (e.g. a specific rock visible along the rover's traverse). Multiple features can be compared across left and right image frames and between current and previous stereo image pairs to provide a rover POSE (Position and Orientation State Estimate). In this work, the employed VO algorithm provides a POSE between two consecutive stereo image pairs. More about an example VO algorithm can be found in [11]. Extracted features can be projected into a 3D map allowing the rover to perceive the environment and thus detect any obstacles and find safe routes. However, processing images on space-qualified hardware takes substantial time. Reducing the number of these operations could potentially lead to lower energy usage and free some computational resources that may be spent on other tasks. For example, with the latest developments in image classification techniques [12], one can envisage an on-board system to prioritise data for downlink or a system for reactive exploration. However, with constant robot speed, the less frequent VO measurements are done, the bigger the measurement error may be. With a bigger distance between two consecutive frames, there is a risk that there would be an insufficient number of features to match resulting in a worse VO estimate.

The best localisation accuracy and the most benefits can be achieved by combining proprioceptive and exteroceptive information. Imagine a walking person to visualise better the effect of slip, WO, and VO. Humans use their eyes (stereo cameras) to analyse the world and localise themselves within it (more complex operations than a simple delta motion). However, when they close their eyes, they can continue walking. They might count how many steps they do (encoders) to localise themselves in their remembered environment (stored map). Nevertheless, it might become harder over time as steps may not necessarily have the same length. Furthermore, if walking on a slippery sand surface, one could slip, introducing errors in the step counting process. They could then open their eyes to correct themselves, and once their position is established, close their eyes again and continue counting steps (infrequent corrections). Often when eyes are closed, other senses are strengthened. Many people close them, even subconsciously, when listening to their favourite music or enjoying that final bite of delicious cake. These senses would not give such vital feedback with their eyes open, which could loosely translate to having free brain resources for other tasks when their eyes (cameras) are inactive. Humans take advantage of processing images less often, so why robots should not?

In robotic applications, wheel slip can be treated as a drift in WO that VO can correct. It is conceptually similar to Visual-Inertial Odometry (VIO), where an inertial sensor (IMU), subject to drift, is combined with visual localisation techniques to offer benefits of both systems: fast IMU update rate with vision's localisation accuracy. However, many modern robotic systems focus primarily on Simultaneous Localisation and Mapping (SLAM) or VIO and do not consider wheel slip. A recent DARPA's (Defence Advanced Research Projects Agency) Subterranean Challenge [13] gives a good overview of the top robotic applications for GPS-denied environments. The top team from the final's Systems Competition, CERBERUS, used walking and flying robots. Even though they did not use wheeled robots, their main approach for localisation was a fusion of VIO, and Thermal-Inertial Odometry (TIO) with LiDAR [14]. Interestingly, they did mention a Leg Odometry [15] for their quadrupedal, however, this is a recursive estimation algorithm that was not referenced in their main paper describing chosen localisation scheme. Also, according to [16], any quadrupedal's slip reflexes were stabilised by internal system [17]. When looking at the final's virtual competition, the second-best team (the winner did not provide any information regarding their solution), CTU-CRAS-NORLAB from the Czech Technical University in Prague, used a combination of wheeled, tracked, crawling, and aerial robots. In [18, 19] authors explain that they used SLAM based on LiDAR using Iterative Closest Point (ICP) method. To provide an initial estimate for ICP, a fusion of wheel odometry and inertial sensors is used, however, no notion of a wheel or track slip is maintained there. Another team that performed well in Tunnel and Urban circuits was CoSTAR from NASA JPL. Their localisation solution follows the same pattern as for previous teams. LiDAR is used for SLAM with IMU, and odometry [20], but without any notion of wheel slip.

It can be safely assumed that for terrestrial applications where there is access to high-grade and accurate LiDARs, SLAM plays a crucial role in robot localisation. There is no need to understand the wheel slip with frequent localisation updates utilising powerful computers. In the case of planetary rovers, wheel slip is primarily used for trajectory control, and it is not often estimated with a sensor fusion framework. In [21] VO measurements "are considered truth", which can be read as no filtering is done, which has the advantages of faster execution on flight hardware, especially when the VO algorithm is deemed to be very precise. As for the slip, it is estimated using true VO and WO measurements. A different approach is presented in [22], where the Indirect form of the EKF is used to estimate state errors for integrated Inertial Measurement Unit (IMU) measurements. It is interesting to note that to better model relative measurements, an augmented state is introduced using an approach called stochastic cloning, which is described in detail in [23]. Slip presence detection is based on [24], and its actual value is then calculated by comparing output from the filter and forward kinematics model. Another approach, presented in [25], estimates slip value using only IMU and WO measurements, whereas the EKF, which also fuses VO, is employed to provide position estimation.

The above solutions provide slip estimation outside Kalman Filter; therefore, there is no notion of how good this estimation is. An integrated solution is presented in [26] where a single longitudinal wheel slip was included in Unscented Kalman Filter's (UKF) state vector when modelling a six degree of freedom movement. Initial results proved better performance compared to a model without slip estimation. Similar work is presented in [27], where a particle filter is used to model the plane motion of a car with four longitudinal slip values being estimated (one per wheel). A different approach to estimate slip is presented in [28], where an EKF is used to estimate the platform's position, orientation and velocity, and a linear Kalman Filter is used to estimate the slip alone. Still, this approach does not capture any correlation between position and slip.

Such correlation can be beneficial to fully understand energy cost of robot's traverse and to support path-planning that would avoid high-slip areas. On one end, work presented in [29] assesses the power usage of a traverse based on the images from forward-facing cameras. The higher the expected slippage, the more energy is required for a drive. In this work, however, the slip is not directly extracted, and therefore it does not support the localisation function. On the other hand, in [30] different sensor suits were proposed for different terrain types to reduce power footprint. It was done purely on vision-based environment assessment, and the sets of sensors were pre-configured, not allowing for dynamic adjustment of, for example, the VO framerate.

1.3 Computational Constraints

Space exploration is known for its limited computational power. It is mainly a result of the expensive and lengthy development of radiation-hardened hardware followed by rigorous testing. As a result, computers specification are orders of magnitude lower than terrestrial equivalents even at the design stage and will be even more so by the time flight occurs, which is often many years later. To put that into perspective, Perseverance has a processor with a clock speed of 200 MHz [31], whereas MER rovers Spirit and Opportunity had processors with a clock speed of only 20 MHz [10]. Because of that, a typical VO operation required Spirit, Opportunity, and Curiosity to stop, acquire images, and wait until the position was estimated [8]. Whether Perseverance is following the same workflow is yet to be seen when more detailed information about the mission is published, however, in one of their latest videos, NASA claims that continuous drive was achieved at the speed of approximately 0.045 m/s (0.1 miles per hour) [32].

With limited computational power, VO could take a significant amount of time to process. In the case of ExoMars rover Rosalind Franklin, the requirement is to process VO updates under 4.25 s [33] every 10 s [21] using a 96 MHz Leon2 co-processor [34]. SFR will be equipped with a more powerful 250 MHz Leon4 co-processor [35]. It is expected to process a VO update in under one second every two seconds. The non-linear improvement in processing speed is a result of not only a faster clock but also better memory and Level-2 cache available on Leon4. Nevertheless, VO will utilise 50% of a single core, which could have been used for other tasks.

Processors are not the only type of hardware available for space missions to perform complex operations. Recently, the Field Programmable Gate Arrays (FPGA) have become increasingly popular due to their better efficiency in floating-point operations per second per watt. FPGAs are great at operations that require parallelism, hence their suitability for image processing. With the latest development in the field, large FPGAs are big enough to support visual SLAM [36] or even Convolutional Neural Network (CNN) to enable objects identification from images [37]. An FPGA is also considered for SFR to support its initial localisation function, SPARTAN [38], however, it is understood that at the time of writing this thesis, SPARTAN competes with VisLoc [33] (the same VO solution designed for ExoMars) and there are no publicly available documents to indicate which algorithm was selected for SFR.

Because FPGAs require a different approach to programming than software, they require a unique set of skills to handle. One of the solutions that offer an easy to use microprocessor for artificial intelligence suitable in the space environment is the Intel Myriad 2 chip (also called Vision Processing Unit (VPU)), which runs a pair of Leon4 controllers [39] among other improvements. Currently, it is in PhiSat-1 CubeSat orbiting Earth, where it acts as a proof of concept for applicability of Intel Myriad 2 for space applications [40]. Furthermore, an improved version, Myriad X, has already been released for terrestrial applications, which has enormous potential by being a dedicated device on a simple USB stick called Intel Neural Compute Stick 2.

Not for the space industry yet, the most powerful devices for parallel operations are Graphics Processing Units (GPUs). The number of applications in the robotics world that utilise these devices is growing rapidly. When writing this thesis, Nvidia remains the leader in providing embedded GPUs - Nvidia Jetson. Recently, radiation testings were performed on Nvidia Jetson TX2 by NASA [41] and on Nvidia Jetson Nano by US Air Force [42]. Both concluded that they could be suitable for Low Earth Orbit, with the second investigation quoting 1.5 to 2 years of operation before Jetson receives its maximum dose of radiation. It does not rule out the possibility of launching Jetson products to the Moon for much shorter missions, however, at this stage, it seems unlikely to use them for long-duration exploration missions.

It is still possible for Nvidia Jetson to make it to the Moon or Mars as a technology demonstration, similar to NASA Mars helicopter - Ingenuity. The helicopter is equipped with a Qualcomm Snapdragon 820 processor, which can also be found in many modern smartphones. It allows Ingenuity to process monocular Visual-Inertial Odometry at 30 Hz framerate [43]. It is a remarkable achievement to launch such modern hardware, however, it comes with some caveats. Because

Ingenuity is a technical demonstration, its design was less constrained by rigid requirements. It nicely shows the possibilities of having a fast processor, but it is unlikely to see Snapdragon more often in deep space. At least, not on its own. Some may expect that future missions could be equipped with two processors: one modern to enable fast processing and data generation on time, and another space-qualified to which the system would switch if the other got too damaged by the radiation. There would be a disadvantage of flying two processors instead of one, but it depends on the trade-off analysis during mission design.

1.4 Reasons for One Dimension: Heading Direction

Intuitively, localising in only a heading direction has minimal applications and might be hard to apply to real-world problems. However, there are several that could benefit from it. The most obvious is pipe inspection, where robots are localising themselves on the axis along a pipe [44]. Other places where a one-dimensional localisation is present in the form of mileposts are motorways (driver location signs every 500 m) and railways (based on section 94 of the Railways Clauses Consolidation Act 1845, the posts have to be erected every quarter of a mile). Interestingly, mileposts in the XIX century were practical for engineers and surveyors and helped determine how much passengers should pay for their trip based on travelled distance.

This total distance along an imaginary line (usually centre line) in structural engineering is called chainage and is widely used in the tunnel industry. The term was coined in the XVII century when Edmund Gunter introduced 66 foot long chains to measure distance (80 chains would measure a mile). Nowadays, chainage can still be read in plans, documents, and reports [45–47]. From a human surveyor perspective, it is easy to understand. For example, a detected defect could be located X metres from the centre of the shaft at nine o'clock. Going to three dimensions for a straight tunnel with a known diameter D would mean X metres forward, D/2 metres to the left, and D/2 metres up. It may not be as easy to explain for tunnels that turn, and hence chainage coordinates are preferred.

Finally, it is not only simpler to explore an idea in one dimension, but also essential to validate theoretical concepts in lower dimensions. If it does not work, it will not work in higher dimensions. It is similar to the technology readiness level, where a proposal is matured from a paper-based study through proof of concept and experiments to industrialisation and operation. Failure at any stage prevents moving to the next one. At the same time, achieving a certain level does not guarantee that the next level can also be achieved. In this work, when analysing movement in the heading direction, only longitudinal wheel slip is considered. Lateral and angular slip movements are only considered in chapter 5.

1.5 Review of Extended Kalman Filter

This section provides basic information about discrete EKF. EKF is a heuristic for non-linear systems which linearises dynamics and observation functions at the current estimate. Readers familiar with the subject can skip it to the next section.

First, consider a stochastic, non-linear, time-varying dynamics system modelled by the state transition function f and observation function h defined as

$$x(k+1) = f\big(x(k), u(k+1), w(k)\big) \tag{1.1a}$$

$$z(k) = h\big(x(k), u(k), n(k)\big) \tag{1.1b}$$

Where $x(k)$ represents the system state at the time k, u is the vector of control input, z is the observation (measurement), and w and n are process noise (also known as the disturbance input) and measurement noise respectively. Both noises are assumed to be zero-mean Gaussian random sequences. Their covariance matrices at time k are known (Q for process noise and R for measurement noise), however, they remain uncorrelated in time and with each other

$$E[w(k)w(j)^T] = \begin{cases} Q(k) & j = k \\ 0 & \text{otherwise} \end{cases} \tag{1.2a}$$

$$E[n(k)n(j)^T] = \begin{cases} R(k) & j = k \\ 0 & \text{otherwise} \end{cases} \tag{1.2b}$$

$$E[w(k)n(j)^T] = 0 \qquad \text{for all } k \text{ and } j \tag{1.2c}$$

It is assumed that Q is at least positive semi-definite, whereas R is a positive-definite matrix. The reason for that is explained further.

Kalman Filter, and thus EKF, is a minimum mean-square error estimator. Its aim is to minimise the expected value of the square state error $E[||x(k) - \hat{x}(k)||^2]$, where $\hat{x}(k)$ is the *a posteriori* state estimate based on all observations up to and including at the time k. The problem is equivalent to minimising the trace of the *a posteriori* estimate covariance matrix $\hat{P}$ defined as

$$\hat{P}(k) = E[\big(x(k) - \hat{x}(k))(x(k) - \hat{x}(k)\big)^T] \tag{1.3}$$

The algorithm is recursive, which means that the new estimate is based only on the previous step. The filter works by first forming a prediction state $\tilde{x}$ (also known as the *a priori* estimate) based on dynamics model defined as

$$\tilde{x}(k+1) = f\big(\hat{x}(k), u(k+1), w(k)\big) \tag{1.4}$$

The *a priori* estimate covariance matrix $\tilde{P}$ is given by propagation

$$\tilde{P}(k+1) = F(k+1)\hat{P}(k)F^T(k+1) + G(k+1)Q(k)G^T(k+1) \tag{1.5}$$

Where F and G are Jacobians defined as

$$F(k+1) = \frac{\partial f}{\partial x}|\hat{x}(k), u(k+1), \bar{w}(k) \tag{1.6a}$$

$$G(k+1) = \frac{\partial f}{\partial w}|\hat{x}(k), u(k+1), \bar{w}(k) \tag{1.6b}$$

Note that $\bar{w}$ refers to the non-zero mean of the process noise. It is only considered in the adaptive filter discussed in chapter 3. However, for completeness, the possibility of a non-zero mean is highlighted in (1.6).

Once the state and its covariance are propagated based on the dynamics model, they can be corrected using observations. The first step in the filter's correction stage is to estimate the innovation, which is the difference between the measurement and its prediction

$$r(k) = z(k) - E[h(\tilde{x}(k), u(k), n(k))] = z(k) - h(\tilde{x}(k), u(k)) \tag{1.7}$$

The whitening filter provides an innovation r which is statistically equivalent to the observation z with its associated measurement noise n. The innovation sequence is generally considered a white noise with zero mean value. However, due to the statistical equivalence between the innovation and observation, should the observation noise n be biased, the innovation r would also have a bias of the same magnitude. Same as in the case of process noise, an adaptive filter, further explored in chapter 3, can extract bias from the innovation, which is later taken into account in (1.11a). The innovation covariance matrix is defined as

$$S(k) = E[r(k)r(k)^T] = H(k)\tilde{P}(k)H(k)^T + R(k) \tag{1.8}$$

Where H is the Jacobian matrix given as

$$H(k) = \frac{\partial h}{\partial x}|\tilde{x}(k), u(k), \bar{n}(k) \tag{1.9}$$

Now, the filter gain K can be defined as

$$K(k) = \tilde{P}(k)H(k)^T S(k)^{-1} \tag{1.10}$$

Note that because of $S(k)^{-1}$, the innovation covariance matrix needs to be positive-definite. Given that the filter may track its state accurately, the $H(k)\tilde{P}(k)H(k)^T$ part from (1.8) might have elements on the main diagonal close to zero, leading $H(k)\tilde{P}(k)H(k)^T$ to be almost positive semi-definite. Thus, to avoid numerical instability during the inverse operation, R should be positive-definite.

With the filter gain defined, the *a posteriori* state estimate with its covariance matrix are defined as

$$\hat{x}(k) = \tilde{x}(k) + K(k)(r(k) - \bar{r}(k)) \tag{1.11a}$$

$$\hat{P}(k) = (\mathbb{1} - K(k)H(k))\tilde{P}(k) \tag{1.11b}$$

Note $\bar{r}(k)$ in (1.11a) which is innovation bias which is statistically equivalent to bias in observation noise and which can be estimated using an adaptive filter. $\mathbb{1}$ in (1.11b) is identity matrix. More information about how to derive Kalman Filter formulas can be found, for example, in [48].

1.6 Metrics for Simulations Analysis

In several places of this thesis, simulations are described to assess the performance of specific models. Unless specified otherwise, the following metrics are used for analysis:

- Root Mean Square Error (RMSE) of the position estimation. The smaller the error, the more precise the estimation. Note, however, that this metric provides only a single value and even though it may be small, there might be an occurrence of an error growing beyond the confidence margin and then reducing back to low values. In other words, this metric does not capture how well the filter tracks the error, only how precise it is when analysing the whole trajectory.

- The approximate area under the 3σ confidence margin of the position estimation. The smaller area, the more accurate the filter believes it is. Ideally, there should a narrow 3σ confidence margin with all errors contained by it (thus low RMSE). However, in practice, it is often seen that filters with narrow confidence margins do not track errors correctly, thus estimated errors exceed the boundary.

- The estimated area between the estimation error and 3σ confidence margin if the error exceeds it. This metric captures scenarios where the filter offers a narrow confidence margin, but does not track error correctly. As a result, estimated errors grow beyond the boundary. In the case of a filter that correctly tracks estimated errors, this value will be zero.

All areas are computed using the trapezoidal method. Note that because the Kalman filter assumes normally distributed errors, the 3σ boundary captures 99.7% of the results. Having an error plot within the 3σ confidence margin indicates that the filter correctly tracks errors.

1.7 Contributions, Structure, and Publications

1.7.1 Contributions

This thesis makes the following contributions:

- A novel approach is laid out in wheel slip estimation. The wheel slip is modelled as a low-pass filter driven by a process noise rather than an estimate based on WO and external measurement (VO in this work). The slip is indirectly measured using WO and VO, but the first-order dynamics allow forgetting the value to process noise mean. It is typically a zero, but it may be non-zero to reflect the environment better with an adaptive filter. This approach benefits from filtering erroneous VO measurements detected as spikes in the slip estimate.

- A system to reactively schedule VO measurements is defined. It does not depend on prior information about the environment. It mainly utilises statistical information of the process noise driving the wheel slip estimation and the innovation from the VO measurements (i.e., the difference between the predicted and measured VO). A reactive VO scheduling approach can substantially reduce the number of VO measurements while maintaining the required navigational. Other algorithms could utilise freed resources to advance the overall robotic application's capabilities further.
- Wheel slip prediction is vital from the planning perspective. It is best to avoid areas that may pose a risk to the locomotion system. To the authors' best knowledge, what has not been done so far is to use the predicted wheel slip to support the localisation function online or in a quasi-real-time manner. Typically, wheel slip predictions are used for rover planning off-board and on the ground, even if only based on heuristic operator judgement. This thesis explores options for improving position estimation by matching predicted wheel slip with known profiles. This behaviour resembles SLAM. It is an exciting observation as the wheel slip prediction could be of dual purpose.

1.7.2 Thesis Structure

Chapter 2 presents the proposed model for fusion of VO and WO with integrated slip estimation. The wheel slip is modelled as a low-pass filter driven by a process noise. It allows for tracking the correlation between the position and wheel slip, improving overall localisation. The low-pass filter also filters out erroneous VO estimates should they happen. The proposed model is analysed using simulated data and several real-world trajectories on two types of terrain: tarmac and grass. The model is also compared with two other popular sensor fusion approaches, where both treat wheel slip as pseudo-control input. The first model to compare is a simple fusion (i.e. without Kalman filtering), and the second encapsulates the simple fusion with the Kalman Filter framework.

Chapter 3 introduces an adaptive filter to the fusion model with integrated slip estimation. Two adaptive filter candidates are presented and evaluated, out of which only one is selected based on simulations and later verified using data captured during field trials. Next, the trigger strategies for the reactive VO scheduling are defined and discussed. They utilise the adaptive filter and are verified using short real-world trajectories. The final system is then validated using long trajectories from one field trial used for system tuning and from the second field trials conducted in a different environment. It is shown that the number of VO measurements can be significantly reduced without significant loss in position estimation accuracy.

Chapter 4 explores three models where information about wheel slip can be predicted, measured, and fused with VO and WO. The first model assumes a point measurement of wheel slip. It could be realised by a camera facing directly at wheels or utilising other measurements such as torque. The second model utilises a known wheel slip profile to perform a look-up operation. This profile could be obtained by analysing the terrain in advance, for example, by using images from an orbiter. The final model includes the profile within the sensor fusion framework. With this approach, wheel slip could be measured using, for example, forward-looking cameras. This model comes with two variants: with linear and non-linear observation functions. It is possible to recognise predicted wheel slip that resembles SLAM in both cases.

Chapter 5 provides an initial insight into moving the fusion of VO and WO with integrated slip estimation into two dimensions. Two additional slip variables are defined: the lateral slip, which behaves in the same way as the longitudinal slip (i.e. scales the lateral velocity), and the angular slip, which is treated as a bias for the vehicle's angular velocity. Similarly, all slips are modelled as low-pass filters driven by their respective process noises. For testing, two candidate models are defined. The first model aims to be platform-agnostic and utilises per-wheel input controls (drive and steering). The second model uses only drive and steering control inputs for the chassis and implements the non-linear Ackermann steering geometry in the WO observation function. An adaptive filter with reactive VO scheduling is implemented only for one model to inspect the feasibility of this approach in higher dimensions.

Finally, chapter 6 summarises the results, highlights key findings, and provides a discussion by answering the research question. All is concluded in chapter 7 which also provides suggestions for improvements and further research.

1.7.3 Publications

This thesis is based on the following publications:

1. M. Malinowski, A. Richards, and M. Woods, Power-aware Fusion of Visual and Wheel Odometry for Mobile Platforms, in UKRAS20 Conference: "Robots into the real world" Proceedings, EPSRC UK-RAS Network, May 2020, pp. 111–113. [49]

2. M. T. Malinowski, A. Richards, and M. Woods, Fusion of Visual and Wheel Odometry with Integrated Slip Estimation, in AIAA Scitech 2021 Forum, American Institute of Aeronautics and Astronautics, Jan. 2021. [50]

3. M. T. Malinowski, A. Richards, and M. Woods, Reactive Visual Odometry Scheduling Based on Noise Analysis using an Adaptive Extended Kalman Filter, in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sept. 2021, pp. 294–299. ISSN: 2153-0866. [51]

4. M. T. Malinowski, A. Richards, and M. Woods, Wheel Slip Prediction for Improved Rover Localization, in AIAA SCITECH 2022 Forum, American Institute of Aeronautics and Astronautics, Jan. 2022. [52]

CHAPTER 2

MODEL DEFINITION AND VALIDATION

This chapter lays the foundations of the work described in this thesis. Here, a new method is proposed to estimate the robotic rover's position corrected with slip. The solution integrates the slip estimation into an Extended Kalman Filter (EKF) fusing Wheel Odometry (WO) and Visual Odometry (VO). The approach can handle the correlation between the slip and the rover's position estimation and occasional errors in VO or WO measurements. Furthermore, it is possible to tune the model to emphasise WO (e.g. when no slip is expected and thus reduce the number of VO measurements) or to rely more on VO (high slippage variability). Accurate tracking of uncertainty offers a route to adaptive use of VO, saving energy and computational resources when conditions permit. In section 2.2 the proposed model is tested and verified using simulation, whereas, in section 2.3, it is validated using results obtained during field trials using a representative robotic platform. All tests are done in a simple one-dimensional case, which is extended to two dimensions in chapter 5. Results are promising as the position estimation is consistent even for various VO update periods. The model is also compared with other sensor fusion algorithms in section 2.3.2. Finally, an example of how a failure in VO measurement is dealt with by the proposed solution is provided in section 2.3.3.

This chapter is based on work published in [49, 50], but increased in scope. In case of any discrepancies, this thesis supersedes the papers.

2.1 Models Definition

In this section, three different ways of VO and WO fusion are discussed:

- Filter Free Estimation that performs a simple fusion of sensor data and treats wheel slip as a pseudo-control input

- Separate Slip Estimation which moves the above solution into a Kalman Filter but still treats the wheel slip as a pseudo-control input
- Integrated Slip Estimation where wheel slip is moved to the state and is driven by a process noise. In this scenario, a non-linearity arises, calling for an Extended Kalman Filter.

When it comes to sensors fusion, it is common to use an indirect filter to estimate state errors rather than states themselves, however, this approach is usually used with IMU, which is subject to bias error [22, 25]. Because most of the work described in this thesis is limited to one dimension, a direct EKF is employed.

Furthermore, the problem is simplified to one dimension, where only the total distance is analysed. Consequently, only the longitudinal wheel slip is taken into consideration. Finally, it is assumed that all wheels are always subject to the same type of terrain; therefore, a single slip value is estimated for all wheels, which is given as

$$\lambda(t) = \frac{\rho\Omega(t) - v(t)}{\max\{\rho\Omega(t), v(t)\}} \approx \frac{\rho\Omega(t) - v(t)}{\rho\Omega(t)} \tag{2.1a}$$

$$\lambda(t) \begin{cases} = 0 & \text{no slip} \\ > 0 & \text{wheels slip} \\ < 0 & \text{wheels skid} \end{cases} \tag{2.1b}$$

Where $\Omega(t)$ is the wheel angular velocity, ρ is wheel radius, and $v(t)$ is the linear velocity of the platform. Typically longitudinal slip can occur in both directions. When the $\rho\Omega(t)$ element is greater than $v(t)$, the platform is usually accelerating or moving at a constant speed. Linear velocity might be greater when the platform slides or during deceleration. Because this research deals with a reasonably slow-moving rover, we assumed that slip caused by sliding could be neglected, and therefore the longitudinal wheel slip formula can be simplified to what is presented on the right-hand side of (2.1a). Furthermore, having a max function in the state transition could result in linearisation errors when $\rho\Omega(t) \approx v(t)$.

Note that (2.1) can be rewritten to provide linear velocity $v(t)$ as

$$v(t) = \rho\Omega(t) - \rho\Omega(t)\lambda(t) = \rho\Omega(t)\big(1 - \lambda(t)\big) \tag{2.2}$$

This version is particularly interesting as it takes, as an input, $\lambda(t)$ (either as a state or pseud-control input, $u_\lambda(t)$, which will be explained later) and provides $v(t)$ as an output which can be then integrated to provide the total distance travelled. Eq. (2.2) can be depicted in a simple block diagram as presented in Fig. 2.1.

Some models treat wheel slip as a pseudo-control input. To distinguish it, it is denoted as $u_\lambda(t)$. The only difference in $u_\lambda(t)$ is that rather than being estimated by a filter, it is calculated using existing VO and WO measurements. $u_\lambda(t)$ is only calculated each time new VO is available and an unmodified pseudo-control value is used until the next VO measurement. In addition, WO

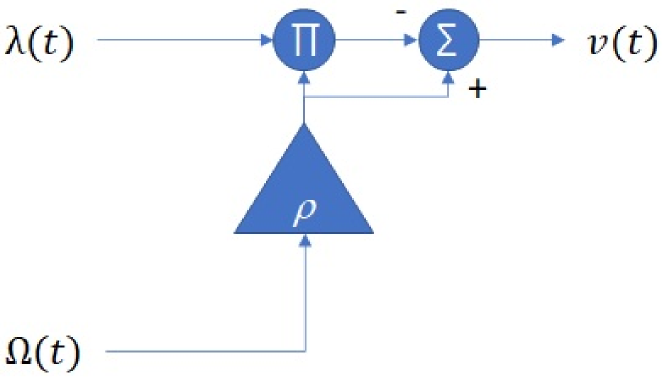


Figure 2.1: Block diagram of a linear velocity, $v(t)$ as a function of wheel slip, $\lambda(t)$, wheels angular rate, $\Omega(t)$, and their radius, ρ. It is a visual representation of (2.2). Note that when this diagram is included in the Kalman filter, the multiplication between $\rho\Omega(t)$ and $\lambda(t)$ introduces non-linearity which calls for an EKF.

measurements used to calculate $u_\lambda(t)$ are accumulated over the time between two consecutive VO measurements. Formal equations are provided when discussing specific models. Similarly to $u_\lambda(t)$, $u_\Omega(t)$ is defined to denote an input angular velocity of wheels. In this case, this is an actual input to the system known at each time step.

2.1.1 Filter Free Estimation

Kalman Filter is not mandatory for a localisation system. The benefits of filtered estimation may be overshadowed by additional computations needed to calculate Jacobian matrices for multi-variable state vector, primarily when VO produces precise measurements and is frequently computed. In the case of the ExoMars Rosalind Franklin rover, the platform's localisation is provided by a component called RelLoc which quoting [21] implements the following algorithm:

> RelLoc employs the angular rates from the GyroMei to propagate the attitude estimate and, together with the incremental positions from WheelOdo propagates the position estimate... It shall be noted that VisLoc estimates are considered truth within RelLoc. RelLoc therefore synchronizes the various inputs, adding the propagated GyroMei and WheelOdo data to the VisLoc samples.

Note that VisLoc (Visual Localisation for ExoMars rover) is a VO solution [33].

From the above description, it is not explicitly known whether any filtering is used. Nevertheless, for the sake of this work, it is assumed that there is none. A block diagram of a simple Filter Free Estimation is presented in Fig. 2.2. u_Ω represents an input wheel angular velocity which after integration gives the total accumulated angle θ that can be measured using WO. Angular velocity multiplied by wheel radius ρ is then subject to wheel slip which in this model is

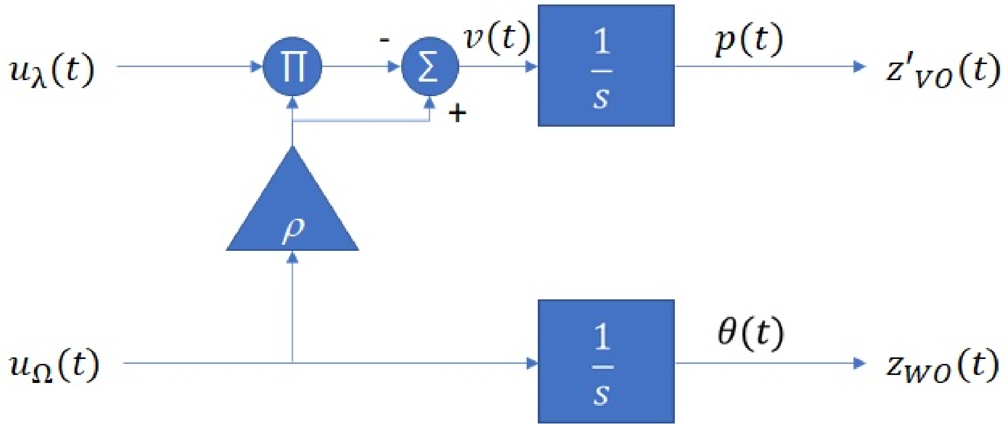


Figure 2.2: Block diagram of a Filter Free Estimation. Note how this block diagram expands the one presented in Fig. 2.1. The system takes as inputs commanded wheels angular rate, $u_\Omega(t)$, and wheel slip as pseudo-control value, $u_\lambda(t)$. The total accumulated wheel angle, $\theta(t)$ is obtained by integrating commanded input and is measured by WO, z_{WO}. Similarly, the total distance travelled, $p(t)$, is obtained by integrating linear velocity and is measured by accumulated VO measurements,$z'_{VO}(t)$.

inserted as pseudo-control input u_λ. The result is a linear velocity of the robot, $v(t)$, which after integration gives the total distance travelled p, which can be measured using VO. In this case, however, delta VO measurements are added together, providing a direct measurement of p given by z'_{VO}.

The pseudo-control wheel slip input is estimated by substituting WO and VO measurements into (2.1) and thus given as

$$u_\lambda(t) = \frac{\frac{\rho\left(z_{WO}(t)-z_{WO}(t-T_{VO})\right)}{T_{VO}} - \frac{z'_{VO}(t)-z'_{VO}(t-T_{VO})}{T_{VO}}}{\max\{\frac{\rho\left(z_{WO}(t)-z_{WO}(t-T_{VO})\right)}{T_{VO}}, \frac{z'_{VO}(t)-z'_{VO}(t-T_{VO})}{T_{VO}}\}} \tag{2.3}$$

Where T_{VO} is the time between VO measurements. Note that even though in (2.3) T_{VO} could be reduced, in a real-world application, it may not always be possible to synchronise VO and WO measurements correctly.

2.1.2 Separate Slip Estimation

In this approach, what was previously defined as Filter Free Estimation is placed inside a Kalman Filter where a control input u_Ω with added random walk (integration of process noise w_α) drives the system. Moreover, an augmented state m, which stores the position where the last VO measurement was performed, is introduced to better model relative VO measurements. This work was inspired by NASA as presented in [22] who also use augmented state to model relative measurements. Their approach is called stochastic cloning and is described in great detail in [23]. It is interesting that even though our proposition and stochastic cloning are similar

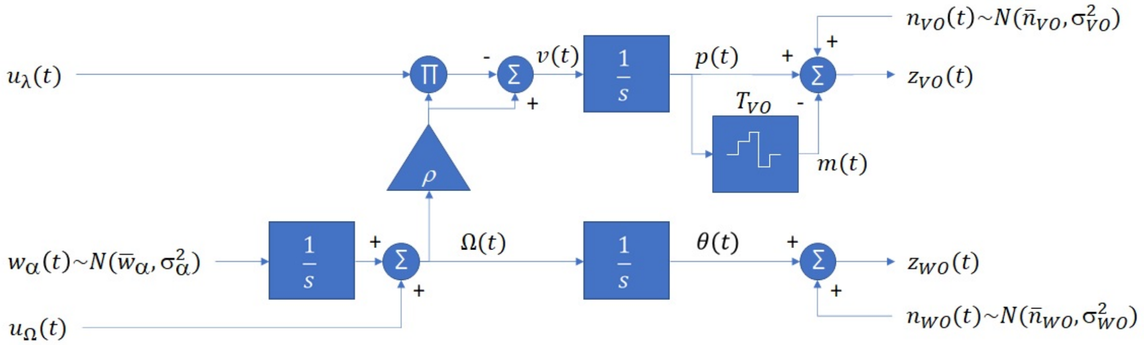


Figure 2.3: Block diagram of the Separate Slip Estimation dynamics system model used in the Kalman Filter (transition and observation functions). This model is similar to Filter Free Estimation presented in Fig. 2.2. There are two major changes. Firstly, as this model is for the Kalman filter, process noise and measurement noises are added. The process noise $w_\alpha(t)$ is integrated and added as a random walk to the input commanded angular wheel rate, $u_\Omega(t)$, resulting in wheels angular rate, $\Omega(t)$. Measurement noises $n_{VO}(t)$ and $n_{WO}(t)$ are added to VO and WO measurements respectively. Secondly, VO is modelled as relative measurements with T_{VO} rate, hence an augmented state $m(t)$ is introduced. $m(t)$ stores the position where the previous VO measurement was performed.

in principle, they differ in how they model associated covariances. Authors in [23] assume state independence so that new measurement does not provide any new information about the previous state. However, in this work, because a direct Kalman Filter is used to estimate states, we believe that each relative measurement provides new information about the previous state. It is best described in the following example: if a system indicates that the rover moved by X meters to a total distance of Y meters, the former state was at Y-X meters. It is new information, as we are confirming the previous belief. This notion leads to the following definition of the augmented state m:

$$m(t^+) = \begin{cases} p(t) & \text{if VO measurement at time } t \\ m(t) & \text{otherwise} \end{cases} \tag{2.4}$$

Where t^+ indicates a time instant after t.

A block diagram that is embedded in the Kalman Filter is presented in Fig. 2.3. Note that relative VO measurement, z_{VO}, and WO measurement, z_{WO}, have added measurement noises n_{VO} and n_{WO} respectively. Furthermore, note that all noises in Fig. 2.3 list mean values denoted by a bar above their symbol (e.g. $\bar{w}_\alpha$). Kalman Filter assumes that all noises are zero-mean Gaussian white noises. This is also assumed in work presented in this chapter, however, non-zero means are explored in chapter 3 where the adaptive filter is introduced.

Using relative VO motion, the slip pseudo-control variable can be simplified to

$$u_\lambda(t) = \frac{\frac{\rho\left(z_{WO}(t) - z_{WO}(t-T_{VO})\right)}{T_{VO}} - \frac{z_{VO}(t)}{T_{VO}}}{\max\{\frac{\rho\left(z_{WO}(t) - z_{WO}(t-T_{VO})\right)}{T_{VO}}, \frac{z_{VO}(t)}{T_{VO}}\}} \tag{2.5}$$

The state vector for Separate Slip Estimation is defined as

$$x(t) = [p(t), \theta(t), \Omega(t), m(t)]^T \tag{2.6}$$

This model is comparable in principle with [25] where a filter with the augmented state is used to estimate the rover's position, and the slip is estimated outside the filter.

2.1.3 Integrated Slip Estimation

The final model leverages the Separate Slip Estimation method by including wheel slip in the filter's state and modelling it as a low-pass filter driven by an unknown process noise. This approach means that the slip's covariance remains constant, similarly to white noise, rather than growing as for a random walk. Generally, the low-pass filter offers similar properties to unmodified white noise with an exception of providing non-zero slip estimation. Should the wheel slip be treated as a white noise, it would always be zero (mean value being zero as reacquired by Kalman filter) unless the noise mean value was modelled as an additional state in the model. When compared to a random walk, a low-pass filer offers smaller growth of the covariance. It is most prominent when exposed to a system where VO measurements are not available. This is further discussed in section 2.2.2. Additionally, the low-pass filter has a tunable 'forgetting' factor T_0 to capture the variability of terrain. Wheel slip depends on the terrain that is directly under the robot. As the robot moves, T_0 allows forgetting wheel slip value associated if the terrain that is behind the robot and allows Kalman filter to provide a new estimate. The gain parameter of the filter is set to one as it is a scaling factor for the w_β noise which here models all terrain properties related to the slip. Hence, slip estimation is controlled by σ_β and T_0. The Block diagram of the proposed system is presented in Fig. 2.4. Because λ is now part of the state, a non-linearity arises from (2.1), which calls for an EKF. Others have already investigated the suitability of an EKF

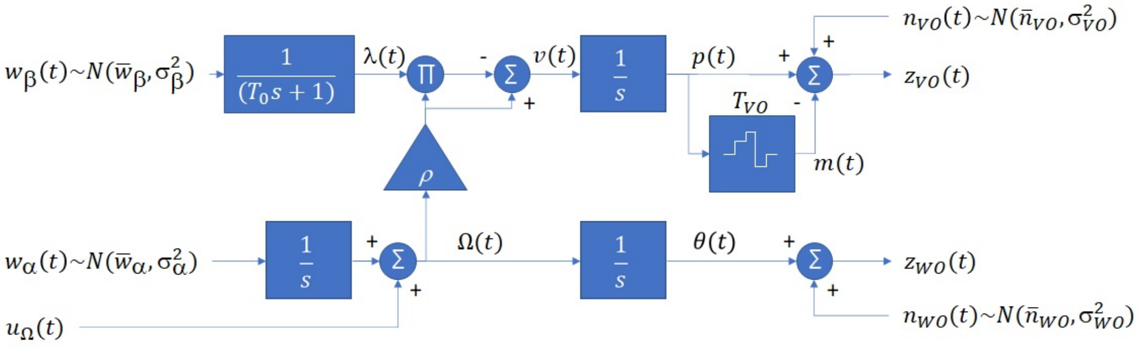


Figure 2.4: Block diagram of the Integrated Slip Estimation dynamics system model used in the EKF (transition and observation functions. This model expands the model with Separate Slip Estimation depicted in Fig. 2.3 by moving wheel slip, $\lambda(t)$, to filter's state and modelling it as a low-pass filter driven by a new process noise $w_\beta(t)$. The low-pass filter's gain is set to one as it only scales the noise. Its other parameter, time constant T_0 is configurable.

for sensor fusion and position estimation, such as [25, 53, 54] to name a few. Also, in this model, the assumption that the system moves at low velocity and is not subject to longitudinal slip during deceleration plays an important role. It allows to use an approximation from (2.1) which is present in the block diagram in Fig. 2.4.

Similarly to Separate Slip Estimation, all process and measurement noises are assumed to be zero-mean Gaussian white noise. Any potential biases are discussed in chapter 3 when analysing potential adaptive solutions. However, with this in mind, all equations are provided with explicit noises' means (note symbols with bars below). Even though those elements are zeroed in this chapter, equations remain valid for chapter 3.

Finally, the state of the Integrated Slip Estimation is defined as

$$x(t) = [p(t), \theta(t), \Omega(t), \lambda(t), m(t)]^T \tag{2.7}$$

Which is similar to (2.6) but extended with λ.

From Fig. 2.4 and using augmented state m as defined in (2.4), the state dynamics, f, discretised with timestep Δt is given as

$$f\big(\hat{x}(k), u(k), w(k)\big) = \begin{bmatrix} \hat{p}(k) + \rho\Delta t\big(1-\hat{\lambda}(k)\big)\left(\hat{\Omega}(k) + \frac{u_\Omega(k+1)-u_\Omega(k)}{2}\right) + \rho\frac{\Delta t^2}{2}\big(1-\hat{\lambda}(k)\big)\bar{w}_\alpha(k) \\ \hat{\theta}(k) + \Delta t\left(\hat{\Omega}(k) + \frac{u_\Omega(k+1)-u_\Omega(k)}{2}\right) + \frac{\Delta t^2}{2}\bar{w}_\alpha(k) \\ \hat{\Omega}(k) + u_\Omega(k+1) - u_\Omega(k) + \Delta t\bar{w}_\alpha(k) \\ e^{-\frac{\Delta t}{T_0}}\hat{\lambda}(k) + (1-e^{-\frac{\Delta t}{T_0}})\bar{w}_\beta(k) \\ \begin{cases} \hat{p}(k) & \text{if VO at } k \\ \hat{m}(k) & \text{otherwise} \end{cases} \end{bmatrix} \tag{2.8}$$

For the full derivation of the formulas refer to annex A. Note that element $e^{-\frac{\Delta t}{T_0}}$ comes from the Z-transform and could be approximated by $1-\frac{\Delta t}{T_0}$ for very small Δt. This approximation is employed in chapter 5.

The state transition Jacobian is defined as

$$F(k) = \begin{cases} \begin{bmatrix} 1 & 0 & \rho\Delta t\big(1-\hat{\lambda}(k)\big) & -\rho\Delta t\left(\hat{\Omega}(k) + \frac{u_\Omega(k+1)-u_\Omega(k)}{2}\right) - \rho\frac{\Delta t^2}{2}\bar{w}_\alpha(k) & 0 \\ 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & e^{-\frac{\Delta t}{T_0}} & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} & \text{if VO at time } k \\ \begin{bmatrix} 1 & 0 & \rho\Delta t\big(1-\hat{\lambda}(k)\big) & -\rho\Delta t\left(\hat{\Omega}(k) + \frac{u_\Omega(k+1)-u_\Omega(k)}{2}\right) - \rho\frac{\Delta t^2}{2}\bar{w}_\alpha(k) & 0 \\ 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & e^{-\frac{\Delta t}{T_0}} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \text{otherwise} \end{cases} \tag{2.9}$$

Note that in (2.8) and (2.9) $\bar{w}_\alpha(k)$ and $\bar{w}_\beta(k)$ refer to process noise mean value. Later they are assumed to be zero, however, in chapter 3 non-zero mean values are considered when discussing adaptation.

No correlation between w_α and w_β is assumed; therefore, the process noise covariance matrix with its associated Jacobian G mapping noise onto state are defined as

$$Q(k) = \begin{bmatrix} \sigma_\alpha^2 & 0 \\ 0 & \sigma_\beta^2 \end{bmatrix} \tag{2.10a}$$

$$G(k) = \begin{bmatrix} \rho\frac{\Delta t^2}{2}\left(1-\hat{\lambda}(k)\right) & 0 \\ \frac{\Delta t^2}{2} & 0 \\ \Delta t & 0 \\ 0 & 1-e^{-\frac{\Delta t}{T_0}} \\ 0 & 0 \end{bmatrix} \tag{2.10b}$$

VO and WO measurements are linear and are expressed, together with their measurement covariance matrices, as

$$z_{VO}(k) = \tilde{p}(k) - \tilde{m}(k) + n_{VO}(k) \tag{2.11a}$$

$$z_{WO}(k) = \tilde{\theta}(k) + n_{WO}(k) \tag{2.11b}$$

$$R_{VO}(k) = \sigma_{VO}^2 \tag{2.11c}$$

$$R_{WO}(k) = \sigma_{WO}^2 \tag{2.11d}$$

WO measurement noise is assumed to be constant and is related to the precision of wheel encoders, and its value can be obtained from the encoder's manufacturer datasheet. VO measurement noise is associated with image overlap: the bigger T_{VO} is, the bigger the noise is. It is because there is less image overlap for a longer distance between two consecutive stereo images. As the VO algorithm tries to match extracted features from two consecutive stereo pairs, a smaller overlap means fewer points are used for VO estimation, contributing to measurement error in the filter. It could be debated whether a smaller overlap introduces a bias, however, for simplicity, we assumed it to be zero. Standard deviations for various T_{VO} were estimated by comparing VO measurements to ground truth during the testing and calibration campaign.

Standard deviations for measurement noises used in this work were:

$$\sigma_{WO} = 0.009\ rad \tag{2.12a}$$

$$\sigma_{VO_{0.2}} = 0.004\ m \tag{2.12b}$$

$$\sigma_{VO_{1.0}} = 0.017\ m \tag{2.12c}$$

$$\sigma_{VO_{2.0}} = 0.033\ m \tag{2.12d}$$

$\hat{x}$ was initialised as

$$\hat{x}_0 = [0, 0, 1.7857, 0.02, 0]^T \tag{2.13}$$

$\hat{\Omega}_0$ was set to 1.7857 rad/s because this is 0.2 m/s (desired linear velocity of the platform used in all trials in this chapter) divided by wheel radius ρ of 0.112 m. Based on trials, the longitudinal wheel slip on the flat tarmac was estimated as 0.02; therefore, this value was selected as $\hat{\lambda}_0$ for all trials.

$\hat{P}_0$ was initialised as

$$\hat{P}_0 = \begin{bmatrix} R_{VO} & 0 & 0 & 0 & R_{VO} \\ 0 & R_{WO} & 0 & 0 & 0 \\ 0 & 0 & R_{WO} & 0 & 0 \\ 0 & 0 & 0 & 9E-04 & 0 \\ R_{VO} & 0 & 0 & 0 & R_{VO} \end{bmatrix} \tag{2.14}$$

Measurement covariance matrices were set as initial variances for states mainly measured using the given method. Ω and θ are both related to WO, therefore their initial variances are set to R_{WO}. Total distance p is mainly measured using VO, hence R_{VO} is used as p's initial variance. Augmented state m is only used for VO thus it is initialised with the same value. Because p and m are correlated by VO measurements, the initial correlation is also set in $\hat{P}_0$. Only initial variance for λ was selected based on trials and errors.

2.2 Basic Simulation

The first step was to perform low-fidelity simulations to verify the mathematical model in isolation from the environment and hardware. Secondly, simulations allowed for better inspection and understanding of the model without 'polluting' it with the randomness of the real world.

WO measurements were simulated every 0.1 s to be representative with telemetry feedback of the platform used for real-world experiments. The simulated ground truth was created using a constant control velocity (0.2 m/s) with pre-configured λ for different longitudinal wheel slip. Thus obtained total accumulated wheel angle and distance travelled were then used as a source of measurements with added white noise. Measurements were fused using only the model with Integrated Slip Estimation defined in section 2.1.3, as it is the most complex of all three models defined, and it is the solution that we propose as an improvement to the rover's localisation.

For testing and verification, a 100-m long test case was defined with wheel slip taking the shape of two bumps, as shown in Fig. 2.5. When simulating the model, three parameters are investigated:

- The standard deviation σ_β which drives the wheel slip estimation. The expectation is that with low covariance values, there will be not enough dynamic allowed to change λ, maintaining its default value (mean value of the noise). The high standard deviation of the process noise should result in noisy λ estimation, and thus, travelled distance p.

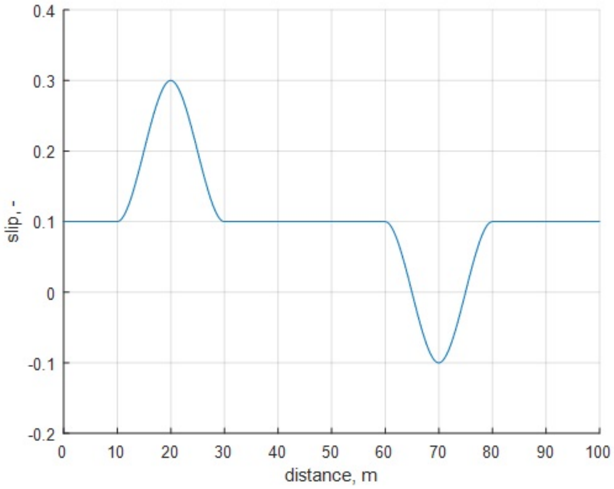


Figure 2.5: Ground-truth wheel slip profile for the simulated test case. The profile has a default non-zero value with two 'bumps' in opposite directions to test how robust models are when exposed to positive and negative changes to slip.

- The time constant of the low-pass filter T_0. It defines how fast λ is reached and how fast it is 'forgotten' to the mean value of the noise. A long time constant may increase error as λ may not be updated fast enough to reflect the environment. On the other hand, a short time constant may not maintain the correct slip estimation long enough between VO measurements.
- VO measurement period T_{VO}. Frequent VO offers a reasonable means of localisation, however, given the limited computational power on robots and the number of complex tasks they are being asked to perform, reducing the need for VO measurements can be beneficial. The drawback is that VO estimation error increases as the separation between consecutive image frames increases. To model that, σ_{VO} is increased linearly with increased T_{VO}. Furthermore, a longer time between VO updates may hurt the system with low T_0, as mentioned earlier.

2.2.1 Simulation Results

This section presents the results of the simulations aiming to assess the model's behaviour. For a given set of σ_β (0.1, 0.5, 1.0, and 2.0) and T_0 (1, 20, and 100 s), the trajectory was re-run with T_{VO} 0.2, 1.0, and 2.0 s. Fig. 2.6 shows an example trajectory and Fig. 2.7 presents the metrics for errors in position estimation as defined in section 1.6. The following conclusions could be drawn regarding the model with Integrated Slip Estimation:

- Low σ_β leads to underestimation of λ. WO drives the system with little information about the wheel slip. In (2.8), λ is defined as an interpolation between the previous value and

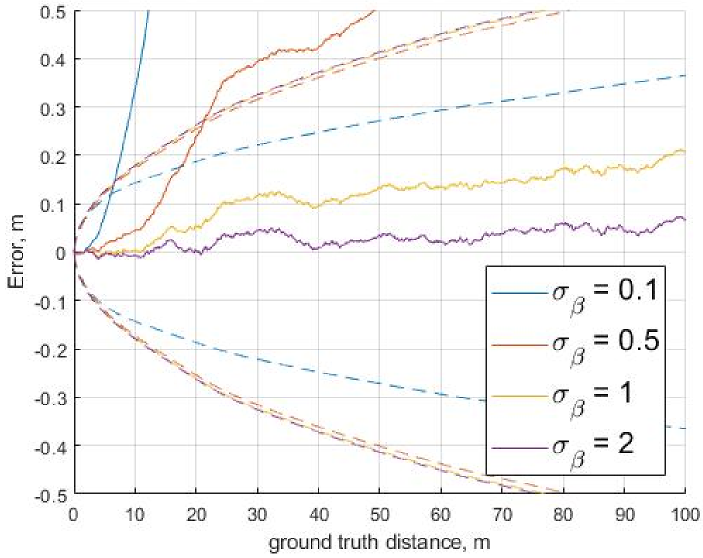
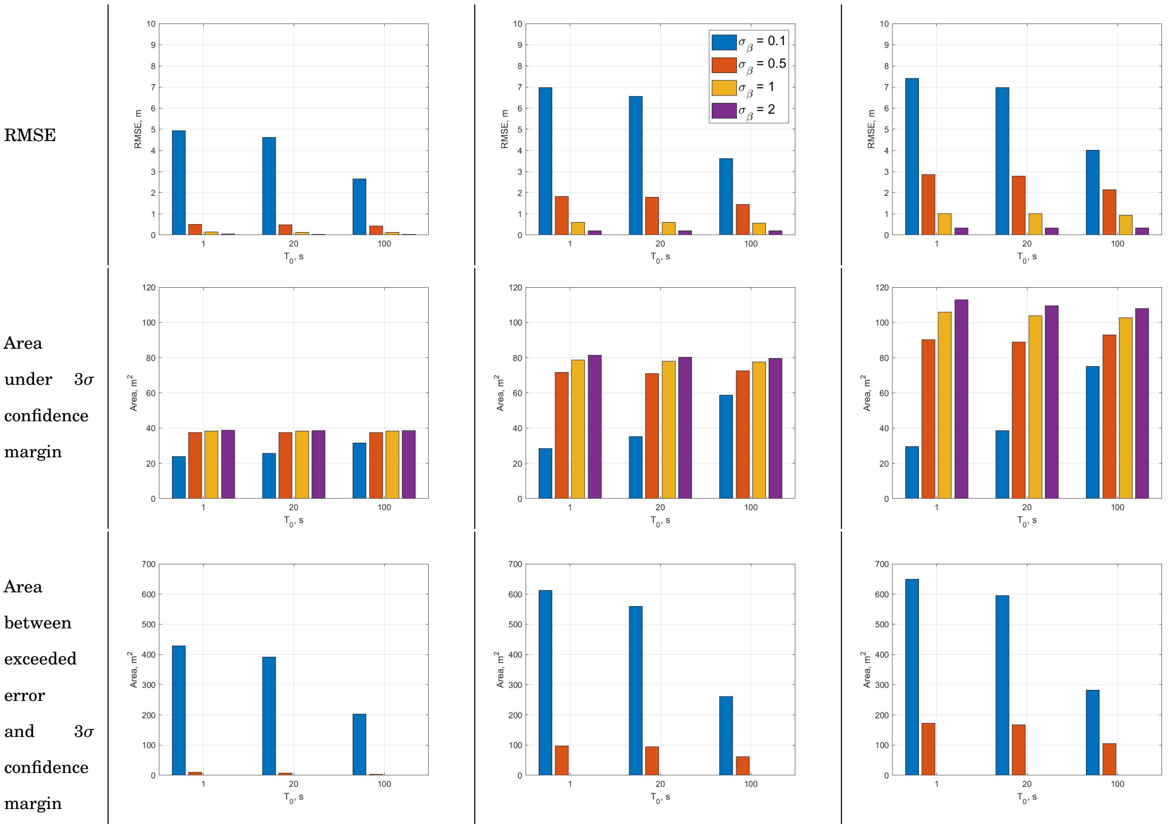


Figure 2.6: An example of simulated trajectory with T_{VO} = 0.2 s and T_0 = 20 s. Each colour represents different σ_β (note the legend). Small σ_β indicates that the terrain is fairly stable and thus the model can trust WO more. It means that not enough information is provided to estimate wheel slip λ which, being modelled as a low-pass filter, gradually becomes zero (mean value of w_β process noise. It is wrong as the ground truth wheel slip is non-zero, as seen in Fig. 2.5. On the other hand, bigger σ_β allows fusing more VO measurements into wheel slip estimation resulting in a small error.

the mean of the process noise w_β which, in the case of EKF, is zero. It leads to wheel slip being 'forgotten' to a wrong value (note that in the test case seen in Fig. 2.5 the slip is always non-zero). Increasing the time constant T_0 improves the performance a little, as the slip has more time to reset back to zero. Also, with small σ_β, the variance in position estimation is not correctly tracked, leading to errors in position estimation growing beyond the confidence margin.

- Increasing σ_β reduces RMSE considerably but at the expense of increased position estimation confidence margin. As more information from VO is fused, so is the variance in position estimation increased to accommodate possible offsets coming from relative measurements. It can also be seen that this increase in the variance is related to T_{VO}, the longer time between VO measurements, and thus the bigger measurement covariance R_{VO}, the bigger is position estimation variance.

- Increasing T_0 has only a noticeable impact at low σ_β, but the former offers minor improvement as the latter increases. Time constant would be more meaningful when a slip was rapidly changing, for example, in a significant step-change. However, such conditions are not real-world representative.



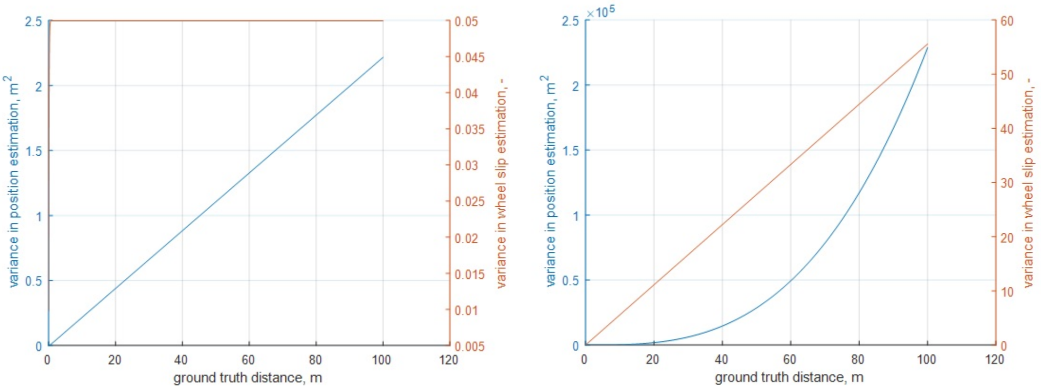
(a) T_{VO} = 0.2 s (b) T_{VO} = 1.0 s (c) T_{VO} = 2.0 s

Figure 2.7: Summary of metrics for errors in position estimation for the model with Integrated Slip Estimation. The time constant T_0 is labelled on each plot separately (1 s, 20 s, and 100 s). Note small RMSE and narrow 3σ confidence margins for frequent VO measurements. Also, the bigger σ_β, the smaller RMSE at the cost of a slightly bigger confidence margin. For very small σ_β, errors were exceeding the confidence margin (bottom plots).

2.2.2 Model Properties

There are a few notable properties of the model which are worth mentioning. The first is that the low-pass filter allows to bound the variance in position estimation when no VO measurements are available, and thus wheel slip remains unobservable. It is possible that in the environment where wheel slip is known and does not change, cameras are turned off to minimise computational resources usage. Example variances in position and wheel slip estimations for a model that uses a low-pass filter to drive wheel slip estimation (Integrated Slip Estimation) and an equivalent using only a random walk (i.e., when $\lambda(k+1) = \lambda(k) + \Delta t w_\beta(k)$) is presented in Fig. 2.8. Note the scale and exponential nature of the variance in position estimation Fig. 2.8(b) as opposed to Fig. 2.8(a).

The second exciting property can be spotted in augmented state m. In section 2.1.2 it was mentioned that we believe that each new VO measurement should provide information about



(a) Wheel slip modelled as a low-pass filter (b) Wheel slip modelled as a random walk

Figure 2.8: Variances in position estimation (blue) and wheel slip estimation (orange) for the model with Integrated Slip Estimation where wheel slip is modelled as a low-pass filter (a) and as a random walk (b) without any VO measurements. Parameters used: $\sigma_\beta = 1$, $T_0 = 1$ s. Note the scale on both Y axes. Variance in wheel slip estimation reaches only 0.05 for a low-pass filter compared to over 50 for a random walk. Should a white noise be used to model wheel slip, its variance would be equal to σ_β^2 (1 in this case). Linear growth in wheel slip estimation variance for random walk translates to exponential growth in variance in position estimation.

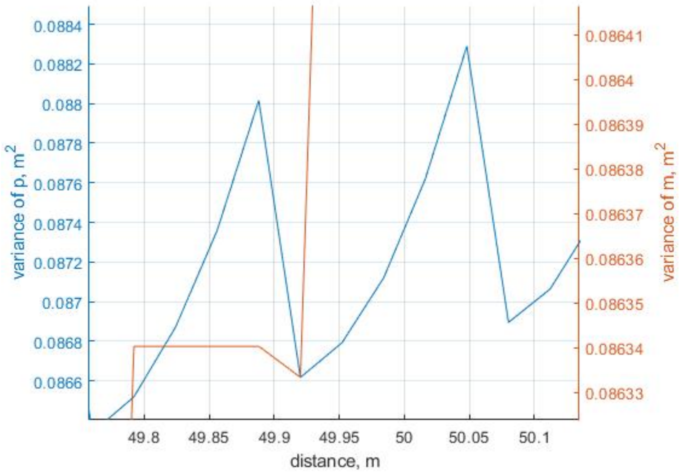


Figure 2.9: Zoomed in variances in p (blue) and m (orange) estimations (note difference scale). Notice how both drop when a new VO measurement is available. It shows that each VO contributes provides new information not only to the current position, p, but also to the position where the previous VO was performed, m.

the position where the previous VO was performed, i.e. m. Fig. 2.9 shows example variances for p (blue) and m (orange) (note different scales on Y axes). It is possible to see that with new VO measurement not only the variance in position estimation is decreased, but so is the variance in the augmented state m. It shows that each VO measurement also provides new information about the position where the previous measurement was performed. Because VO compares delta motion between current image pair and previous; therefore, if it detects that rover moved, for example, 20 cm, it may well update the past position having all the WO measurements in between to support that decision.

2.3 Experimental Results

2.3.1 Field Trials

2.3.1.1 System Setup

For experimental results, CGI's[1] custom-built rover was used, as presented in Fig. 2.10. The rover was equipped with two hardware synchronised Grasshopper3 cameras for VO measurements, eight Dynamixel MX-106R servos (four steering and four drive) with AS5045 rotatory position sensors for WO measurements, and Real-Time Kinematic (RTK) Global Navigation Satellite System (GNSS) for ground truth (OxTS xNAV 550 with dual antenna and GenPro 325e modem

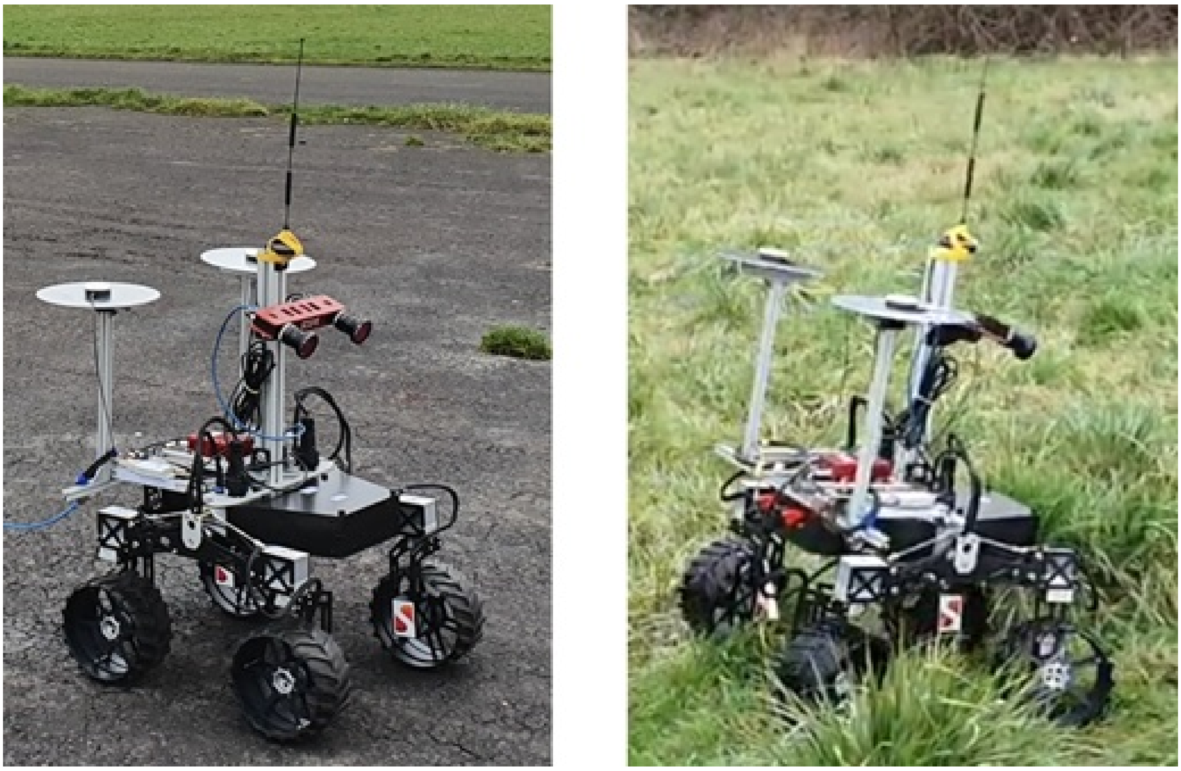


Figure 2.10: Mobile platform used during the experiments shown driving on the tarmac and grass. A stereo camera (red on the mast) was used for VO measurements. Each wheel had a servo mounted in their hub that provided telemetry used for WO.

---

[1] CGI is an IT and business consulting services firm that operates across the globe. In 2019 CGI acquired SCISYS together with their robotics team whose hardware and software were used in this thesis.

using Leica SmartNet). The target rover speed was 0.2 m/s, and WO measurements were done at around 10 Hz. VO provided six degrees of freedom measurements that were transformed to rover frame, whose origin is in the geometrical centre of the rover projected at the ground level. Ground truth was transformed into the same frame for comparison. Raw VO imagery was captured at 5 Hz, enabling post-processing using different T_{VO}. The rover was manually driven using CGI's proprietary software.

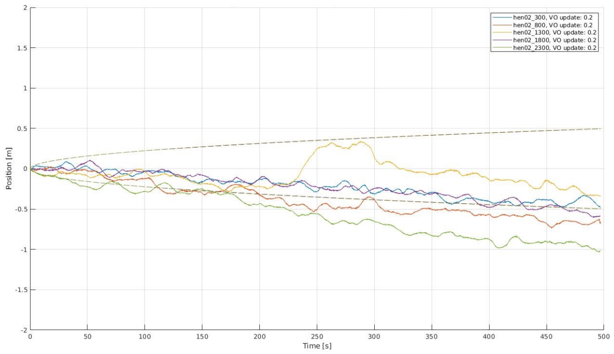
Because the VO algorithm that was used in this project (a variation of [33]) did not provide associated covariance, VO measurement noise was estimated during several test runs in CGI's car park. The rover was operated on flat terrain, and imagery for VO was captured. VO was re-run multiple times during the post-processing step with different update frequencies. All results were compared against the ground truth to calculate the measurement error. Thus obtained standard deviation values were used to model measurement noise for various T_{VO}. This 'calibration' trajectory was not used to analyse the proposed model's performance.

2.3.1.2 Calibration

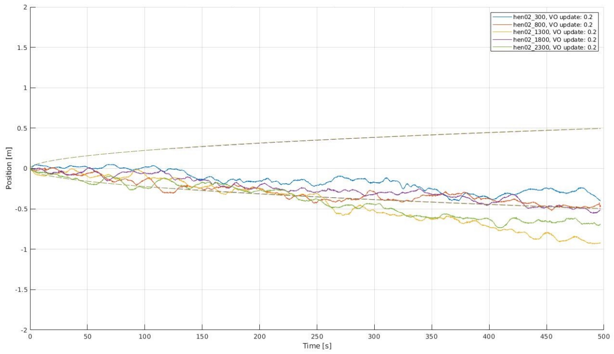
Correct sensor calibration is key to any robotics system. Small offsets in sensors mounting may lead to significant estimation errors. Also, individual sensors calibration is essential, as a miscalibrated camera may lead to wrong focal length or principle point measurements that directly impact VO performance. A good example is presented in Fig. 2.11 using data from the Hengrove data set, which is explained in section 2.3.1.3. The top Fig. 2.11(a) illustrates the initial results with one trajectory reporting an unexpected jump in the position estimation error (yellow plot). After further investigation, this trajectory was identified as wide Ackermann's platform turn to continue driving back towards the original start point. Such error during the turn is often associated with poorly aligned or calibrated mounting points of sensors. After a more detailed inspection, RTK's GPS antenna mounting was identified as wrongly specified by order of magnitude: 2 cm instead of 20 cm, which could have been a user error. The middle Fig. 2.11(b) illustrates position estimation error with updated ground truth, which smoothes the jump. However, knowing that the used VO algorithm tends to be quite reliable, further investigation was done towards the camera itself to understand why errors were growing outside the filter's 3σ confidence margin (dashed lines). After a detailed examination of the images themselves, it occurred that the stereo camera was miscalibrated. Using a checkerboard to recalibrate it and re-running VO lead to better position estimates as presented in the bottom Fig. 2.11(c).

2.3.1.3 Dataset Overview

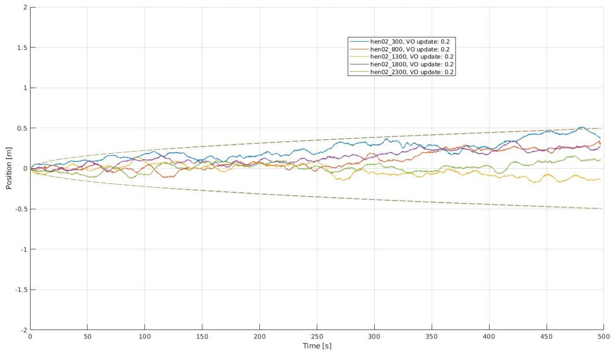
Field trials were planned in the area with varying terrain to validate the proposed model better. Hengrove Park in Bristol was identified as a candidate site (former airfield). It offers a long flat tarmac surface (runway) with large fine grass areas and meadow-like terrain. The path overview



(a) Default GPS antenna and default stereo camera calibration



(b) New GPS antenna and default stereo camera calibration



(c) New GPS antenna and new stereo camera calibration

Figure 2.11: Example of calibration impact on error in position estimation. Top graph: initial results. Central graph: RTK antenna position was recalibrated. Bottom graph: in addition to the previous, the stereo camera was recalibrated.

is presented in Fig. 2.12 where green lines indicate full RTK coverage and red sections show lost RTK coverage where only Differential GPS localisation was available. The path between points A and B was used for RTK GNSS warm-up and initialisation. The path between points B and C is purely on the tarmac. The path between points C and D is on the grass, where the first stage was on a fine-cut pitch. Later the rover climbed a small hill, drove back down, and finally drove across long grass terrain, which can be seen in Fig. 2.10 on the right. Although the model under investigation is only 1D, the rover was moving in 2D, estimating the total distance travelled as per Fig. 2.12.

2.3.2 Tuning and Validation

In this section, the impact of σ_β, T_0, and T_{VO} on the proposed model's performance are analysed and discussed. The aim is to verify the model's consistency in position estimation. To achieve this, first, the entire trajectory on the tarmac (between points B and C in Fig. 2.12) is inspected, which is approximately 500 meters long. Tarmac has very little slip; therefore, model performance can be assessed in isolation. Next, both trajectories on tarmac and grass (between points B and D in Fig. 2.12) were divided into 500 seconds-long sections without any overlap. Because the rover drove at 0.2 m/s, it gave 100 metre-long trajectories.

2.3.2.1 Tuning Process Noise Which Drives Wheel Slip

First to analyse is the impact of σ_β on the Integrated Slip Estimation model. The expectation is that the model's behaviour on changing noise statistics would follow the pattern presented in section 2.2.1 when simulating the model. As a starting point, T_{VO} is set to 2 s. Faster VO

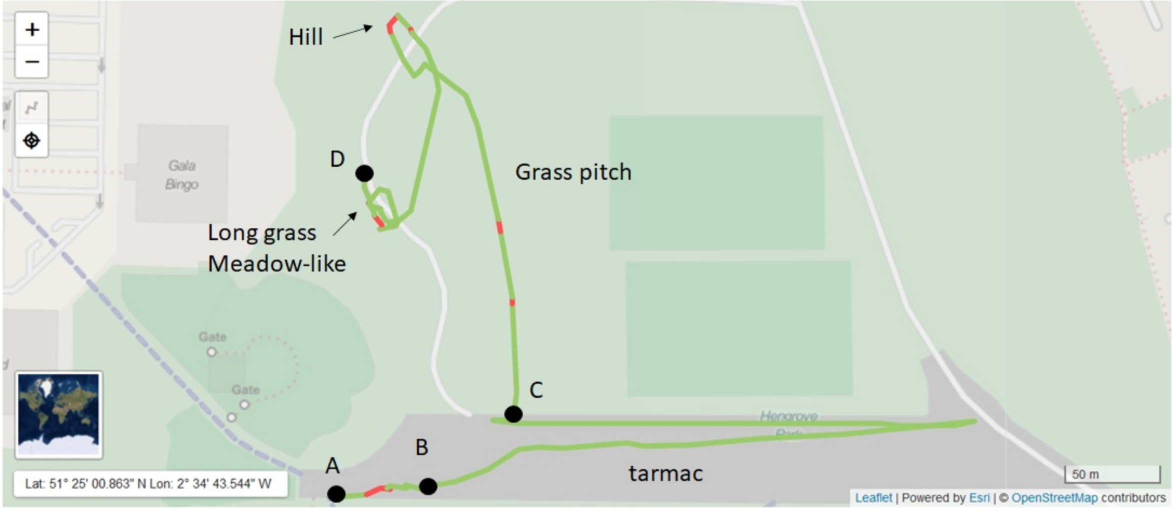
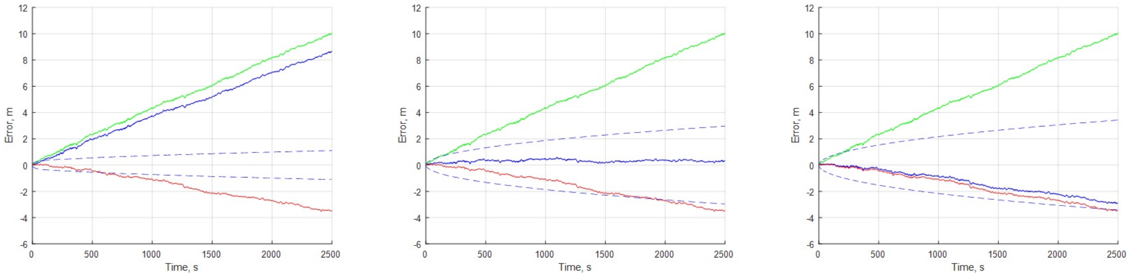
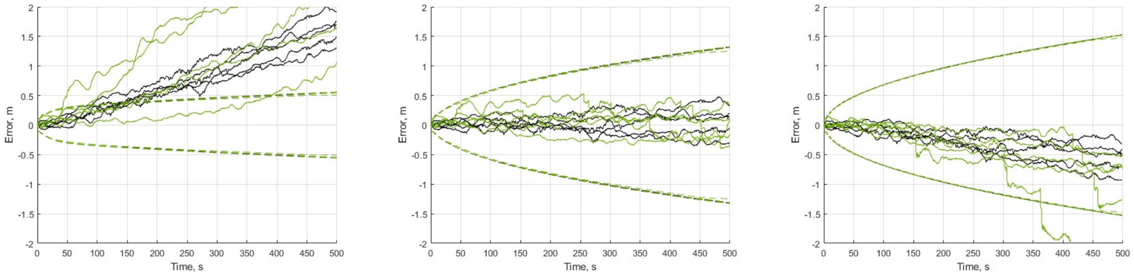


Figure 2.12: Recorded ground-truth path in Hengrove Park from SmartNet. Green lines indicate full RTK coverage, whereas red only dGPS.



(a) σ_β = 0.1 (b) σ_β = 0.5 (c) σ_β = 1.5

Figure 2.13: Impact of different σ_β on the proposed model's position estimation and confidence margin. Solid blue line - model, solid red - unfiltered VO, solid green - unfiltered WO, dashed blue - model's 3σ confidence margin.



(a) σ_β = 0.1 (b) σ_β = 0.5 (c) σ_β = 1.5

Figure 2.14: Integrated Slip Estimation's position error across multiple trajectories with different σ_β. Black lines indicate trajectories on tarmac and green on grass. Dashed lines represent 3σ confidence margin.

measurements would dominate the position estimation over WO with estimated slip. On the other hand, a long time between VO updates may result in some errors introduced into the model. At this stage, the aim is to inspect the model's behaviour; thus, edge cases should be minimised. T_0 is set to 20 s based on results from simulations.

Results presented in Fig. 2.13 with real-world data confirm the findings from simulations. Plot 2.13(a) shows how the Integrated Slip Estimation model (solid blue) closely follows unfiltered WO measurements (solid green) for small σ_β, whereas on graph 2.13(c) model's error in position estimation is almost exact as for unfiltered VO (solid red). Note how EKF's confidence margin (dashed blue) changes its shape based on information, including measurement noise, from sensors it relies on.

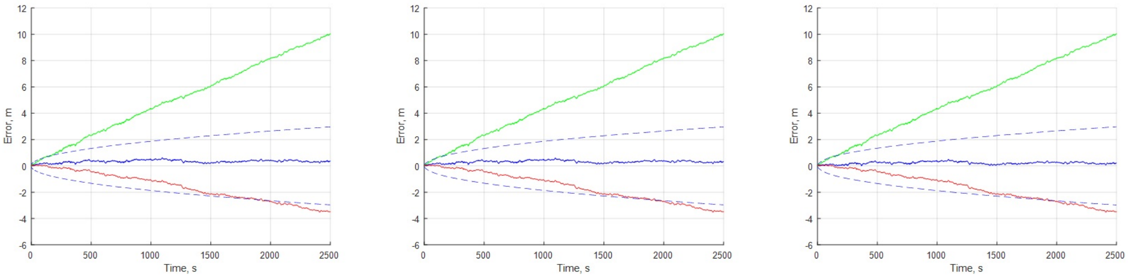
Next, the same analysis is performed on multiple 500-second-long trajectories on tarmac and grass. Similarly to single trajectory analysis, plots for low σ_β in Fig. 2.14(a) follow WO measurements, which overestimated the distance (positive error). On the other hand, plots with bigger σ_β follow VO measurement more closely, which, in this case, may slightly underestimate (negative error). Because the filter trusts VO measurements more, any failures in VO estimation impact the system performance severely. It is seen as two green lots in 2.14(c). This problem is further discussed in section 2.3.4.

Based on Fig. 2.13(b) and Fig. 2.14(b), it can be seen that the best performance is achieved for σ_β equal to 0.5. From now on, this value will be used when assessing other parameters and for models' comparison in section 2.3.3. It is also worth noting that the exact value is not necessarily required, as tuning process noise statistics is a job for an adaptive filter that is discussed in chapter 3.

2.3.2.2 Tuning Time Constant

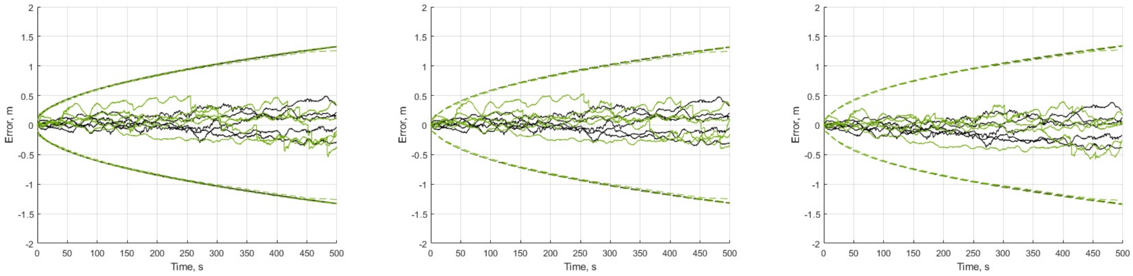
As presented in Fig. 2.4, a model with Integrated Slip Estimation has a tunable time constant T_0 which smoothens slip estimation. Also, it allows tuning how fast past slip information should be "forgotten". Properties of T_0 are best illustrated with a spike in the slip estimation (such as simulated Test Case 2 with a step-change in slip). Because when using real-world data it is impossible to know the ground truth slip (even with RTK, it requires to be combined with WO), the same approach is chosen as before. First, the entire path on the tarmac is analysed for three different T_0 values. Next, the same parameters are assessed on both types of terrain using 500-second-long trajectories. Based on the outputs from the previous section, σ_β is set to 0.5, and T_{VO} remains set to 2 s.

From Fig. 2.15 and 2.16 it cannot be conclusively said which T_0 value offers better performance. It seems there is not much difference between 1 s and 20 s, whereas a model with the time constant set to 100 s improves visibly only one grass trajectory. However, the improvement is not substantial. Because of that and knowing how the slip estimation behaved during simulations, we set T_0 to 20 s.



(a) T_0 = 1 s (b) T_0 = 20 s (c) T_0 = 100 s

Figure 2.15: Impact of different T_0 on the proposed model's position estimation and confidence margin. Solid blue line - model, solid red - unfiltered VO, solid green - unfiltered WO, dashed blue - model's 3σ confidence margin.



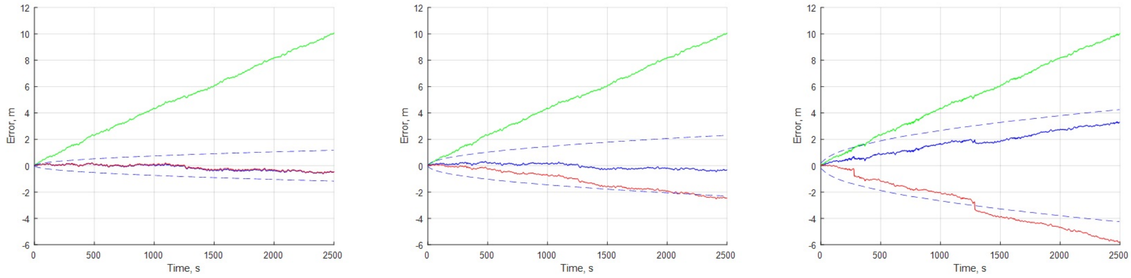
(a) T_0 = 1 s (b) T_0 = 20 s (c) T_0 = 100 s

Figure 2.16: Integrated Slip Estimation's position error across multiple trajectories with different T_0. Black lines indicate trajectories on the tarmac and green on grass. Dashed lines represent 3σ confidence margin.

2.3.2.3 Effect of VO Frequency on Position Estimation

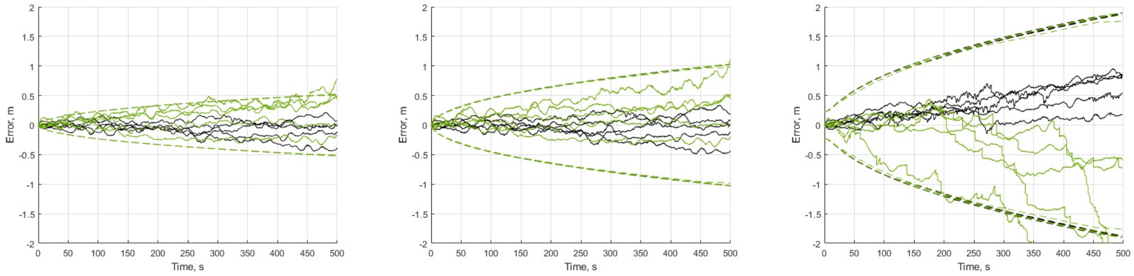
The period between VO updates is not a configurable parameter, but knowing how the model behaves for various T_{VO} can improve the overall understanding of the system's performance. The expectation is that the EKF will perform better with more frequent VO measurements and decrease its precision with longer T_{VO}, as VO measurement noise should contribute more to the final position estimation. In contrary to simulation results where VO's measurement covariance was scaled linearly with T_{VO}, here measurement noise was calibrated using dedicated trials in a car park for each T_{VO} under investigation. Based on previous sections, σ_β was set to 0.5 and T_0 to 20 s.

Results for a single long trajectory and multiple short are presented in Fig. 2.17 and Fig. 2.18 respectively. Note how the model follows almost exactly the unfiltered VO estimation in Fig. 2.17(a). As expected, the longer T_{VO}, the more information from WO is fused. Fig. 2.18(c) gives a good indication that 3 seconds is the longest time between VO updates for a rover driving



(a) T_{VO} = 0.2 s (b) T_{VO} = 1.0 s (c) T_{VO} = 3.0 s

Figure 2.17: Impact of different T_{VO} on the proposed model's position estimation and confidence margin. Solid blue line - model, solid red - unfiltered VO, solid green - unfiltered WO, dashed blue - model's 3σ confidence margin.



(a) T_{VO} = 0.2 s (b) T_{VO} = 1.0 s (c) T_{VO} = 3.0 s

Figure 2.18: Integrated Slip Estimation's position error across multiple trajectories with different T_{VO}. Black lines indicate trajectories on the tarmac and green on grass. Dashed lines represent 3σ confidence margin.

at 0.2 m/s. Trajectories on tarmac remain fairly accurate, and within 3σ margin, however, some trajectories on grass start to fail.

In some plots from Fig. 2.18(c) it is possible to notice a few sharp changes in the errors. These are believed to be erroneous VO measurements fused into the model. This problem will be further discussed in section 2.3.4.

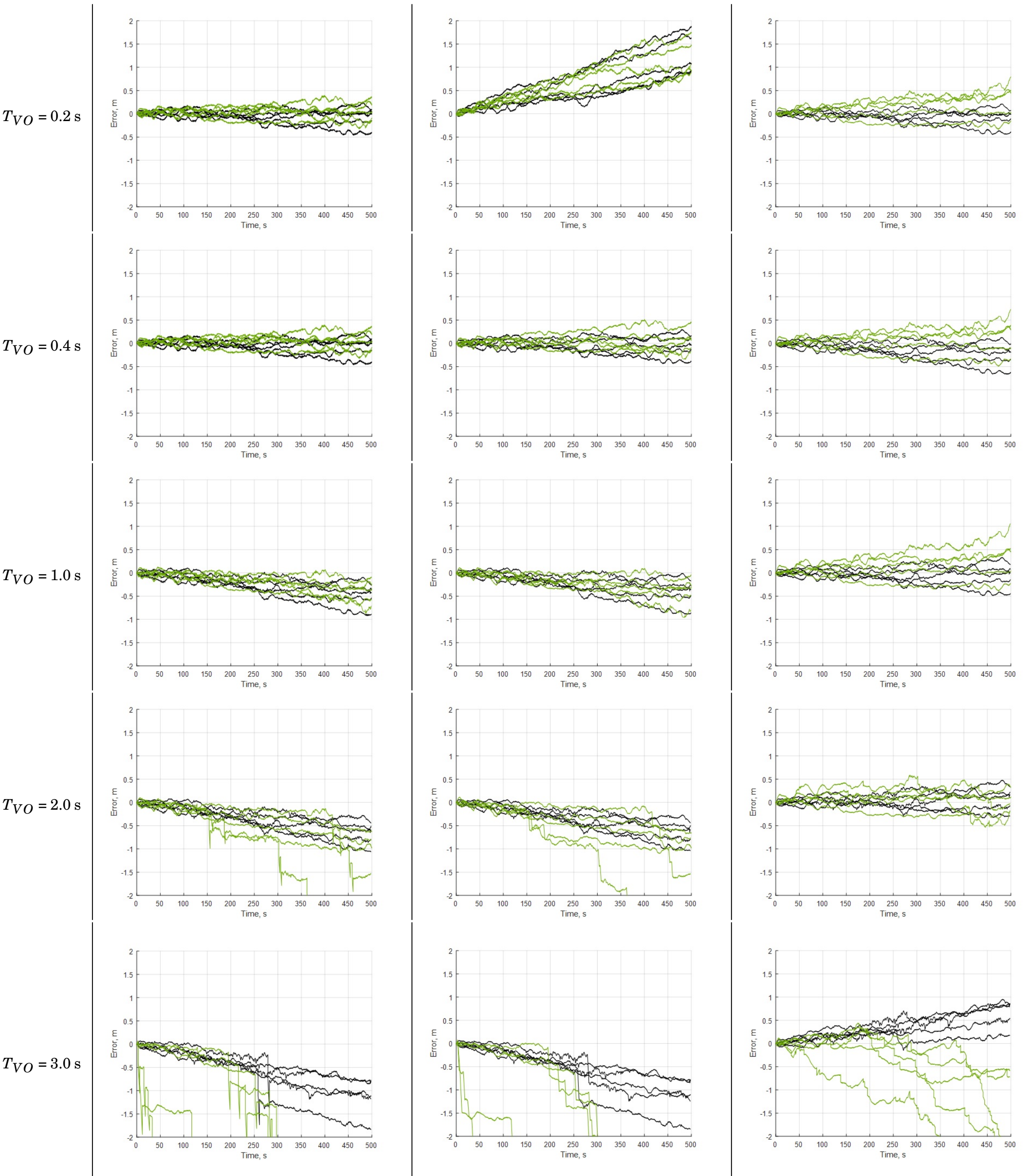
2.3.3 Models Comparison

The model with Integrated Slip Estimation has been tested and verified using low-fidelity simulations and then validated using real-world data from field trials. The important question is how its performance compares with other similar solutions? In this section, a comparison between models defined in section 2.1 is provided. Where applicable, all parameters are the same with σ_β and T_0 for Integrated Slip Estimation being set to 0.5 and 20 s, respectively. For all three models, T_{VO} will change to see how they compare at various VO update periods. Finally, because there is no estimation about the confidence margin in Filter Free Estimation, only errors in position estimation are provided for clarity.

There are several remarks based on results presented in Fig. 2.19.

- The Filter Free Estimation in Fig. 2.19(a) indeed treats VO updates as truth and closely follows its updates. Recall from previous plots that WO was always overestimating due to slip, but VO, at larger T_{VO} was underestimating. It is clearly seen for Filter Free Estimation. Furthermore, there are erroneous spikes for some grass trajectories when T_{VO} is set to 2 s and 3 s. These are correctly filtered out when using a basic Separate Slip Estimation in Fig. 2.19(b).

- Separate Slip Estimation with frequent VO is remarkably bad. If the VO frequency is slightly slower, the results are similar to Filter Free Estimation. It is understood that the problem comes from noisy wheel telemetry. In the article, [50] the time base was incorrectly



(a) Filter Free Estimation (b) Separate Slip Estimation (c) Integrated Slip Estimation

Figure 2.19: A comparison of errors in position estimation between different fusion models: Filter Free Estimation on the left, Separate Slip Estimation in the middle, and Integrated Slip Estimation on the right. Note T_{VO} on the left which is applicable to all plots in the row.

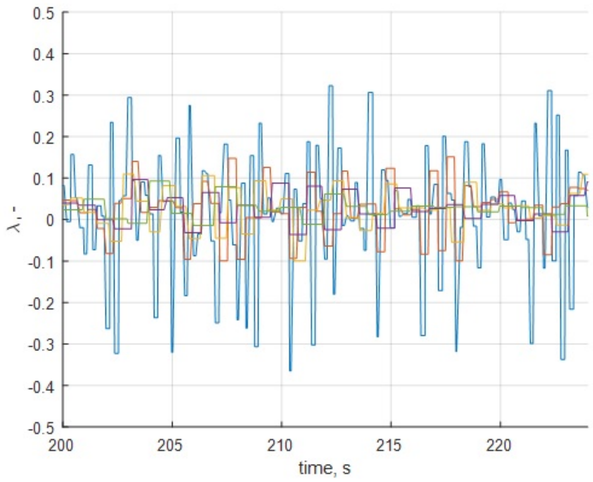


Figure 2.20: Pseudo-control wheel slip for various T_{VO}. The blue plot is for 0.2 s, and other colours are for 0.4 s, 0.6 s, 0.8 s, and 1.0 s. Note how they are comparable in contrast to the blue plot.

increased, leading to smoother slip estimation and thus position estimation. It is best depicted in Fig. 2.20, where slip estimation for frequent VO measurements is very noisy. It happens because the WO is noisy, and there are not enough WO measurements between VO to smoothen the noise. The problem does not exist if T_{VO} is increased to 0.4 s. Filter Free Estimation has the same pseudo-control slip, but the estimated position is always set to what VO reports (VO measurements are considered truth). Furthermore, Separate Slip Estimation uses delta VO measurements using augmented state. When pseudo-control slip is incorrect, errors are propagated, likely, because the filter also adjusts the augmented state m. It is in line with model properties listed in section 2.2.2 and refers specifically to Fig. 2.9, where each VO measurement adds new information about the location where the previous VO measurement was performed (augmented state). In this case, it works to Separate Slip Estimation disadvantages.

- Integrated Slip Estimation has a slightly bigger error spread at the end of trajectories. However, as opposed to other solutions for which error decreases as T_{VO} increases, there is no noticeable gradient pattern in the errors for the model with Integrated Slip Estimation. Because of that, the model proves to be repeatable regardless of VO frequency. Its performance only deteriorates when T_{VO} is 3 s, which is understood, as the VO operates at the edge of its capabilities. Nevertheless, position estimates do not degrade as fast as for other models.

- It is interesting to observe that for T_{VO} set to 3 s, errors from trajectories on tarmac grow in the negative direction for models where slip is treated as pseudo-control input, whereas for Integrated Slip Estimation, they grow in the positive direction. It is understood that for the former two, even though VO measurements have a considerable higher measurement error, they dominate the fusion. The discrepancy between VO and WO confuses the filter (in the case of Separate Slip Estimation) because only one process noise drives the system. In the case of Integrated Slip Estimation, wheel slip is also driven by the process noise, and the discrepancy between VO and WO may be used to identify this noise better. Also, knowing higher VO measurement error associated with such large T_{VO}, the filter prefers WO. As a result, the fused position is closer to the unfiltered WO estimation that overestimates the travelled distance.

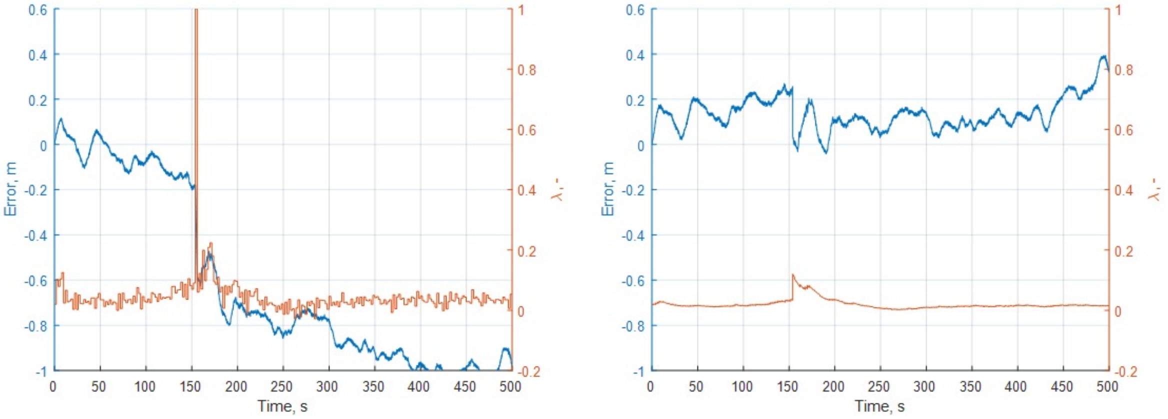
2.3.4 VO Error Handling

Both Filter Free Estimation and Separate Slip Estimation differ from the Integrated Slip Estimation as they treat slip as a pseudo control parameter. There is no notion of how good the slip estimation is in these approaches. Assuming no terrain can result in slip estimation spiking at specific time instances, such spikes can indicate an error, most likely, in VO measurement. As slip is a function of VO, any inaccurate measurement contributes directly to the estimate. With large T_{VO}, these corrupted values are used longer to "correct" WO until new VO measurement is available, resulting in increased position estimation error. The low-pass filter used in the Integrated Slip Estimation model allows filtering these slip estimates providing better VO error handling in the fusion process.

Spike in slip estimation may happen simultaneously when the filter produces a sharp change in position estimation, which can indicate a fault in measurement. As an example, a single trajectory on grass is investigated. Note a rapid change in the error in Fig. 2.19(b) for T_{VO} set to 2 s in one green plot around the second 150. It is best seen when overlaying error in position estimation with estimated slip as presented in Fig. 2.21. Fig. 2.22 presents two left camera images from the stereo pair that generated the VO estimate under question. According to RTK, there was 0.4 m displacement; however, the VO measurement reported only 0.04 m distance travelled, which is an order of magnitude lower than expected. It can be acknowledged that images present a displacement with significant rotation, which, together with complex texture, could mislead the VO algorithm. In this example, as seen in Fig. 2.21, it was a single event in the trajectory. In the case of pseudo control u_λ estimation presented on graph 2.21(a), once calculated, the wrong slip is used for another 2 seconds and the system never fully recovers. However, the Integrated Slip Estimation model can recover from this error because of the exponential nature of the low-pass filter, and thus it provides better position estimation in between VO measurements.

2.4 Summary

This chapter presents a new approach to estimating wheel slip when fusing VO and WO by integrating the slip state inside an EKF. This Integrated Slip Estimation model was first tested



(a) Pseudo-control u_λ from Separate Slip Estimation

(b) λ from Integrated Slip Estimation

Figure 2.21: Single trajectory with overlaid error in position estimation (blue) and slip estimation (orange). Note how the system with Integrated Slip Estimation better filters wheel slip due to its low-pass filter. As a result, the error in position estimation is relatively small compared to the system with wheel slip treated as a pseudo-control input.

(a) Left camera image

(b) Next left camera image

Figure 2.22: Left camera images which produced erroneous VO measurement.

and verified using low-fidelity simulation and later validated during field trials and compared against other similar approaches. The new method allows the EKF to choose which sensor it relies on more for a given terrain type. It can become a powerful tool when traversing across different types of terrain as it enables the estimation of slip variance. Chapter 3 will exploit the filter's knowledge of wheel slip estimation's variance to reduce VO usage in a reactive scheme.

CHAPTER 3

ADAPTIVE EKF AND REACTIVE VO SCHEDULING

In the previous chapter, a new model was proposed for the fusion of VO and WO with integrated wheel slip estimation. It was shown that it has a single σ_β^2 parameter (a variance in w_β process noise) which allows selecting whether the system trusts VO or WO more. The parameter was analysed and tuned manually, but it is terrain-dependant. For a solid surface (like tarmac), one can expect the covariance to be relatively small, and thus WO could be treated as the primary source of localisation. However, wheel slip may vary with the less uniform surface, resulting in a need for more frequent VO measurements. It may lead to higher power output and computational demand that could be otherwise spent performing other critical or scientific operations. To address this problem, section 3.2 investigates the adaptive nature of EKF so that the σ_β may be adjusted online. Two solutions are explored and compared using simulated data in section 3.2.4. In section 3.2.5 the algorithms are validated using the Hengrove dataset mentioned in the previous chapter. Furthermore, reactive VO scheduling is proposed to reduce the number of VO measurements required to free computational resources whilst maintaining good navigational accuracy. Various VO trigger strategies are discussed and compared in section 3.3 but only for the most promising adaptive EKF candidate.

This chapter is based on our IROS paper [51]. Also in this chapter, in case of any discrepancies, this thesis supersedes the papers.

3.1 Introduction

There are three main areas in mobile robot navigation where knowledge about the terrain may improve the localisation accuracy, lower energy usage, or decrease computations. The first is *a priori* knowledge of the terrain, where either wheel slip can be evaluated from imagery [55] or information about the slip was calibrated on a representative terrain [25]. The second area deals with how wheels interact with the terrain. Novel approaches may see reduced energy consumption and extended traverse duration [56, 57]. Finally, an intelligent scheduler may be proposed that trades the number of measurements made by localisation sensors for an acceptable increase in the variance in position estimation. The first work in that area is presented in [58], where the decision on when to perform accurate localisation is based on belief state. The work is then expanded into a more detailed resource trade-off analysis in [59]. Camera scheduling may also be employed in a situation where the vehicle speed is variable, and image acquisition needs to be dynamically adjusted to maintain required overlap, as presented in [60].

This work is similar to [58] and [59], but it differs in two aspects. Firstly, it is assumed that no *a priori* information is available, i.e. no feature maps are generated beforehand for VO, which makes it another dead reckoning sensor. Secondly, rather than using a belief Markov decision process, a process noise variance σ_β^2 is analysed, directly impacting the slip estimation, translating to poorer quality of WO measurements. We also believe to be the first to adapt VO measurement based on slip estimation.

The high-level system overview is depicted in Fig. 3.1, where relations between EKF, adaptation and VO scheduling are provided. EKF's innovation residual r, as defined in (1.7) as $r = z - h(\tilde{x}, u)$ (time index skipped for clarity), is used during the adaptive step. The trigger component utilised the residual together with estimated mean values for noises and their respective covariance matrices to decide whether to schedule VO measurement. The same results of

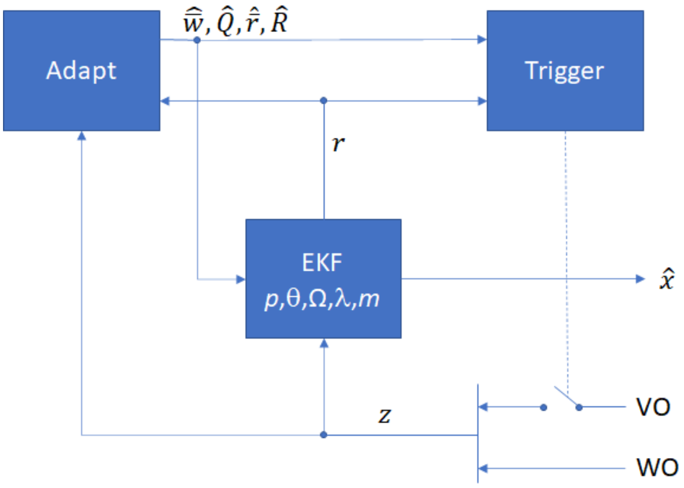


Figure 3.1: An overview of the adaptive process for Kalman Filter and Visual Odometry triggering. Before EKF provides its estimation of the state, $\hat{x}$ based on measurements, z, the measurement residual, r is sent to the adapt algorithm. It provides updated covariance matrices and means values for process and measurement noises. These are not only used during the correction step of EKF but also used to determine if VO measurement should be performed.

adaptation are then used in the next EKF iteration. Note that there is an expectation that some noises, in particular, w_β, have unknown means and covariances. While means could be estimated as a bias by augmenting the Kalman Filter state [61, 62], covariances require an adaptive filter approach. To avoid defining biases in EKF, the means are estimated in the adaptation module and treated as nonlinearity in the model. Note that (1.6) and (1.11a) already take that into account.

If the observation noises were white, as assumed and expected by EKF, adding an adaptive filter would give only a slight advantage. However, when running the test for whiteness, explained for example in [48], it becomes evident that neither WO nor VO observation noises are white and unbiased. Even though an adaptive filter still assumes white noise, it allows to estimate mean (bias) and modifies covariances so that they may better represent the non-white noise. To properly handle non-white noises, a colouring filter is required, however, for this work, it was skipped in favour of a more straightforward approach of just implementing an adaptive filter.

3.2 Adaptive EKF

There is no single solution for adaptive filtering. Out of four categories defined in [63], namely: Bayesian, maximum likelihood, correlation, and covariance matching, two most popular candidates were selected for investigation and implementation. Regardless of the category, the adaptation process remains the same as listed in Algorithm 1.

Algorithm 1 Adaptive Filtering Process

```
1: Prediction, as defined in (1.4) and (1.5)
2: $R$-adaptation: observation noise adaptation
3: Correction, as defined in (1.11)
4: $Q$-adaptation: process noise adaptation
```

Note that for R-adaptation the innovation residual (1.7) is used. Interestingly, in [64] the author proposes to use the post-fit residual, given as $r' = z - h(\hat{x}, u)$ (note $\hat{x}$ instead of $\tilde{x}$), for the R-adaptation performed after the correction step. It has the bonus of ensuring that the R matrix always remains positive. However, in the case of this work, there was no visible benefit of employing this technique compared to the traditional approach in adaptive filtering. Thus, to simplify the process, both adaptive filters follow Algorithm 1.

3.2.1 Adaptive Limited Memory Filter

The first adaptive filter identified for investigation is Adaptive Limited Memory Filter (ALMF), first introduced in [65]. The algorithm enjoys popularity among researchers even in more recent times [66, 67] proving its usefulness despite its age. A moving window provides an unbiased estimator of w, Q, r and R. The first impression is that such an algorithm would not be optimal for on-board frequent position estimation due to its batch processing (moving window). However, the authors of [65] provide an implementation that alleviates this problem, making it suitable for demanding applications. As they state, the provided implementation is necessary for time-variant systems.

Based on [65], the unbiased estimators for the model with Integrated Slip Estimation are defined as

$$\bar{r}(k) = \frac{1}{N}\sum_{j=k-N+1}^{k} r(j) \tag{3.1a}$$

$$\hat{R}(k) = \frac{1}{N-1}\sum_{j=k-N+1}^{k}\Big(\big(r(j)-\bar{r}(k)\big)\big(r(j)-\bar{r}(k)\big)^T - \frac{N-1}{N}H(j)\tilde{P}(j)H(j)^T\Big) \tag{3.1b}$$

$$\bar{w}(k) = \frac{1}{N}\sum_{j=k-N+1}^{k} w(j) \tag{3.1c}$$

$$\hat{Q}(k) = \frac{1}{N-1}\sum_{j=k-N+1}^{k}\Big(\big(w(j)-\bar{w}(k)\big)\big(w(j)-\bar{w}(k)\big)^T - \frac{N-1}{N}G^{\dagger}(j)\Big(F(j)\hat{P}(j-1)F(j)^T - \hat{P}(j)\Big)G^{\dagger}(j)^T\Big) \tag{3.1d}$$

where

$$w(j) = G^{\dagger}(j)\Big(\hat{x}(j) - f\big(\hat{x}(j-1), u(j), 0\big)\Big) \tag{3.2}$$

Eqs. (3.1a) and (3.1c) are trivial as they are simple mean estimated over N samples. As for the (3.1b) ((3.1d) follows similar steps), recall (1.8) given as $S(k) = E[r(k)r(k)^T] = H(k)\tilde{P}(k)H(k)^T + R(k)$. $\hat{R}$ is calculated by taking the expected value of (1.8), such that

$$E[S(k)] = E\big[H(k)\tilde{P}(k)H(k)^T + R(k)\big] = E\big[H(k)\tilde{P}(k)H(k)^T\big] + E\big[R(k)\big] \tag{3.3}$$

Eq. (3.3) can be evaluated for $\hat{R} = E\big[R(k)\big]$ to match (3.1b) as follows (assuming indices from one to N for better readability)

$$E\big[R(k)\big] = E\big[S(k)\big] - E\big[H(k)\tilde{P}(k)H(k)^T\big] \tag{3.4a}$$

$$\hat{R}(k) = E\Big[E\big[r(k)r(k)^T\big]\Big] - \frac{1}{N}\sum_{j=1}^{N} H(j)\tilde{P}(j)H(j)^T \tag{3.4b}$$

$$\hat{R}(k) = E[r(k)r(k)^T] - \frac{1}{N}\sum_{j=1}^{N} H(j)\tilde{P}(j)H(j)^T \tag{3.4c}$$

$$\hat{R}(k) = \frac{1}{N-1}\sum_{j=1}^{N}\Big(\big(r(j)-\bar{r}(k)\big)\big(r(j)-\bar{r}(k)\big)^T\Big) - \frac{1}{N}\sum_{j=1}^{N} H(j)\tilde{P}(j)H(j)^T \tag{3.4d}$$

$$\hat{R}(k) = \frac{1}{N-1}\sum_{j=1}^{N}\Big(\big(r(j)-\bar{r}(k)\big)\big(r(j)-\bar{r}(k)\big)^T - \frac{N-1}{N}H(j)\tilde{P}(j)H(j)^T\Big) \tag{3.4e}$$

There are two highlights regarding (3.4): Firstly, the expected value of an expected value is just an expected value, i.e. $E\big[E[a]\big] = E[a]$. Secondly, the expected value $E\big[H(k)\tilde{P}(k)H(k)^T\big]$ was evaluated as a simple average over N samples. It comes from the definition of expected value as mean.

Interestingly, the classic Kalman Filter usually has one process noise per state. When they are of different sizes (in this work, there are five states and two process noises), the state transition function f describes how the process noise is mapped into states. As a result, a Jacobian G defined in (1.6b) may not be a square matrix. Consequently, only a pseudoinverse matrix exists, $G^{\dagger}$, as present in (3.1d) and (3.2). In (3.1) a single-window N is defined for both R-adaptation and Q-adaptation, however, realistically these may be different values. Even more, VO and WO could have different window sizes. In [65] the assumption was that only one measurement source is used, which directly impacts process noise estimation. In the case of sensor fusion, this process is not as straightforward, and correct window sizes need to be established.

3.2.2 Sage-Husa Adaptive Bayes Filter

The algorithm proposed by Sage and Husa in [68] and Husa's dissertation [69] is another viable solution for an adaptive filter due to its recursive nature. Even [65] identifies it as the most promising solution for online applications, despite criticising the assumptions to be "rather vague". The iterative nature of this algorithm makes it also popular with modern applications [70], and other filtering techniques such as with UKF [71].

The algorithm exists in two forms: optimal and suboptimal. The former estimates noise statistics by maximising the *a posteriori* density function $p[x(k), w, Q, r, R|z(k)]$. This approach utilises past information in a complex manner that requires storing matrices used in each filter's prediction and correction steps. On the other hand, the suboptimal solution used in this work is a recursive algorithm that simplifies the optimal solution. Because of that, it is also the more popular version. An optimal version could be evaluated in the future to compare its accuracy and complexity with the suboptimal counterpart. Formulas for $\bar{r}$, $\hat{R}$, $\bar{w}$, and $\hat{Q}$ used in Integrated Slip Estimation model are given as

$$\bar{r}(k) = \frac{1}{N}\big((N-1)\bar{r}(k-1) + r(k)\big) \tag{3.5a}$$

$$\hat{R}(k) = \frac{1}{N}\Big((N-1)\hat{R}(k-1) + \big(r(k) - \bar{r}(k)\big)\big(r(k) - \bar{r}(k)\big)^T - H\tilde{P}(k)H^T\Big) \tag{3.5b}$$

$$\bar{w}(k) = \frac{1}{N}\big((N-1)\bar{w}(k-1) + w(k)\big) \tag{3.5c}$$

$$\hat{Q}(k) = \frac{1}{N}\Big((N-1)\hat{Q}(k-1) + G^{\dagger}(k)\Big(\big(\hat{x}(k) - \tilde{x}(k)\big)\big(\hat{x}(k) - \tilde{x}(k)\big)^T + \hat{P}(k) - F(k)\hat{P}(k-1)F(k)^T\Big)G^{\dagger}(k)^T\Big) \tag{3.5d}$$

There are a few remarks regarding the (3.5):

- (3.5d) differs from what was presented in [51]. The IROS paper has an incorrect formula missing the $\big(\hat{x}(k) - \tilde{x}(k)\big)\big(\hat{x}(k) - \tilde{x}(k)\big)^T$ component. The formula was correctly implemented making the results correct and not invalidating the contribution made.

- Time indices associated with (3.5b) could be debatable. From the original papers, [68, 69] it seems as if the R-adaptation should be made after the EKF correction. However, in the literature, it is common to perform the R-adaptation as mentioned in Algorithm 1. This approach is also in line with ALMF, which aims to remove the bias from the observation noise prior to the EKF correction. We assumed this was the correct behaviour and updated the Sage-Husa algorithm accordingly.

- The original algorithm would see N set to k; however, this approach may not be best suited for non-stationary and time-invariant systems.

3.2.3 Implementation Considerations

Some valuable considerations when implementing adaptive Kalman Filter have already been discussed in [65]. Similarly to this approach, all values on the main diagonal of covariance matrices are reset to absolute values. Negative values on the main diagonal may arise as a result of subtraction in (3.1b), (3.1d), (3.5b), and (3.5d).

Even though a constant time step Δt is assumed, this may not always be satisfied in a real-world application. If the time step between two consecutive predictions is small enough, the Jacobian G (defined in (1.6b) as $\frac{\partial f}{\partial w}|\hat{x}, u, \bar{w})$ may become ill-conditioned rendering $G^{\dagger}$ unusable. Should this happen, the Q-adaptation step is skipped. It was determined by trials and errors that a satisfactory test to evaluate the usability of G is given as

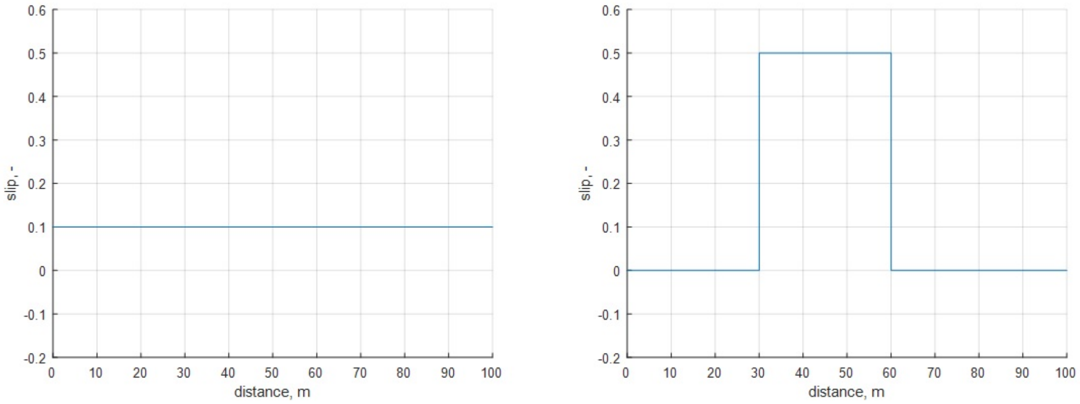
$$\|G\|\|G^{\dagger}\| - T_0 > 0.01 \tag{3.6}$$

Where $\|G\|\|G^{\dagger}\|$ is 2-norm condition number of matrix G and T_0 is time constant of the low-pass filter driving wheel slip estimation.

It is also important how the correction step is implemented. There could be two separate corrections for each sensor alone with only VO and WO measurements. If only milliseconds were between the measurements, they would not impact EKF performance. However, this tiny time step may significantly impact the adaptation. It is one of the reasons (3.6) is required. A third correction step is included for combined VO and WO measurements should they happen within 10 ms to better handle such a situation. Even though with this approach the performance is better, the check (3.6) is still necessary.

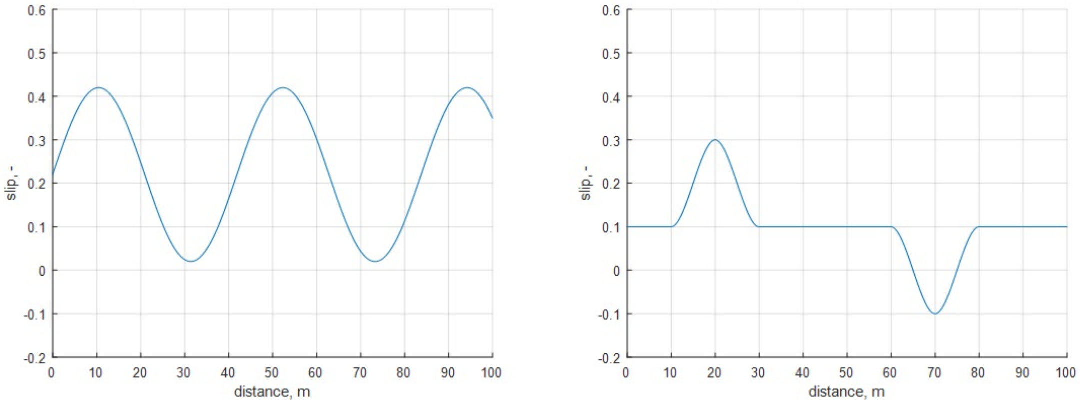
3.2.4 Basic Simulations

In this section, both algorithms are compared using the test cases as presented in Fig. 3.2. Because VO and WO happen at different times, each adaptive algorithm has its own window size and gain (from now on both are referred to as gains), such that N_{WO} will be used for R_{WO}-adaptation, N_{VO} for R_{VO}-adaptation, and N_Q for Q-adaptation. With simulated data, all measurements are expected to have zero mean noise, therefore, (3.1a) and (3.5a) are skipped and zero mean residual is assumed. The key challenge is to identify gains for optimal filter performance. Simulations were done for each test case, where N_{VO} and N_{WO} were varying from 10 to 100 every 10, whereas



(a) Test Case 1: Constant Non-Zero Slip

(b) Test Case 2: Step-Change in Slip

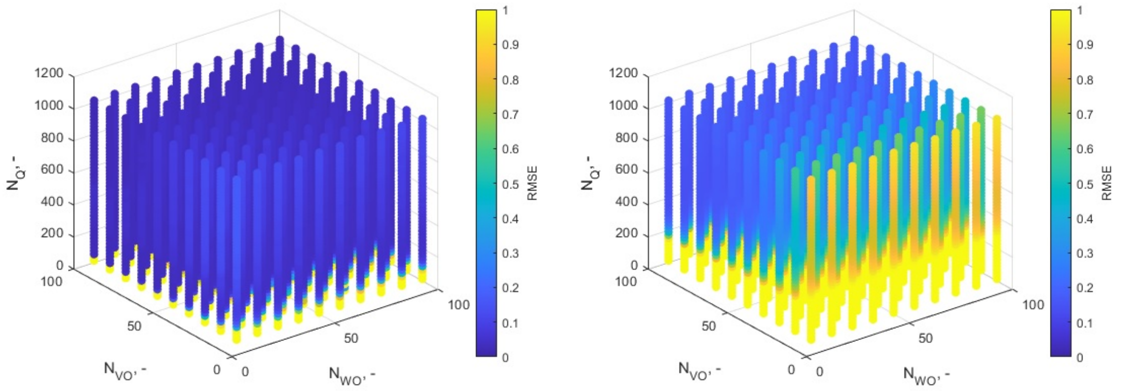


(c) Test Case 3: Sinusoidal Slip

(d) Test Case 4: Bumps

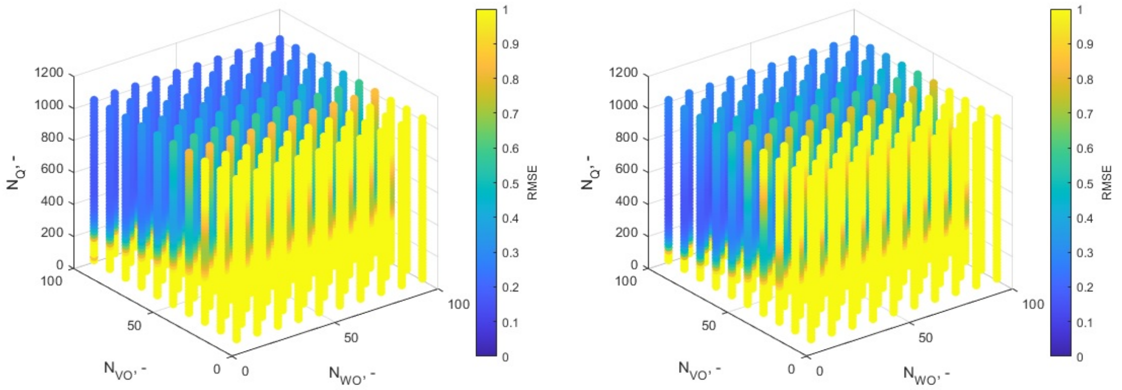
Figure 3.2: Ground-truth slip profile for four test cases used to evaluate ALMF and Sage-Husa filters. Note that the last test case is the same as in Fig. 2.5 but with a different scale. Test cases two and three are quite unrealistic, however, they are good at capturing how well filters adapt to either rapid or continuous change in wheel slip.

N_Q was varying from 10 to 1010 every 20. The edge values were selected based on trials and errors, noting that optimal parameters for simulation for both algorithms were within the range. From a practical perspective, increasing values by 10 or 20 reduced the number of simulations without any major impact on RMSE. Other EKF parameters were set as $T_0 = 20$ s, and $\sigma_\beta = 1$. Each simulation was repeated for different $T_{VO} : 0.2, 1, 2, 3$ s. After simulations, RMSE was calculated for each run and then averaged across all test cases for a given T_{VO}. Because ALMF gave worse RMSE compared to the Sage-Husa filter, only the latter is discussed with its results presented in Fig. 3.3, where the colours of points indicate RMSE for a given set of gains. Sets for which RMSE was the smallest for both algorithms are listed in Table 3.1. Note how Sage-Husa (Sim) provides a much smaller RMSE compared to ALMF (Sim). It is especially visible for $T_{VO} = 3$ s (bottom row). It is of particular interest given that the aim is to reduce the frequency of VO



(a) T_{VO} = 0.2 s

(b) T_{VO} = 1.0 s



(c) T_{VO} = 2.0 s

(d) T_{VO} = 3.0 s

Figure 3.3: RMSE for the Sage-Husa filter across all test cases for different T_{VO} using simulated data. Deeper blue colour indicates a smaller RMSE. Note how, in general, large RMSE was associated with small N_Q and small N_{VO}.

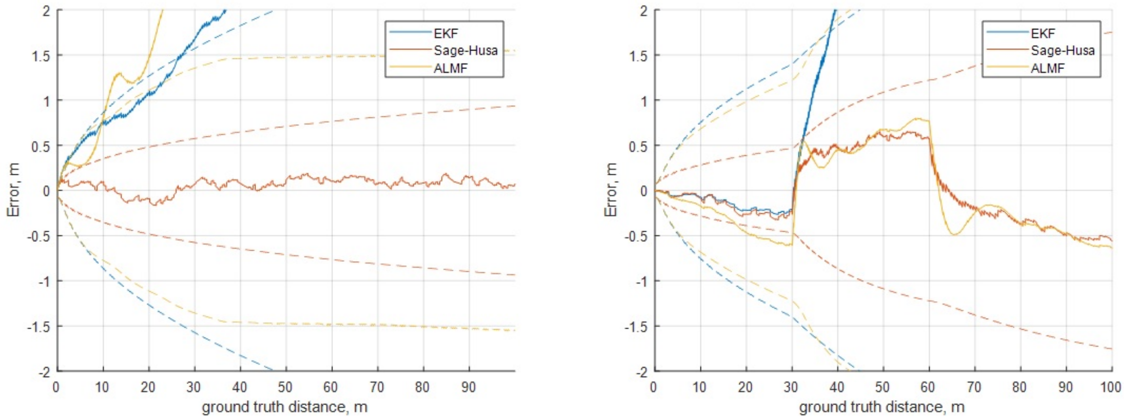
while maintaining good localisation performance.

The initial assumption was that the same parameters obtained during simulations could be used as-is when validating the Sage-Husa filter using the Hengrove dataset. However, when the algorithm was subjected to real-world data, its performance was far worse than non-adaptive

Table 3.1: Optimal gains for adaptive filters across all test cases

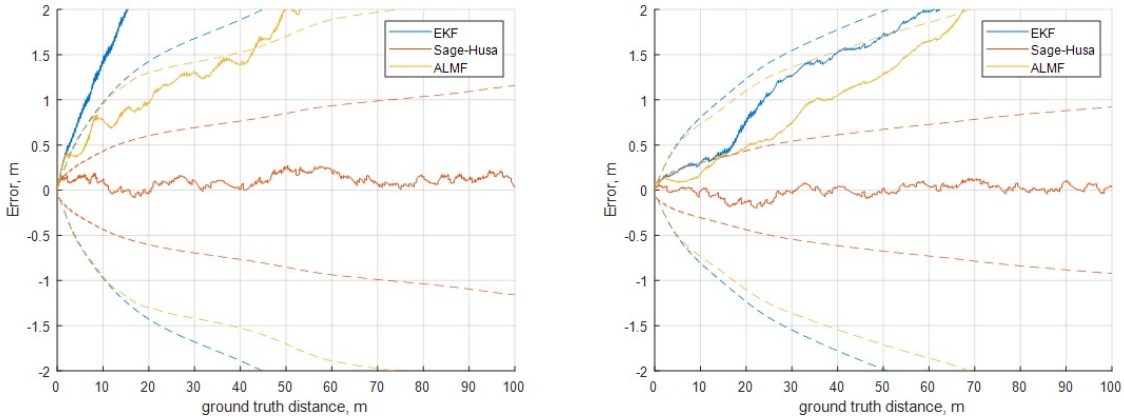
	ALMF (Sim)				Sage-Husa (Sim)				Sage-Husa (Hengrove)			
T_{VO}	N_{VO}	N_{WO}	N_Q	RMSE	N_{VO}	N_{WO}	N_Q	RMSE	N_{VO}	N_{WO}	N_Q	RMSE
0.2	80	40	30	0.008	100	10	810	0.001	30	100	730	0.139
1.0	50	80	10	0.053	100	10	510	0.018	100	100	370	0.225
2.0	70	10	10	0.086	100	10	490	0.027	100	100	570	0.277
3.0	30	50	30	1.298	100	40	370	0.020	30	100	1010	0.378

EKF. As such, the same process for obtaining gains was repeated using the Hengrove dataset with the new sets of parameters listed in Table 3.1 and labelled as "Sage-Husa (Hengrove)". The results of the validation are described in the next section 3.2.5, but what can be noted at this stage is that the parameters vary much. For example, there could be some correlation between T_{VO} and N_Q when inspecting "Sage-Husa (Sim)", but the real-world data contradicts it. Furthermore, Fig. 3.3 indicates lower RMSE for higher N_{VO}, which is also not the case when using the Hengrove dataset. It shows that simple simulations that do not capture realistic noises could be misleading. As such, no conclusions are drawn regarding individual gains. The only valid observation that can be made regarding the Sage-Husa filter is whether the search for optimal parameters should have been extended. In the case of simulated data, N_{VO} reached its maximum



(a) Test Case 1: Constant Non-Zero Slip

(b) Test Case 2: Step-Change in Slip



(c) Test Case 3: Sinusoidal Slip

(d) Test Case 4: Bumps

Figure 3.4: Errors in position estimation for the Sage-Husa filter with T_{VO} = 3 s and gains as defined in Table 3.1. Blue lines represent EKF, red Sage-Husa filter, and yellow ALMF. Dashed lines are 3σ confidence margins with colours associated with models. Other parameters used, σ_β = 1, T_0 = 20 s.

(100), whereas, for the Hengrove dataset, it was N_{WO} (100) in all cases, and N_Q (1010) for T_{VO} = 3 s. However, in the case of simulated data, the RMSE was already very small, thus, it was decided not to extend the search. In regards to the real-world data, even though existing sets could be suboptimal, it allows visualising better the impact of reactive VO scheduling that is explained in section 3.3 and which successfully rectifies all erroneous trajectories.

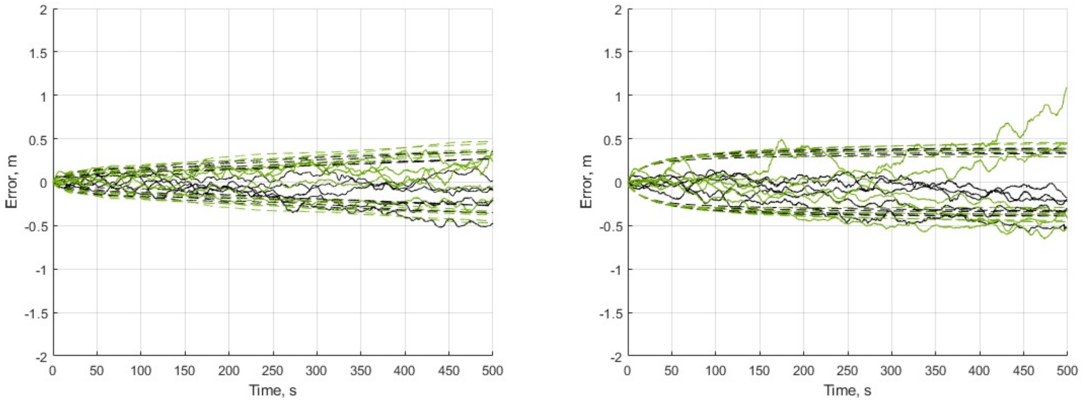
What is vital at the simulation stage is that ALMF was outperformed by the Sage-Husa filter, which is from now on referred to as the Adaptive EKF (AEKF). To illustrate it better, example errors in position estimation for both algorithms using simulated data with T_{VO} = 3 s are presented in Fig. 3.4.

3.2.5 Validation using Real-World Data

After simulations, the chosen AEKF is validated using data from Hengrove trials. As indicated earlier, the N parameters obtained during simulations were poor when the algorithm was exposed to realistic noises and measurements. As such, a new set of N parameters was obtained. Based on the findings from the section 2.3.2, during the second search campaign, T_0 was set to 20 s and the initial σ_β was set to 0.5. The resulting parameters were listed in Table 3.1 and using these exact same parameters for each T_{VO}, plots in Fig. 3.5(a), 3.5(b), 3.5(c), and 3.5(d) were generated. Good performance, with some minor exceptions, is well-understood given that the adaptive parameters were found using the same data. Note how all trajectories in Fig. 3.5(d) are improved compared to EKF in Fig. 2.18.

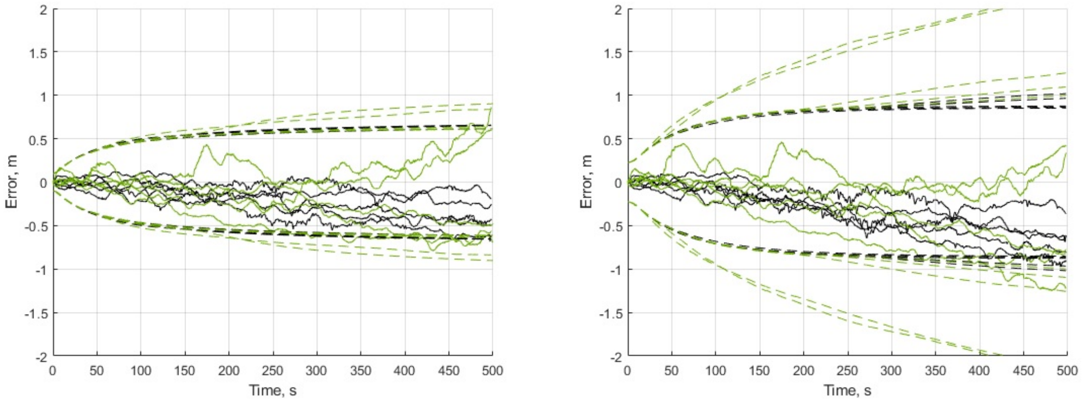
Note that adaptive gains were found with combined VO-WO measurements disabled, i.e. only individual correction steps were allowed even when the time between them was a few milliseconds (compare with section 3.2.3). With the combined measurements allowed, the resulting N_Q was very small. As a consequence, the 3σ confidence margin was too strict making most errors fall outside the margin (even though RMSE was small). It is an effect of analysing only RMSE without taking the filter's covariance into account. A more comprehensive parameter search could be done in the future, for example, by excluding parameters for which the errors exceed 3σ confidence margin.

In the next section, a reactive VO scheduling algorithm is proposed to improve the quality of AEKF while performing VO as rarely as possible. Because of that, the system with VO measurements every 3 s is inspected in more detail beforehand. Note that in Fig. 3.5(d), even though most of the errors are within the 3σ confidence margin, there is a noticeable trend that could lead to errors growing outside the margin if the system was operated longer. A possible solution to this problem is to estimate the mean residual (as defined in (3.5a)), which was not taken into consideration during simulations. Examples of errors in position estimation are presented in Fig. 3.5(e) and 3.5(f). There are two immediate observations: firstly, this approach does not work for small T_{VO}. It is understood that there is an observability issue, where it is impossible to determine whether the discrepancy in observation comes from the wheel slip



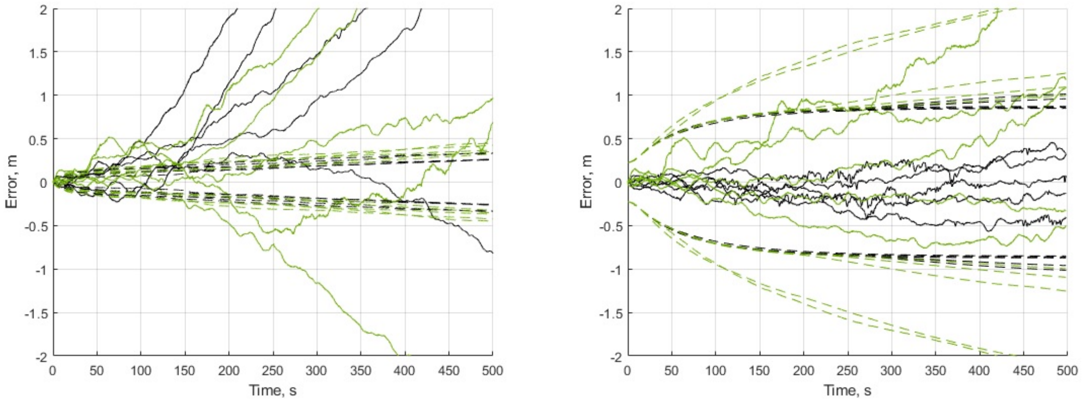
(a) T_{VO} = 0.2 s

(b) T_{VO} = 1.0 s



(c) T_{VO} = 2.0 s

(d) T_{VO} = 3.0 s



(e) T_{VO} = 0.2 s with non-zero $\bar{r}$

(f) T_{VO} = 3.0 s with non-zero $\bar{r}$

Figure 3.5: Error in position estimation for Adaptive EKF using the Hengrove dataset. Black plots are for trajectories on the tarmac and green for those on grass. Notice improved 3σ confidence margin tracking (dashed lines) compared to EKF. The last two plots are with non-zero $\bar{r}$ which improves performance for longer T_{VO} but vastly deteriorates it for smaller.

or if there is a measurement bias that should be captured by $\bar{r}_{VO}$. However, these problems are not visible for longer T_{VO}. It is understood that with the larger T_{VO}, the initial R_{VO} is big enough to accommodate sources of discrepancies in the measurements better. Furthermore, most of the trajectories seem to have a rather stable trend, especially on the tarmac (black). On the other hand, there are some drawbacks to this approach: there are a few grass trajectories (green) in Fig. 3.5(f) which have errors of a bigger magnitude than before. It is, however, a good starting point for a reactive VO scheduling algorithm to rectify these by requesting additional VO measurements while maintaining that relatively flat trend of errors. In addition, it should be investigated in the future if the reason for which system with T_{VO} = 3 s works better when the mean residual is estimated comes from a VO measurement bias that is associated with reduced image overlap. In section 2.1.3, it was highlighted that increased T_{VO} in this work is only associated with increased measurement noise (standard deviation) with example values presented in (2.12). It is thus likely, that there could be additional bias that is being picked by the adaptive filter.

3.3 Reactive VO Scheduling

So far, VO was always performed at a fixed frequency. From the results, it can be noted that the position estimation is quite good on tarmac even when performing VO every 3 s. However, for some trajectories on the grass, more frequent VO measurements would be required to provide a similar level of accuracy. Therefore, the goal is to allow the algorithm to choose when the next VO measurement should happen. The aim is to optimise the number of image acquisition and VO measurements which may free up some computational resources or even lead to lower power consumption. A system with VO measurements every 3 s is used as a baseline. For the purpose of this task, adaptive parameters are fixed to $N_{VO} = 30$, $N_{WO} = 100$, and $N_Q = 1010$ according to Table 3.1.

3.3.1 Trigger Strategies

As indicated in Fig. 3.1, the Trigger component is used to determine if VO measurement is required. It utilises information about the process and measurement noises, however, other information can be fed into it to build more comprehensive trigger strategies. In this work, four strategies are evaluated in the following order:

1. A known VO error - the VO library used in this project provides an error code for each VO estimate. So far, all VO measurements were fused as-is. However, knowing that there was an error, it is possible to skip the measurement and schedule a new one immediately after. When a known error occurs, the predicted measurement is stored in a temporary variable

and added to the next measurement. As such, (2.11a) can be rewritten as

$$z_{VO}(k) = \tilde{p}(k) - \tilde{m}(k) + n_{VO}(k) + z_{VO_{acc}}(k) \tag{3.7a}$$

$$z_{VO_{acc}}(k) = \begin{cases} z_{VO_{acc}}(k-1) + \tilde{p}(k-1) - \tilde{m}(k-1) & \text{if VO failed at } k-1 \\ 0 & \text{otherwise} \end{cases} \tag{3.7b}$$

Note that in this approach, the covariance matrix R_{VO} is not increased. It could be argued whether a successful VO measurement should inherit errors of past failures, however, it is left for an adaptive algorithm to adjust the matrix.

2. An unknown VO error - as discussed in section 2.3.4, there are moments where VO provides a wrong estimate but the algorithm does not report any error. EKF can already handle it to some extent because such errors are treated as spikes in wheel slip. The low-pass filter allows to 'forget' the erroneous value back to $\bar{\sigma}_\beta$. Nevertheless, it is advantageous to perform another VO measurement after such an event to reconfirm the wheel slip estimate. To determine if the VO measurement was statistically valid, a squared Mahalanobis distance [72] is calculated as

   $$d^2(k) = \big(r(k) - \bar{r}(k)\big)^T S(k)^{-1} \big(r(k) - \bar{r}(k)\big) \tag{3.8}$$

   Because the model is limited to one dimension, $d(k)^2$ follows a Chi-square distribution with one degree of freedom. In the appropriate tables, one can find that for such measurement to be valid with 95% probability (the equivalent of 2σ), the squared distance needs to be not greater than 3.841. Even though a measurement may not be statistically valid, it is still fused, and measurement noise statistics are updated accordingly. We noticed that a more relaxed rule was required once identified problems reported in the introduction to this chapter were fixed. Furthermore, 95% probability was also selected by NASA in their work on slip prediction using squared Mahalanobis distance [22, 53], which seems to be a proven trade-off.

3. The maximum time from the last VO measurement $T_{VO_{MAX}}$ - VO algorithm compares the current stereo pair with the previous one. If the distance between the two is too large, the estimate may be erroneous. Because the robot is assumed to move at a constant speed, this translates to a time between VO measurements. For this work, we assume a constant value of $T_{VO_{MAX}} = 3$ s.

4. Increase in σ_β^2 by 1.6% - w_β process noise is responsible for driving the wheel slip estimation λ. An increase in the noise covariance may result from higher uncertainty of the slip, which should prompt a VO measurement to provide a fresh estimate of λ. With all other trigger strategies implemented and enabled, a simulation campaign was performed (similar to one presented in section 3.2.4) for a number of $\frac{\sigma_\beta^2(k)}{\sigma_\beta^2(k-1)}$ thresholds. The smallest RMSE was achieved for 1.016, hence the 1.6% value in the strategy.

3.3.2 Adapting N_{VO}

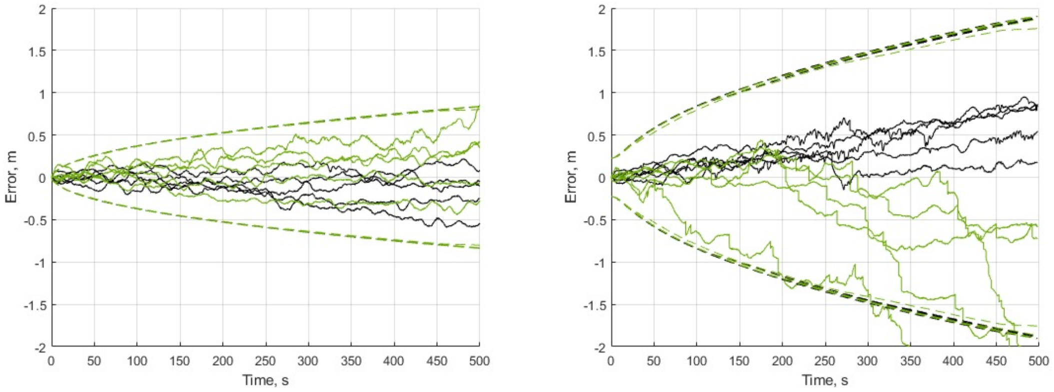
One of the trigger strategies evaluates the quality of the VO update to determine if there was an error or not. Even if the measurement is not statistically valid with 95% probability, the measurement is fused to allow covariance matrix R_{VO} to expand. The estimated mean value $\bar{r}_{VO}$ is more problematic because updating it based on an erroneous measurement would only propagate that error further. To lower the negative impact it might have, should an unknown VO error be detected, N_{VO} is set to 100 (30 by default). By increasing the gain, the negative impact of the residual on the adaptation is reduced, while the information is still fused as opposed to skipping R_{VO}-adaptation altogether.

3.3.3 Results

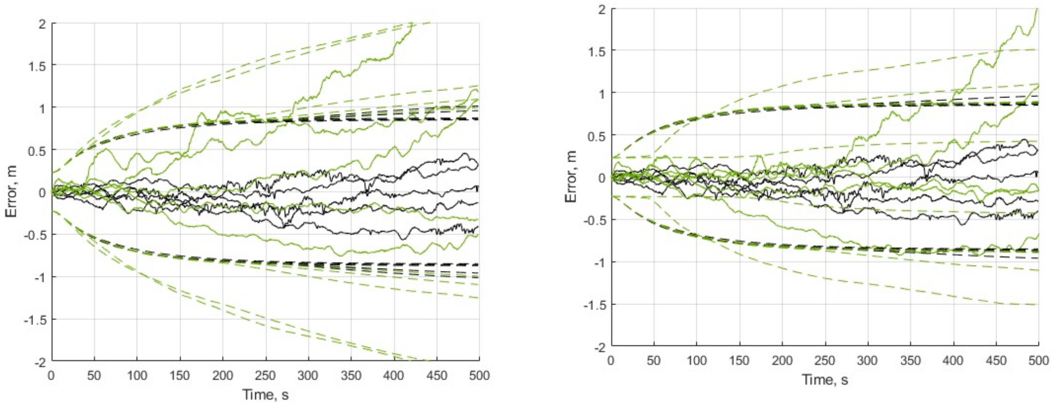
In this section, results of AEKF with reactive VO scheduling algorithm are provided. It is shown how strategies compare to each other (e.g. a strategy three without the reduction of N_{VO}) and how the system compares with EKF as defined in chapter 2. For each model, RMSE and the number of VO measurements are compared. Results are presented in Fig. 3.6. In addition, Fig. 3.7 presents a tradeoff between RMSE and the number of VO measurements for individual trajectories and as a mean for each model.

The most accurate model, presented in Fig. 3.6(a), is EKF with T_{VO} = 0.6 s but at the expense of 834 VO measurements. VO update rate was selected to mimic ExoMars and Sample Fetch Rover (SFR) behaviour. In the case of the ExoMars rover, the nominal speed is 1.1 cm/s, and VO is going to perform every 10 s, which translates to an 11 cm distance between stereo pairs. For SFR, the nominal speed is expected to be reduced to 6.67 cm/s, while VO is expected to perform every 2 s giving 13.34 cm separation between image frames. When capturing the Hengrove dataset, the robot was driven at 20 cm/s which with T_{VO} = 0.6 s gives a 12 cm distance between images - similar to ExoMars and SFR.

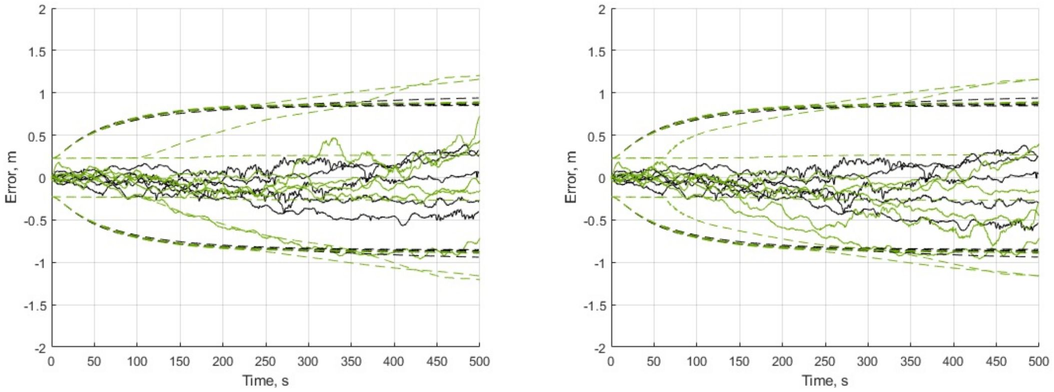
EKF and AEKF with T_{VO} = 3.0 s presented in Fig. 3.6(b) and Fig. 3.6(c) are only shown for convenience to better illustrate the impact of extending the time between VO measurements and adding adaptation on errors in position estimation. Plots start to improve when reactive VO scheduling is added. Beginning with Fig. 3.6(d), adding only strategies 1-3 without adapting N_{VO} (section 3.3.2) delays error accumulation for the most prominent grass trajectories (green). The positive impact of adapting N_{VO} when a suspected error is detected is seen when analysing Fig. 3.6(e). All error plots are already with a 3σ confidence margin. Surprisingly, adding the fourth strategy does not improve RMSE. Visually, the error plots seen in Fig. 3.6(f) seem similar to the previous model. However, as seen in tradeoff plots in Fig. 3.7, the average RMSE is slightly increased, as is the number of VO measurements (Fig. 3.7(b)). It could be concluded that more VO measurements may negatively impact the performance of the model, however, given that this analysis is done only in one dimension, further investigation should be done when assessing higher dimensions with additional sensors, such as IMU.



(a) EKF with T_{VO} = 0.6 s (b) EKF with T_{VO} = 3.0 s

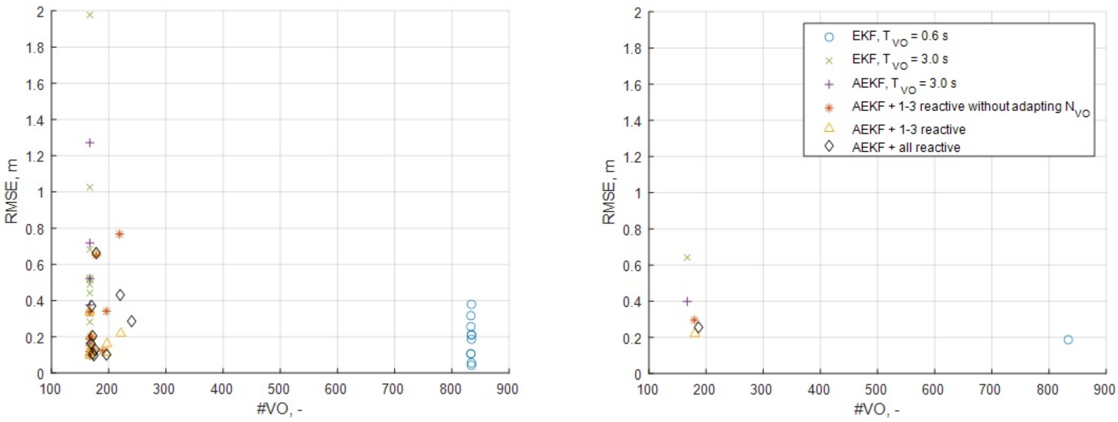


(c) AEKF with T_{VO} = 3.0 s (d) AEKF with reactive VO scheduling, strategies 1-3 without adapting N_{VO}



(e) AEKF with reactive VO scheduling, strategies 1-3 (f) AEKF with reactive VO scheduling, all strategies

Figure 3.6: Error in position estimation for various models and reactive VO scheduling strategies. Black plots are for trajectories on the tarmac and green for those on grass. Dashed lines indicate 3σ confidence margins. Both adaptive solutions at the bottom provide a smaller spread of errors compared to Fig. 3.6(b) and bound confidence margin better than Fig. 3.6(a).



(a) Individual trajectories

(b) Mean

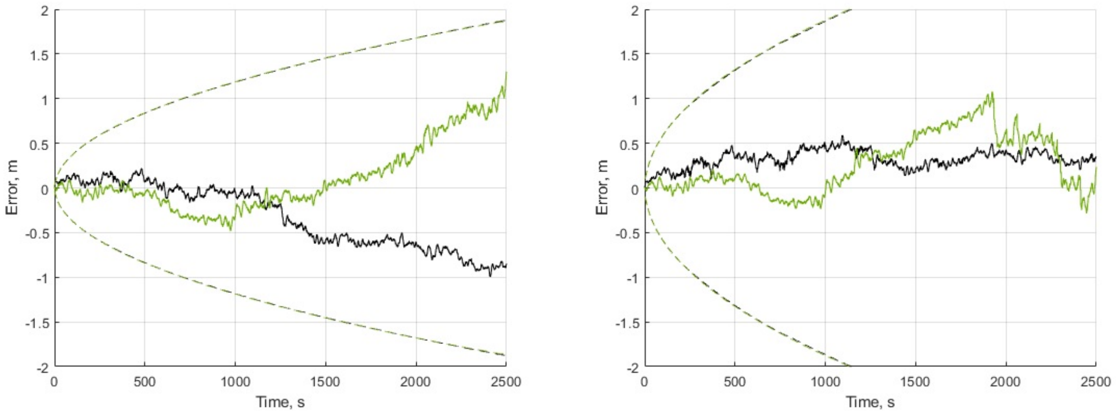
Figure 3.7: A tradeoff between RMSE and the number of VO measurements. An ideal solution would be confined to the bottom-left corner.

It is worth noting that AEKF with reactive VO scheduling provides a much better track of the position estimation variance. It still grows, albeit at a lower rate than for EKF. It can be seen as if it reaches a semi-steady state. Also, the level at which it settled is similar to EKF with T_{VO} = 0.6 s. Combining this with the fact that, on average RMSE values are comparable and the final system requires four times fewer VO measurements, the reactive VO scheduling system could contribute vastly to a reduction of computational resources.

3.3.4 Validation Using Two Long Trajectories from the Hengrove Dataset

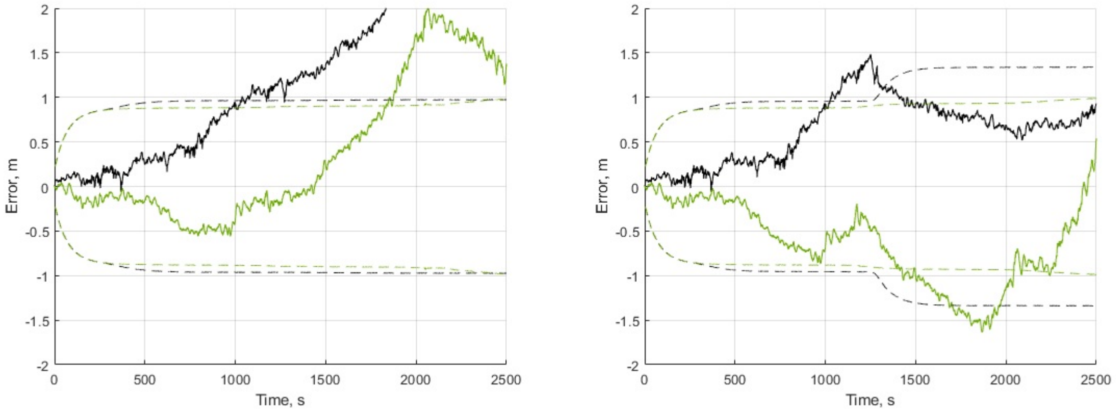
Multiple short trajectories (100 metres) were evaluated in the previous section. In addition, the same trajectories were used in simulations to identify optimal gain parameters for the adaptive filter and the ideal threshold for the fourth reactive VO scheduling strategy. When using data for tuning, it is unsurprising to see systems well-behaving when subjected to the same data. Therefore, two 500-m long trajectories are used to validate the system: one on the tarmac and one on the grass. Note that the tarmac trajectory was also used in section 2.3.2 when tuning EKF's parameters, however, the time constant T_0 had little impact, and the adaptive filter now controls σ_β. Fig. 3.8 shows errors in position estimation and Fig. 3.9 presents a tradeoff between RMSE and the number of VO measurements for each model and each trajectory. Rather than showing EKF with T_{VO} = 3 s, a 2 s period was selected as it offers a surprisingly low RMSE that is worth highlighting.

First, regarding the reactive VO scheduling, the fourth strategy improves the performance considerably. Even though the errors in Fig. 3.8(d) sometimes exceed the 3σ confidence margin, their trend remains fairly flat compared to only three reactive strategies seen in Fig. 3.8(c). It is reflected in Fig. 3.9, where the model with all reactive strategies (black diamonds) achieves lower



(a) EKF with T_{VO} = 0.6 s

(b) EKF with T_{VO} = 2.0 s



(c) AEKF with reactive VO scheduling, strategies 1-3

(d) AEKF with reactive VO scheduling, all strategies

Figure 3.8: Errors in position estimation for two long trajectories validating reactive VO scheduling algorithm using the Hengrove dataset. Black plots are for the trajectory on the tarmac and green is for the one on the grass. Dashed lines indicate 3σ confidence margins. Note how the AEKF with all trigger strategies (the last plot) provides good position estimation with errors maintained within 3σ confidence margin for most of the time. The performance EKF with T_{VO} = 2.0 s is surprisingly good, but do note a sharp change in the green trajectory. It is believed to be an error in VO which, in this case, helped the position estimation.

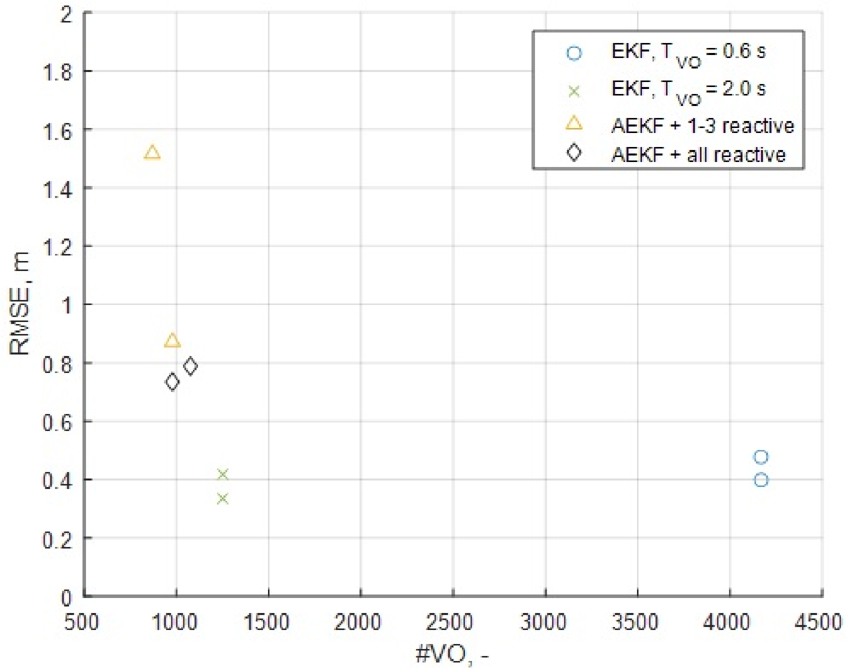


Figure 3.9: A tradeoff between RMSE and the number of VO measurements based on two long trajectories from the Hengrove dataset. An ideal solution would be confined to the bottom-left corner. Note how the AEKF with all triggering strategies implemented (black diamond) trades higher RMSE for a lower number of VO measurements compared to EKF with T_{VO} = 2 s (green x's). It would be up to the mission requirements to decide which model is more favourable, but do note the caveats mentioned regarding the EKF in Fig 3.8(b).

Table 3.2: Total execution time for each model

model	total execution time [s]
EKF, T_{VO} = 0.6 s	140.8812
EKF, T_{VO} = 2.0 s	115.0567
AEKF + 1-3 reactive	131.0465
AEKF + all reactive	125.2199

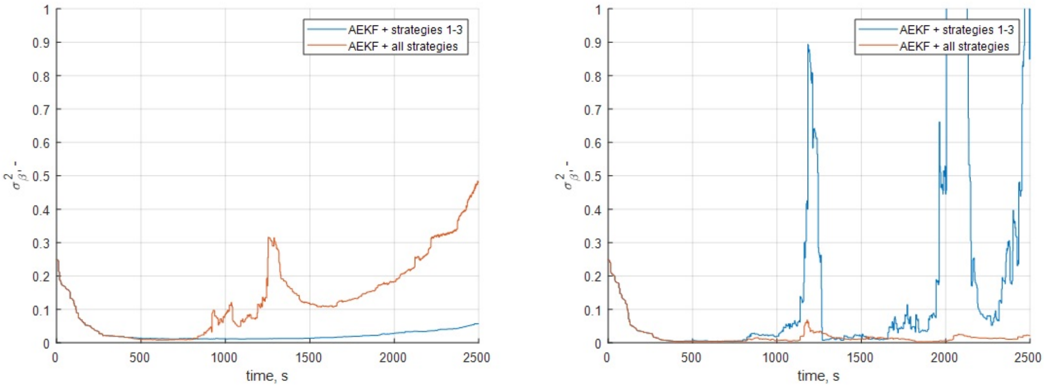
RMSE at the cost of a few more VO measurements compared to only three strategies (yellow triangles). When AEKF with reactive VO scheduling is compared to EKF, the latter performs very well and has smaller RMSE. In addition, EKF with a lower VO framerate provides even better performance (green crosses in Fig. 3.9). However, note how the grass trajectory for this model (green plot in Fig. 3.8(b)) has a sharp change around 2000^{th} second. The error plot had a growing trend but was suddenly stopped. It does not seem to be an effect of Gaussian noises but rather a possible error in VO measurement, which happened to improve performance in this case. Furthermore, the 3σ confidence margin for EKF continues to grow in both Fig. 3.8(a) and Fig. 3.8(b), whereas for AEKF with reactive VO scheduling it remains fairly stable.

Table 3.2 lists the total time of execution for all models. This time excludes any VO processing. It captures only the time to perform sensor fusion. Note that AEKF with only reactive strategies 1-3 takes more than the other model, even when performing fewer corrections with VO measurements. It is assumed that the time comes from the inversion of matrices during the Q-adaptation. As the errors grow for this model, there might be situations where numerical computations take longer than usual to perform. What is interesting, however, is that the AEKF with reactive VO scheduling may not offer much gain compared to EKF with $T_{VO} = 2$ s. EKF requires, on average, 225 more VO measurements. The algorithm takes about 0.1 s on a modern machine to execute (all models were also executed on a modern machine), which, if VO processing were taken into account, would extend the total time by an additional 22.5 s. It would take only 12 s more to execute than AEKF with all reactive VO scheduling strategies. From the system perspective, it might be debatable if this additional processing time outweighs the complexity that comes with the proposed model. The main advantage of the proposed solution is the ability to track the variance, albeit at a slightly worse quality of position estimation. On the positive side, there is a huge amount of time that can be saved when compared to EKF with $T_{VO} = 0.6$ s, the equivalent of what can be expected on ExoMars and SFR.

Finally, it is worth discussing why this additional trigger strategy made such a difference for two long trajectories, whereas there were no noticeable differences in the position estimation when analysing multiple short trajectories. To illustrate this behaviour better, σ_β^2 for both models and both trajectories is presented in Fig. 3.10. For both trajectories, both models adapt σ_β^2 similarly, which is reflected in a comparable λ estimate, to around 500^{th} second. This timestamp marks the end of all short trajectories, which can indicate why their results were nearly identical in the previous section. After that, covariances and wheel slip estimations diverge. In the case of AEKF with only reactive strategies 1-3 (blue plots), it seems that on the tarmac its σ_β^2 is stabilised on a fairly constant value (Fig. 3.10(a)), which starts to grow only towards the end of the trajectory. This is reflected in a constant λ estimate in Fig. 3.10(c). When the process noise reaches a small level, the filter makes little use of measurements and trusts mainly the mathematical model. Because of that, no new information about the environment can be fused into the state leading to bigger errors in the position estimation. AEKF, with all reactive VO scheduling strategies implemented (orange plots), triggers additional measurements should σ_β^2 increase by 1.6%. It seems enough to notify the filter that there are some discrepancies between the mathematical model and the environment. As a result, λ is tracked more accurately. The grass trajectory presents the other edge case for process noise covariance. Here, because of the terrain variability, AEKF with only reactive strategies 1-3 implemented adapts the covariance to large values (Fig. 3.10(b)) making λ noisier and less precise (Fig. 3.10(d)). The additional reactive strategy allows scheduling more VO measurements when the covariance grows, thus maintaining it on a level that is not too high (noisy slip estimates) and yet not too low (disregarding information from the measurements).

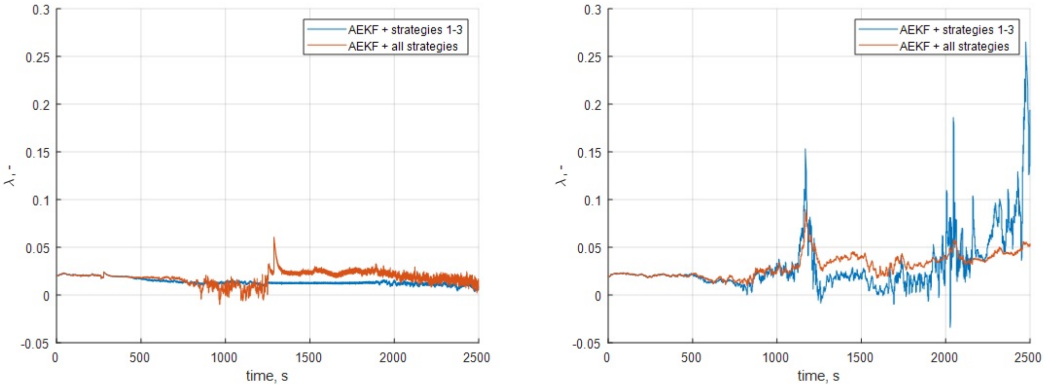
3.3.5 Validation Using One Long Trajectory from the Sand Bay Beach Dataset

Reactive VO scheduling algorithm work on two 500 m long trajectories from the Hengrove dataset. There might be still some doubts regarding the performance given that the complete analysis and tuning was done using the same dataset. This section analyses one ~500 m long trajectory from the Sand Bay Beach dataset. The additional field trials were scheduled for the purpose of chapter 4, and therefore the dataset is described in section 4.4.1. The trajectory in question is interesting for two reasons: firstly, the terrain consists of concrete pavements, loose sand, and grass; secondly,



(a) σ^2_β on Tarmac

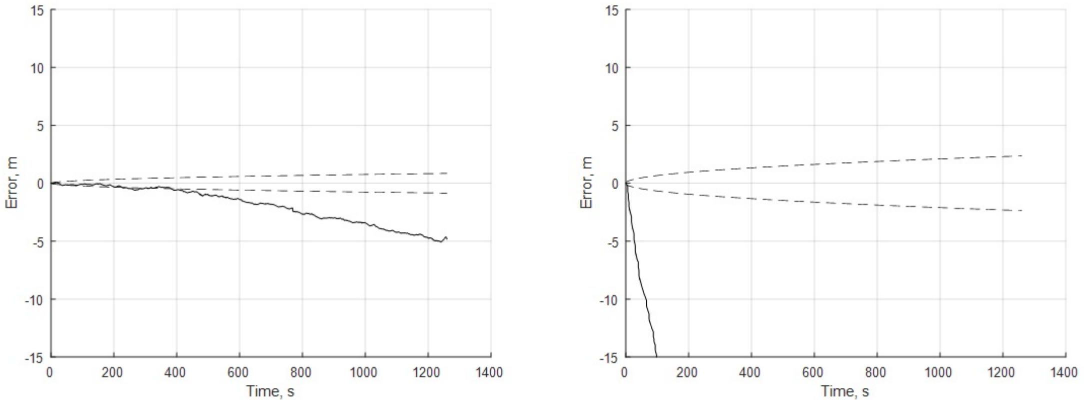
(b) σ^2_β on Grass



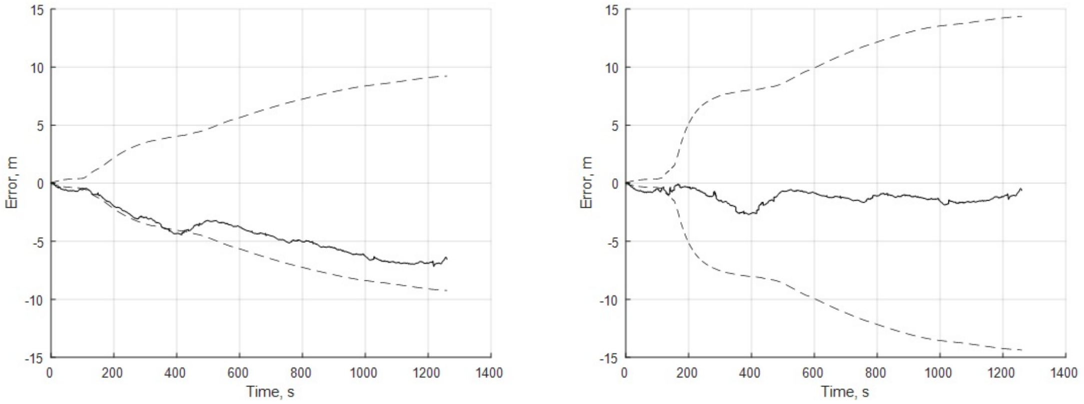
(c) λ on Tarmac

(d) λ on Grass

Figure 3.10: σ^2_β and λ for long tarmac and grass trajectories for AEKF with two reactive VO scheduling strategies approach (blue - only strategies 1-3, orange - all strategies). Note how the first approach on tarmac maintains very low σ^2_β which consequently leads to the filter preferring WO over VO (compare with chapter 2). This is reflected by the black trajectory in Fig. 3.8(c) having a growing error plot associated with WO. The trajectory on the grass, however, has much larger peaks of σ^2_β for the first approach which lacks the strategy to trigger VO when the covariance grows too much. Because of it, the second approach can maintain lower covariance and thus a less noisy λ estimate.



(a) EKF with T_{VO} = 0.2 s (b) EKF with T_{VO} = 2.0 s



(c) AEKF with reactive VO scheduling, strategies 1-3 (d) AEKF with reactive VO scheduling, all strategies

Figure 3.11: Errors in position estimation for one long trajectory validating reactive VO scheduling algorithm using the Sand Bay Beach dataset. Dashed lines indicate 3σ confidence margin. Note how even the EKF with frequent VO measurements struggle to correctly track the error (error plot exceeds 3σ confidence margin). AEKF with only strategies 1-3 implemented tracks error correctly, however, the error plot has a gradient that may eventually exceed the confidence margin. AEKF with all trigger strategies implemented offers a stable error plot around zero at the cost of a bigger confidence margin. Note though that from the perspective of driving approx. 500 m, this confidence margin is still small.

the rover was driving 0.4 m/s, which is double the speed used in the whole Hengrove dataset. No parameters are adjusted, they are all unchanged. The only variation is setting $T_{VO_{MAX}}$ to 2 s. It was shown that VO performance greatly deteriorates when estimating every 3 s with a rover speed of 0.2 m/s. This translates to a 0.6 m distance between two consecutive frames. For a representative comparison, VO should trigger twice as fast when driving at double speed. However, to make it even more challenging, $T_{VO_{MAX}}$ is set to 2 s, giving a 0.8 m distance between the measurements. Errors in position estimation are presented in Fig. 3.11.

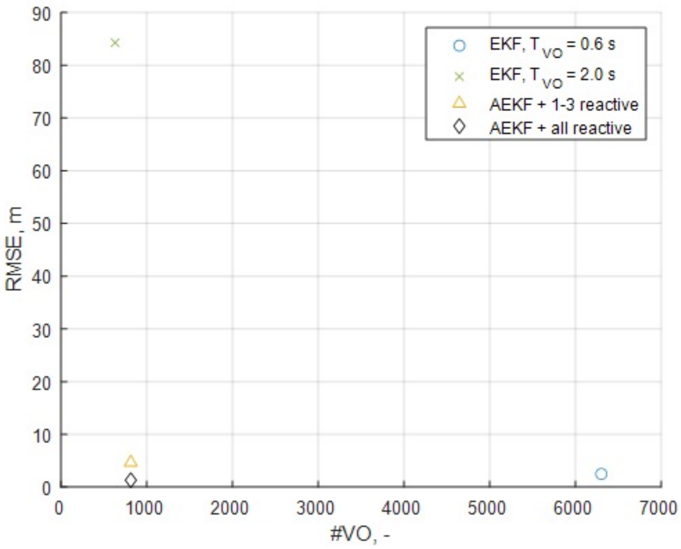


Figure 3.12: A tradeoff between RMSE and the number of VO measurements based on one long trajectory from the Sand Bay Beach dataset (double the nominal rover speed). An ideal solution would be confined to the bottom-left corner.

What becomes apparent from Fig. 3.11 is that when driving at double the speed, EKF can only provide relative good position estimation when VO is providing measurements at a fast rate (Fig. 3.11(a)). The error of -5 m is only 1% of the total distance travelled, which is in line with expectations regarding the VO performance [33]. The variance, however, is not well-tracked, and the filter is overconfident with its estimates. Reducing the number of VO measurements for EKF to one every two seconds (Fig. 3.11(b)) has an expected behaviour in worsening the quality of position estimation to an unusable level. Introducing AEKF with reactive VO scheduling but only with strategies 1-3 implemented (Fig. 3.11(c)) has a noticeable better accuracy. Nevertheless, the error continues to grow. Only when all four strategies are implemented (Fig. 3.11(d)) does the filter offer accurate tracking of the position without any trend in error. It has a relatively large 3σ confidence margin, however, 15 m indicates only 3% of the total distance travelled, which, depending on mission requirements, may not be that bad. Especially when compared to faster travel time than nominal and far fewer VO measurements required than typically. It opens an exciting option for mission planning where a rover first covers a considerable distance fast and starts to perform more accurate and more detailed localisation to reduce the localisation covariance at the end of the trajectory.

Furthermore, as presented in Fig. 3.12, the final solution's RMSE (black diamond) is lower by half compared to EKF with T_{VO} = 0.2 s (blue circle) while performing 7.75 times fewer VO measurements. The final solution offers lower RMSE than AEKF with only 1-3 trigger strategies (yellow triangle). It is interesting to inspect based on which trigger strategies these solutions scheduled VO measurements. To illustrate that better, Fig. 3.13 provides stacked bar charts

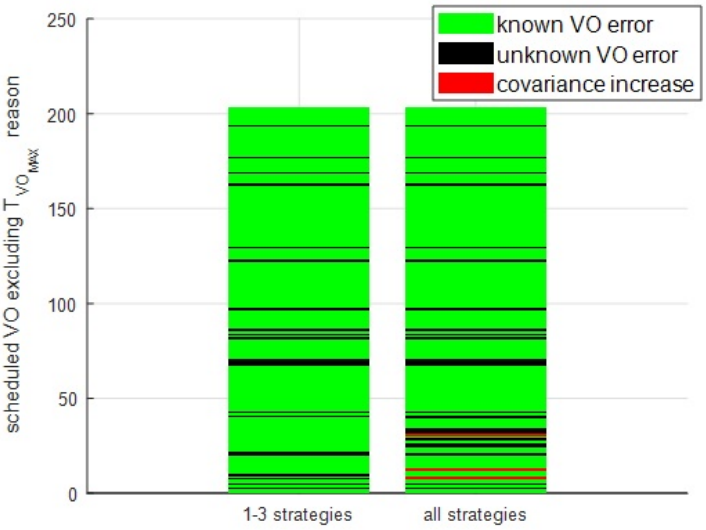


Figure 3.13: Scheduled VO measurements for AEKF with 1-3 trigger strategies and all strategies. Strategy 3, the maximum time from the last VO measurement $T_{VO_{MAX}}$, was removed for better clarity. Note how the last strategy, implemented only for the second approach, was triggered only three times (red stripes). At the same time, note how most of the strategies were the same from about 50^{th} VO measurement onward (up). Now investigate Fig. 3.11(c) and Fig. 3.11(d). Their shapes are similar, yet three additional measurements in the second approach offered a big improvement in position estimation.

with strategies that were selected to schedule VO measurement. Strategy 3, the maximum time between VO measurements, was removed from the plot to make it clearer. Notice that both models used the same number of VO measurements. The model that implemented all four strategies had only four VO measurements related to the σ_β^2 strategy (red stripes). After about 40 scheduled VO measurements, both models used the same strategies. This is reflected in a similar shape of plot in Fig. 3.11(c) and Fig. 3.11(d). It shows that only a few well-placed measurements can drastically improve position estimation.

3.4 Summary

This chapter presents a new approach for dealing with wheel slip which reduces the number of VO measurements while maintaining navigational accuracy. It may save resources when driving on stable ground (no need for frequent measurements), whereas additional VO can be scheduled on more dynamic terrain to maintain precise position estimation. An adaptive filter plays a crucial role in the proposed solution as it allows for adjusting noise statistics to better represent the environment's variability. In addition, the adaptive element enables better tracking of position estimation variance. Interestingly, when supplied with rare VO updates, the adaptive filter offered a better quality of position estimation when assuming non zero biases in observation noises. On the other hand, they negatively impacted the system when VO was performed frequently. It may raise a question regarding the observability of the system, hence, it is advised to repeat trials with additional sensors (for example, another stereo pair pointing in a different direction).

The main noises that substantially impact system performance are the VO measurement noise and the process noise driving wheel slip estimation. Consequently, they are vital in defining trigger strategies for an algorithm to decide when to schedule a new VO measurement. It has been shown that even though triggering new VO measurements based on the increase in process noise statistics does not happen often, it may significantly impact the quality of the position estimation. The proposed solution was tuned using several short trajectories and validated on three long trajectories from two different datasets. What is more, the robot was travelling at double the speed in one trajectory. The solution offered a comparable accuracy to EKF, where the latter used 4-7 times more VO measurements

Finally, only a system with one degree of freedom was here analysed. Findings may indicate what to expect when analysing higher dimensions, however, the same performance is not guaranteed. This problem is addressed, to some extent, in chapter 5.

CHAPTER 4

WHEEL SLIP MEASUREMENTS AND PREDICTION

Previous chapters established a one-dimensional model for improved localisation with integrated wheel slip estimation. With the addition of adaptive filtering and reactive VO scheduling, the system can optimise the number of required VO measurements to maintain good localisation performance. This chapter adds another option of potentially reducing the number of VO measurements further by discussing and analysing various ways how predicted wheel slip can improve localisation accuracy. With the perfect knowledge of wheel slip, WO can be directly used without the need for any VO measurements.

This chapter, based on [52], investigates how the wheel slip predictions derived from, for example, forward-faced vision or drive current can be fused with WO and VO. The solution is based on the model with Integrated Slip Estimation defined in section 2.1.3, using either point slip measurements, a slip prediction model, or a slip profile within the state. The investigation also compares different VO measurement periods, studying how VO effort trades with accuracy for each slip prediction scheme. The solutions provide improved localisation accuracy and hint at the intriguing possibility of slip-based SLAM for future investigation.

4.1 Introduction

As opposed to estimating slip while the robot drives, some research analyses the terrain in front of the platform using images [29, 73–76]. Such information feeds nicely into a path-planning function which may plot a path that avoids high-slip areas that may lead to increased power usage or potentially damage the rover wheels. The latter can be solved by applying algorithms that adapt wheels' velocity [56] to reduce the risk.

This chapter explores methods of fusing slip predictions and measurements into the Extended Kalman Filter (EKF) to improve the rover's localisation performance. The work is similar to [55], where authors combine vision-based information with wheel slip to improve the slip prediction, but not the robot's localisation. The work is also related to [77], where a system combines vision-based information with the torque profile of a hybrid legged-wheeled robot to improve the localisation, but without the integrated filter. WO is correlated with vision information in that work because both are used to train the Support Vector Machine (SVM) classifier, whereas, in this project, any correlation is avoided. Furthermore, in the case of this project, only wheel slip is analysed, which is a more general problem for wheeled or tracked platforms, whereas [77] seems to apply only to one class of robots.

This work extends what was presented in chapter 2. Similarly to previous chapters, all mathematical models, experimentation, and evaluation is limited to one dimension for the initial exploration of the concept.

4.2 Models Definition

4.2.1 Model 0 - The Base Model

This is the model with Integrated Slip Estimation defined in the section 2.1.3. In this model, a new estimate of wheel slip is an interpolation between its previous estimate and the process noise, as defined in (2.8). It describes how the terrain changes in the time domain. For the purpose of this chapter, Model 0 without the low-pass filter that drives wheel slip is referred to as the Core Model.

4.2.2 Model 1 - Direct Slip Measurement

The filter can be readily modified to include a point measurement $z_\lambda(t)$ of the slip experienced at time t, deduced either from wheel current as in [78], from a downward-facing camera or by interpolating over a profile estimated using a forward-facing camera. The block diagram of this model is presented in Fig. 4.1. All equations for this model (prediction, correction) are the same as for Model 0 with additional observation model z_λ. Using wheel current for point slip measurements would need to be done with care as one would need first, most likely, to perform a calibration to understand how much power the wheel drain when driving on various terrain types and at various tilt angles. Furthermore, a control system may have an impact on the wheel speed and its power consumption. For example, when there is a slippage and a wheel suddenly starts driving much faster (no load), an internal PID controller can reduce its speed, thus reducing power consumption and invalidating current-based wheel slip estimation.

4.2.3 Model 2 - Known Slip Profile

A more complicated way of incorporating a predicted slip *profile* is to incorporate it into the prediction model within the filter. For example, an expert system [12] can analyse HiRISE

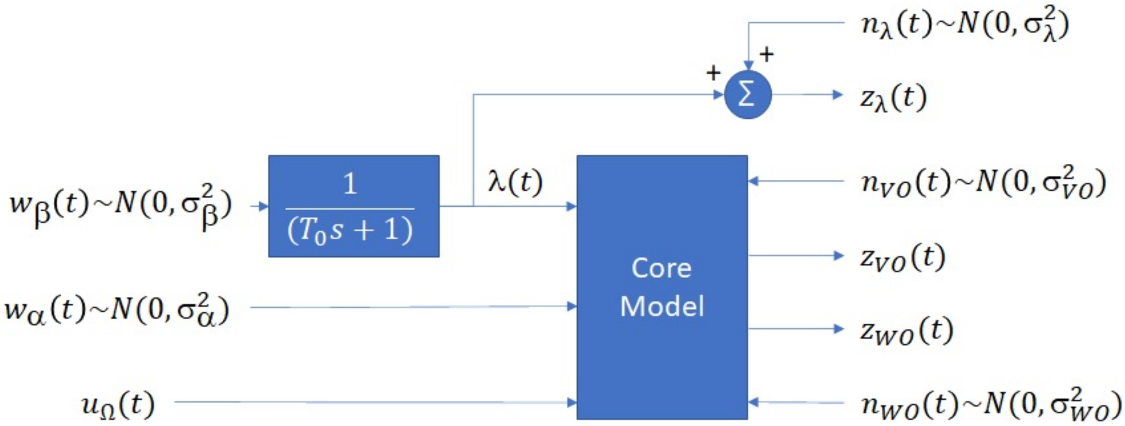


Figure 4.1: The block diagram of Model 1, which provides a direct slip measurement. The Core Model is the model with Integrated Slip Estimation from chapter 2 except for the low-pass filter being outside of it. This approach allows for a better illustration of how different models proposed in this chapter impact $\lambda(t)$.

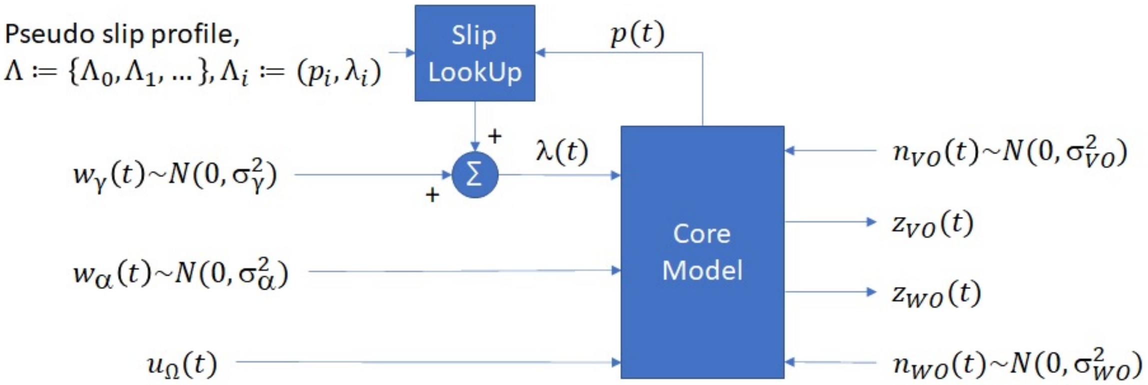


Figure 4.2: The block diagram of Model 2 with known slip profile. In this model, wheel slip, $\lambda(t)$, rather than being directly measured, is estimated by a look-up operation based on the input slip profile, Λ, and estimated robot position, $p(t)$. To capture errors in the pseudo slip profile, a process noise $w_\gamma(t)$ is introduced.

images of Mars and translate them into a low-fidelity slip assessment. Then, as the robot drives, it compares its position against the known profile and uses the correct wheel slip prediction. Alternatively, the profile could be analysed using a forward-facing camera. What is distinguished in this model is that the pseudo slip profile exists outside the Kalman Filter. As a result, there is no correlation between the position estimation and the profile. However, because the filter's state does not include the profile, all filter computations are faster. The block diagram of this model is presented in Fig. 4.2. As the wheel slip is now estimated based on the profile, to model the uncertainty of the profile, a process noise w_γ is introduced which replaces the w_β term related to the more straightforward low-pass slip estimate used in Model 0.

The look-up function is a simple interpolation between two nearest values. Together with the process noise, a new equation for slip prediction, in its discrete form, is given by

$$\tilde{\lambda}(k+1) = \frac{\tilde{p}(k+1)-p_i}{p_{i+1}-p_i}\lambda_{i+1} + \frac{p_{i+1}-\tilde{p}(k+1)}{p_{i+1}-p_i}\lambda_i + w_\gamma(k), \quad \tilde{p}(k+1) \in < p_i, p_{i+1}) \tag{4.1}$$

Notice the similarities between (4.1) and λ in (2.8). In both cases, the predicted slip at $k+1$ is an interpolation between two values. In (2.8), this interpolation is made in the time domain, whereas in (4.1), it is done in the spatial domain. The look-up function seems to be an elegant way of replacing the low-pass filter with the slip profile.

4.2.4 Model 3A - Predicting Slip Profile with Linear Observation Model

The natural culmination of slip prediction, with the highest fidelity but also the most complicated filter, includes the complete slip *profile* as part of the state. Two different ways of parameterising the profile are explored, yielding Models 3A and 3B, respectively. The block diagram illustrating this model is presented in Fig. 4.3. Note that the complete treatment of slip as a Gaussian process [74, 75] is left for future investigation: simple 1-D interpolation is employed in this study.

Similarly to Model 2, this model employs a slip profile, but rather than using it as pseudo-control input, the profile is predicted via z_{λ_i} measurements. As such, the previously defined process noise w_γ can be omitted and (4.1) is reduced to a simple interpolation. Note that the linear observation model implies slip measurements at the exact locations of the profile points stored in the state vector, as

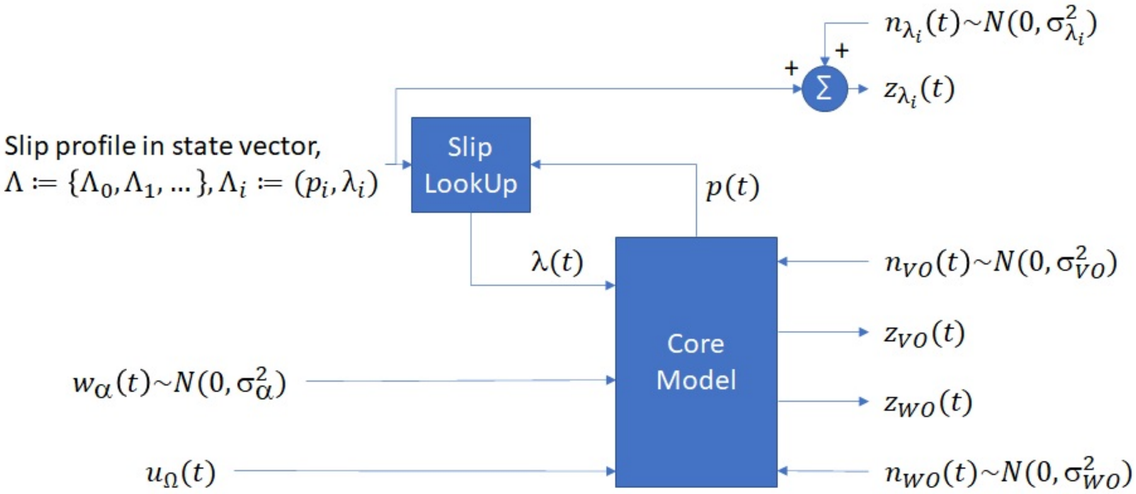


Figure 4.3: The block diagram of Model 3A with slip profile prediction. In this model, the slip profile, Λ, is moved to the Kalman filter state and a linear observation model is employed to estimate it. The observation model is linear because z_{λ_i} measures the exact points of the profile.

$$z_{\lambda_i} = \lambda_i + n_{\lambda_i}, \quad p_i = \tilde{p}(k+1) + i, \quad i = l_{MIN}, l_{MIN} + step, l_{MIN} + 2 * step, ..., l_{MAX} \tag{4.2}$$

Equation (4.2) defines that slip observation is made for all profile points between l_{MIN} and l_{MAX} in front of the predicted position. It is assumed that the vision system that performs slip prediction is oriented forward. For this work, l_{MIN} is defined as 1 m and l_{MAX} as 4 m. The number of slip profile points stored in the EKF state impacts computational performance and the correct slip estimation. As such, slip profile points were arbitrarily decided to be stored at 1 m intervals.

In this work, slip prediction is simulated, even when operating a rover. However, one can expect an expert system which divides an image into patches that correspond to areas in front of the robot. Each patch is then assigned with a predicted position and associated wheel slip value that could be used by this model. Once the robot is driving over this area, wheel slip would be predicted by interpolating estimated slip values for the two nearest patches (Λ profile points).

4.2.5 Model 3B - Predicting Slip Profile with Non-Linear Observation Model

As indicated, Model 3A is limited to a linear observation model. From an implementation perspective, storing an extensive profile in the state vector may be impractical, as it would result in a massive covariance matrix that needs to be inversed once multiplied by the observation model Jacobian. On the other hand, few profile points may provide limited information about the slip. It is possible to introduce a non-linear observation model that allows sampling multiple points around the profile point to solve this problem. When deciding on the sampling density/rate, it should be considered that too many sample points may lead to overfitting the observation model.

Fig. 4.4 presents the block diagram of Model 3B. The observation function is similar to (4.2) with interpolation from (4.1).

$$\begin{gathered} z_{\lambda_i} = \frac{p_i - p_j}{p_{j+1} - p_j}\lambda_{j+1} + \frac{p_{j+1} - p_i}{p_{j+1} - p_j}\lambda_j + n_{\lambda_i}, \\ \forall p_i \in \langle \tilde{p}(k+1) + l_{MIN}, \tilde{p}(k+1) + l_{MAX}\big) \wedge p_i \in < p_j, p_{j+1}) \end{gathered} \tag{4.3}$$

Equation (4.3) may seem quite different from (4.2), but conceptually it is very similar. Several slip measurements z_{λ_i} are performed at p_i locations between l_{MIN} and l_{MAX} in front of the predicted position. Each measurement then contributes to the two nearest slip profile points λ_j and λ_{j+1}. For the experiments, we decided to perform slip measurements at 0.5 m intervals between l_{MIN} and l_{MAX}. Note that observation non-linearity arises because these intervals are not pre-defined but are associated with the predicted position. Similarly to Model 3A, we decided to store slip profile points in the EKF state at 1 m intervals.

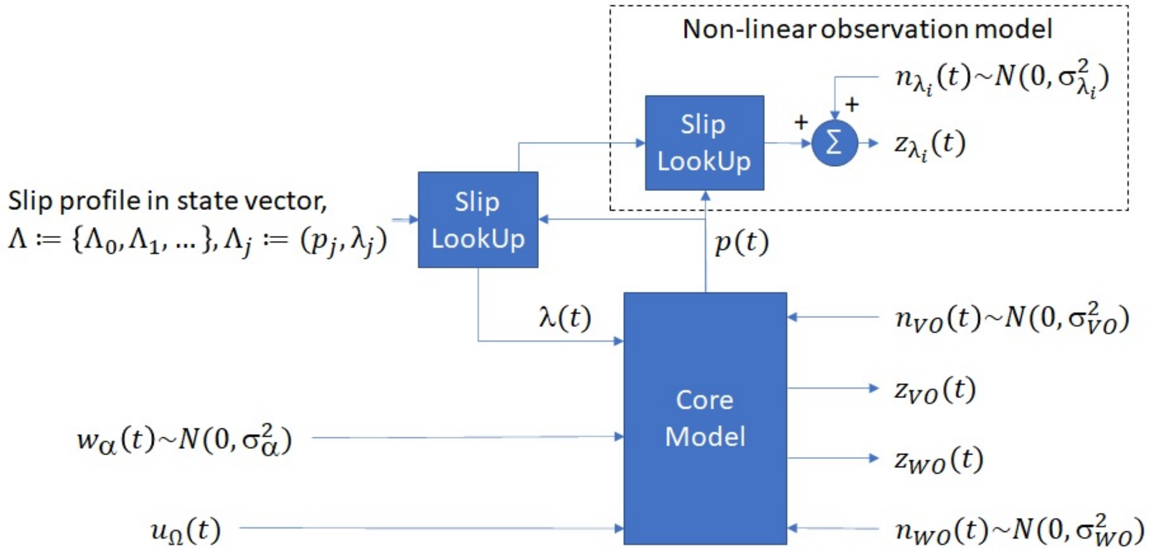


Figure 4.4: The block diagram of Model 3B with slip profile prediction and non-linear observation model. This model is very similar to Model 3A. The only difference is in the non-linear observation model which provides estimates of points that do not directly map to the slip profile Λ. Because of that, another look-up/interpolation function is required to map all measurements to the nearest profile point. Note different subscriptions in slip profile and profile measurements so to avoid confusion.

4.3 Simulations

As before, the first step was to perform low-fidelity simulations to determine the behaviour of each model under various circumstances. Similarly to what was presented in section 2.2, three 100 m-long test cases were defined. In each test case, the control velocity was 0.4 m/s to reflect the rover's speed from the field trials described in the next section. WO estimated every 0.1 s, the time constant for Model 0 and Model 1 was set to 20 s, and the simulated wheel slip measurement was performed every 5 s. This value was selected arbitrarily, assuming that terrain classification might be quite compute-intense and would not happen too often. At the same time, the aim was to achieve an overlap in slip measurements for Models 3A and 3B. For process noise statistics, σ_α was set to 1.5, σ_β to 0.5, and σ_γ to 0.05. Each test case was repeated with a different VO update period to investigate its impact on each model. T_{VO} used was zero (no VO measurements), one, and three seconds. VO every three seconds at 0.4 m/s may provide largely erroneous measurements, thus, it shows how each model copes with them.

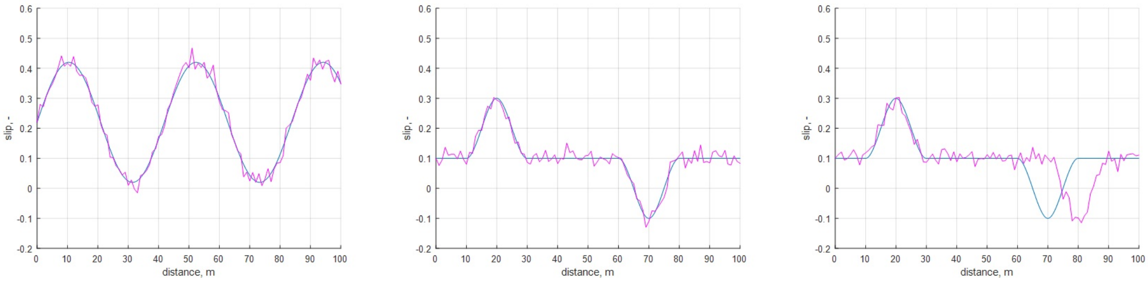
The first test case simulates a sinusoidal wheel slip. It investigates how well each model tracks the continuous change in the slip. The second test case is more realistic, where there are only two bumps in the wheel slip, however, one is negative to highlight some interesting properties of the models. It can be understood as driving through two patches of different terrain. The last test case is almost identical to the previous. The only difference is that part of the model's erroneous slip profile is shifted. It represents a different type of error, where the second terrain patch was incorrectly located. Fig. 4.5 shows slip profiles for each test case. This will help highlight the different capabilities of the three models. Blue lines represent the ground truth slip, which is used to simulate the measurements with added random noise. The magenta lines are erroneous slip profiles (different noise statistics compared to simulated measurements) used for look-up operation by Model 2 and as the initial state in Models 3A and 3B.

The metrics, as defined in section 1.6 for each model and each test case are presented in Fig. 4.6, where blue bars are for Model 0 (EKF), red - Model 1, yellow - Model 2, purple - Model 3A, and green - Model 3B. EKF had a much larger RMSE, and its error exceeded the confidence margin in a far greater way than illustrated on the relevant plots, but these were capped to allow better readability of other data. Fig. 4.7 presents an example of errors in position estimation with 3σ confidence margin for each model and each test case with $T_{VO} = 1$ s.

There are several observations based on the results provided in Fig. 4.6:

- Model 0, the EKF with Integrated Slip Estimation as defined in chapter 2, does not provide good performance in any of the test cases. Admittedly, they all were quite challenging because the ground truth wheel slip changed significantly. Compare that with results presented in Fig. 2.21(b) where the wheel slip oscillated around 0.02 with one spike up to 0.1 (error in VO). Furthermore, without the adaptive filter, wheel slip λ in between VO measurements is being forgotten to zero (mean value of the w_β process noise). Challenging these test cases as they may be, any form of slip prediction improves position estimation accuracy considerably.

- From Fig. 4.6(a) top row: models 2, 3A, and 3B improve accuracy compared to baseline with no VO. Moving along the top row, introducing VO measurements improves the accuracy even further with the expected behaviour of the more frequent VO, the lower RMSE.

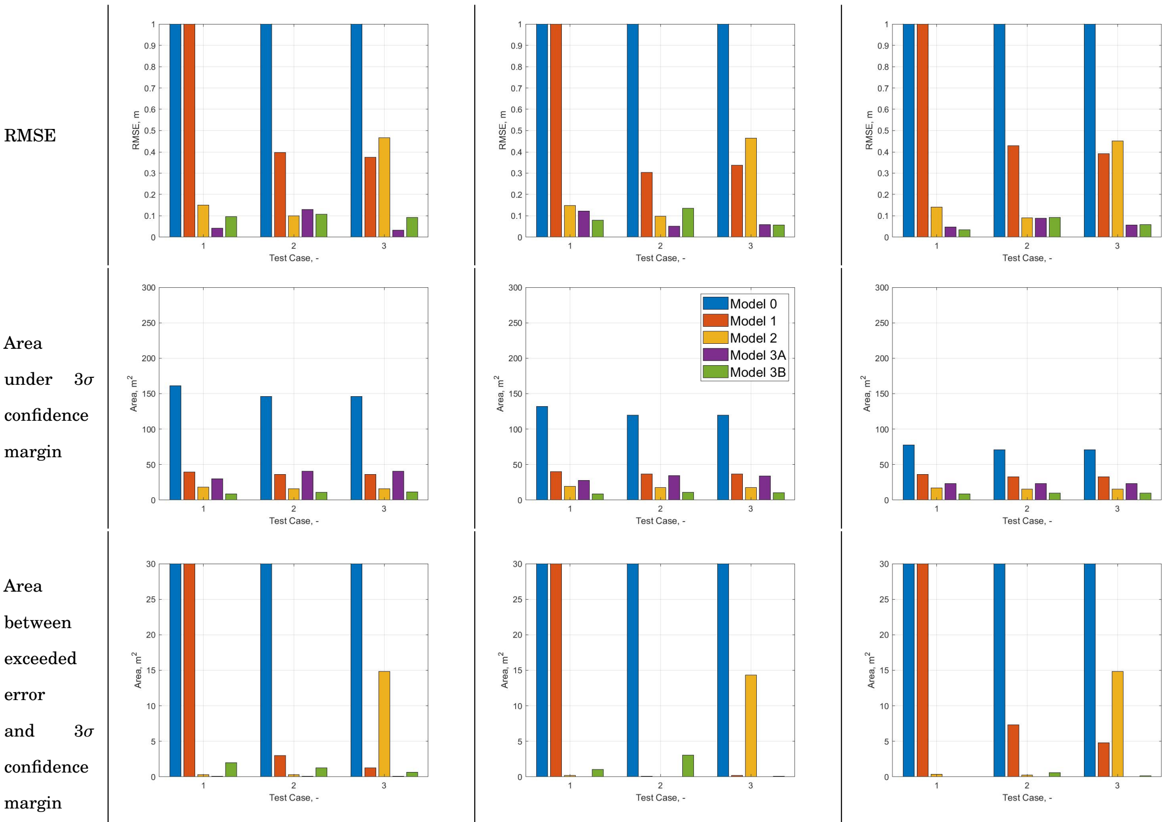


(a) Test Case 1: Sinusoidal Slip (b) Test Case 2: Bumps (c) Test Case 3: Bump Offset

Figure 4.5: Three test cases for models verification. Solid blue lines are the ground truth wheel slip, whereas solid magenta lines represent profiles with an added error used as a source of measurements for Model 2 and the initial state for Model 3A and 3B.

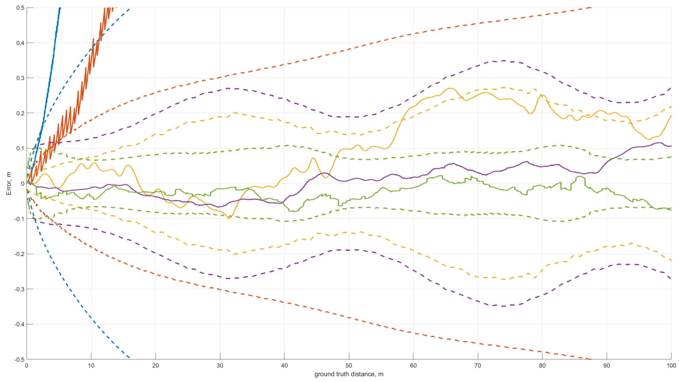
- Model 1 seems the least effective of all proposed models. Direct slip measurement makes it overconfident. Note how the area under the 3σ confidence margin is small (middle plots), yet the RMSE of the position estimation is large (top plots), leading to the error growing beyond the confidence margin (bottom plots). What drives this large RMSE is that wheel slip measurement is done only every 5 s. Similarly to Model 0, because of the low-pass filter, λ decreases to the mean value of the process noise w_β, which is zero without the adaptive filter. A longer time constant T_0 could be used to mitigate this problem. Alternatively, a different method for direct slip measurement could enable a faster acquisition, for example, based on motors current or torque.

- Model 2 can provide accurate position estimation as long as the input profile is correct, i.e., in test cases 1 and 2. The third test case indicates Model 2's main weakness, as it cannot refine the corrupted profile. It is best illustrated in Fig. 4.7(c) where there is a noticeable

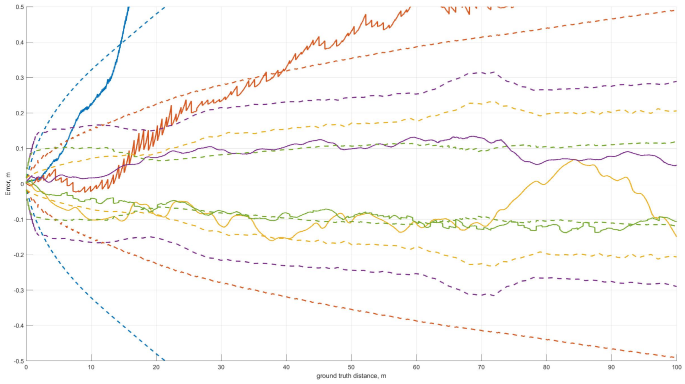


(a) $T_{VO} = \infty$ (no VO) (b) $T_{VO} = 3$ s (rare VO) (c) $T_{VO} = 1$ s (frequent VO)

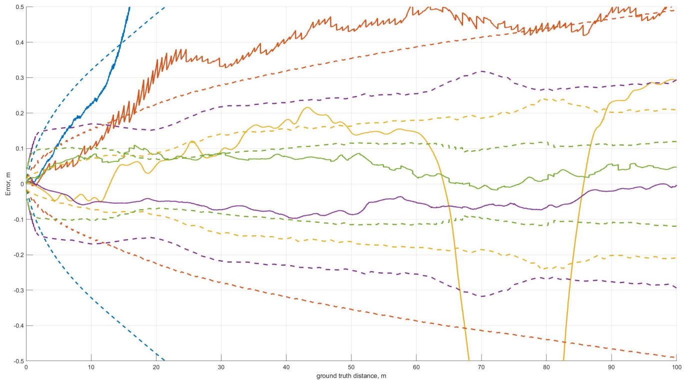
Figure 4.6: Summary of metrics for errors in position estimation for each model and each test case. The number of test cases is labelled on each plot separately (1, 2, and 3). Vertical axis limits truncate some bars. Note the legend in the middle plot. In general, models which utilise slip profiles offer smaller RMSE. The exception is the Model 2 with input pseudo slip profile in the last use case, where an error was introduced to the profile. In this case, the model cannot rectify the slip profile based on measurements resulting in wrong information being used.



(a) Test Case 1: Sinusoidal Slip



(b) Test Case 2: Bumps



(c) Test Case 3: Bump Offset

Figure 4.7: Errors in position estimation (solid lines) with 3σ confidence margins (dashed lines) for the second and third test case with T_{VO} = 1 s. Blue colour represents Model 0, red - Model 1, yellow - Model 2, purple - Model 3A, and green - Model 3B. Note how models utilising slip profiles provide smaller errors with an exception for Model 2 in the last use case.

change in the error (solid yellow line). However, Model 2 offers good position estimation accuracy and its associated variance tracking for all other test cases.

- Both Model 3 variants A and B provide outstanding accuracy in position estimation regardless of VO frequency. The version with the non-linear observation function (Model 3B) offers a narrower confidence margin which may lead to error exceeding it, but, overall, both methods are comparable.
- Model 3B seems to have a unique property when it comes to how it tracks the variance in position estimation. First, notice in Fig. 4.7 how positive confidence margins for Model 2 (dashed yellow line) and Model 3A (dashed purple line) have shape similar to the opposite of the ground truth slip profile. It is because the predicted slip $\tilde{\lambda}(k+1)$ in (4.1) is correlated with the predicted position $\tilde{p}(k+1)$. From (4.1), partial derivative for predicted slip is defined as

  $$\frac{\partial\tilde{\lambda}(k+1)}{\partial x} = \frac{\lambda_{i+1}-\lambda_i}{p_{i+1}-p_i}\frac{\partial\tilde{p}(k+1)}{\partial x} \tag{4.4}$$

  From (2.9), $\frac{\partial\tilde{p}(k+1)}{\partial x}$ has the $\big(1-\lambda(k)\big)$ element which dictates, in Model 2 and Model 3A, the shape of the position estimation variance (opposite profile). Should the $\tilde{p}(k+1)$ in (4.1) was treated as a pseudo-control input (i.e. (4.4) would be zero), the variance in position estimation would follow a similar shape to Model 0 with a similar magnitude. But this is not the case for Model 3B. Notice in Fig. 4.7(a) and Fig. 4.7(b) that the confidence margin remains relatively flat, and in Fig. 4.7(c) it decreases, whereas the $\big(1-\lambda(k)\big)$ would require it to increase (compare with dashed yellow and purple lines). It is a thrilling observation as it may indicate that the model has recognised the pattern in slip profile. It is more prominent in Test Case 3 when the initially profile was not correct. This behaviour resembles SLAM and is further discussed in section 4.4.3.2.

Overall, based on the results from the simulations, Models 3A and 3B seem to be the most accurate, however, they also require more computational resources due to the included profile in the filter's state. For comparison, the state for Models 3A and 3B have 106 elements (one additional state for every metre) compared to only five for all other models. Model 2 may offer an acceptable trade-off in position estimation accuracy as long as the known profile is correct, which might not always be guaranteed.

4.4 Experimental Results

4.4.1 Trial Setup

Because trajectories from the Hengrove Park did not offer enough terrain variability to verify wheel slip prediction models, additional field trials were conducted in Sand Bay Beach, Somerset, UK. The new venue offered long, reasonably straight traverses with a mix of sand, concrete, and grass. The trial setup was similar to what was described in section 2.3.1.1. The only differences were: i) the RTK corrections were received from the Topnet Live service, and ii) the nominal rover speed was increased to 0.4 m/s. Maps with recorded ground truth are presented in Fig. 4.8. The path between points A and B was used to verify slip prediction models. The path between B and C was used to validate reactive VO scheduling as mentioned in the section 3.3.5. Note the spikes in the ground-truth in Fig. 4.8(b) which are the result of adjusting the rover manually (operator occluding the GPS antennas).

The evaluated trajectory consisted of a 4 m distance on concrete, 21 m on the sand, and 3 m back on concrete (the path between points A and B in Fig. 4.8). The rover on the test path is depicted in Fig. 4.9. Example slip profiles calculated using raw telemetry and averaged using a

(a) An overview of the whole field trials area

(b) Zoomed in region with path used for slip prediction

Figure 4.8: Recorded ground-truth from the Sand Bay Beach trials. Maps are taken from the OxTS NAVsolve tool.

Figure 4.9: Mobile platform used during the experiments driving on concrete for 4 m, then on the sand for 21 m, and again on concrete for 3 m.

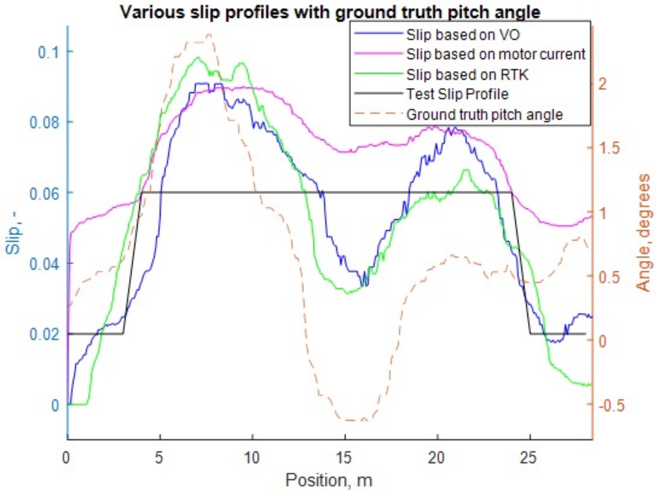
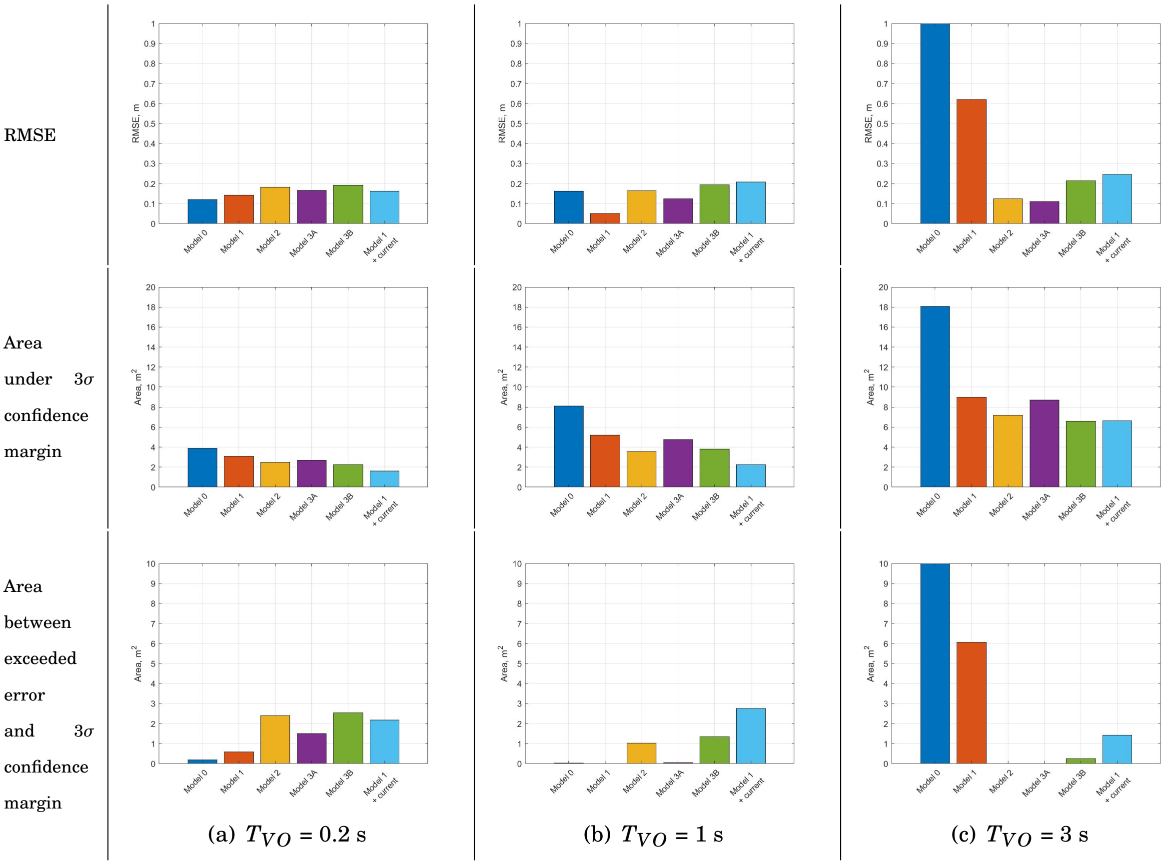


Figure 4.10: Various slip profiles: blue – as estimated by VO with WO, green – as estimated by RTK with WO, magenta – as estimated by motor current, black – test profile for vision-based mock-up, dashed – ground-truth pitch angle to indicate the change in terrain which may have impacted the slip value while on the sand (for visualisation and a better understanding of the terrain only).

median filter are presented in Fig. 4.10. The green line shows the ground truth profile measured using RTK and WO. The blue line shows the slip profile measured using VO and WO, where VO was estimated every 0.2 s (i.e. the distance between consecutive images was approximately 0.08 m). The magenta profile is achieved using motor current measurements and is discussed in the section 4.4.3. The black line is a simulated terrain classification profile: because the main focus of this work is on sensor fusion and position estimation, the classifier was mocked up by using pre-configured slip values of 0.02 for concrete and 0.06 for sand. This binary classification was assumed as a simple output of an expert vision-based slip detection system. For example, if the robot was at the 2 m position, the profile point associated with 3 m would have a slip value predicted as 0.02, and points associated with positions 4, 5, and 6 m would have slip values of 0.06. Finally, the dashed line is the ground truth pitch angle to indicate a slight change in terrain angle affecting the slip. Because the system was analysed only in one dimension, the changes in the slope were not considered.

4.4.2 Models Analysis

The results are provided with T_{VO} = 0.2, 1, 3 s. In this analysis, VO is always providing measurements. In simulations, it was already established that any form of slip prediction allows for fairly good position estimation without any VO, but these measurements improve the overall performance. The maximum T_{VO} was set to three seconds, which implies approximately 1.2 m between VO measurements. Given the mounting point of the stereo camera, such a long distance



(a) T_{VO} = 0.2 s (b) T_{VO} = 1 s (c) T_{VO} = 3 s

Figure 4.11: Summary of metrics for errors in postion estimation for each model when driving in Sand Bay Beach. Vertical axis limits truncate some bars. Models from left to right: Model 0, Model 1, Model 2, Model 3A, Model 3B, and Model 1 with wheel current measurements.

between measurements often provides not enough overlap and results in erroneous VO estimates, which EKF fuses without further validation. The aim is to assess how the wheel slip prediction improves position estimation even when fusing imperfect VO measurements. On the other hand, frequent VO provides a precise estimation of wheel slip that may collide with predicted results.

For this work, measurement covariance for slip prediction, $\sigma_{\lambda}^{2} = 0.000279$, was carefully selected based on trials and errors to achieve comparable results and was always the same across all models. It resulted from not having any terrain classifier implemented – slip prediction is simulated based on the test profile presented in Fig. 4.10. The test profile is not necessarily the best and optimal solution, but it is comparable across different models. Also, all slip measurements in Models 1, 3A, and 3B were done every 5 s as in simulations. Finally, the initial profile within Models 3A and 3B was initialised with flat values of 0.02 (expected wheel slip on tarmac and concrete) without any indication of a planned traverse over the sand. As indicated in simulations, in the case of Model 3B, an initial profile that differs from what is expected results in decreased variance in position estimation when a change in slip is detected.

The metrics results for all models are presented in Fig. 4.11, where blue bars are for Model 0

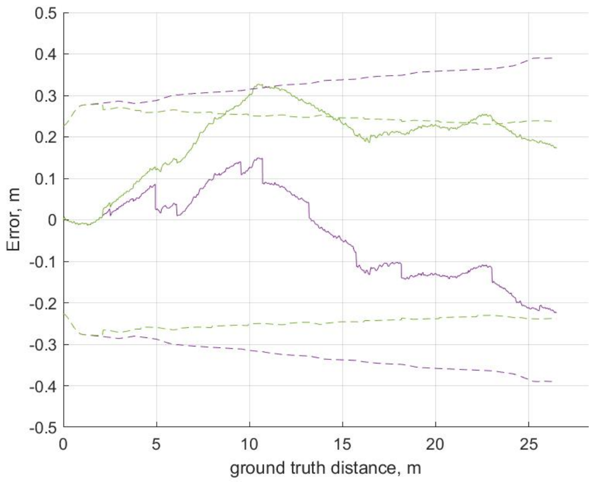


Figure 4.12: Example of errors in position estimation for Models 3A (purple) and 3B (green) with T_{VO} = 3 s. Dashed lines indicate 3σ confidence margin. Note that even though Model 3A provides a lower error, Model 3B is more immune to sudden changes in position estimation resulting from erroneous VO measurements.

(EKF with Integrated Slip Estimation), red - Model 1, yellow - Model 2, purple - Model 3A, green - Model 3B, and cyan - Model 1 with wheel current measurements. The latest model in this set is discussed in section 4.4.3.

What stands out is that when frequent VO measurements are available, as seen in Fig. 4.11(a), Models 0 and 1 give the best results. It is understood that imperfect wheel slip measurements for Models 2 and 3 using simulated test profile dominated λ estimated by VO and WO. Even though the same test profile was used for Model 1 (red bars), it has more flexibility in adjusting λ based on other available measurements. From Fig. 4.11(b), it is interesting to note that Model 1 gave the best accuracy when T_{VO} was one second. In this setting, VO measurement covariance was large enough to allow information from direct slip measurement to be fused. Otherwise, VO dominates in providing the indirect slip estimate, similarly to Model 0. However, when VO starts to provide erroneous estimates, it also harms Model 1 accuracy, as seen in Fig. 4.11(c) top row. What is exciting, is that even when VO measurements are rare, models that utilise profile provide small error enabling accurate position estimation.

All models which utilise slip profiles (Models 2, 3A, and 3B) gave comparable results. It is exciting that they all provided good accuracy, even for rare VO measurements. From those models, Model 3B gave the biggest RMSE at T_{VO} = 3 s. However, these metrics do not capture how immune models are to errors in VO measurements. Fig. 4.12 show errors in position estimation (solid lines) with 3σ confidence margin (dashed lines) for Models 3A (purple) and 3B (green). Note that even though Model 3A has a smaller error, it also has many sharp changes that are not present in the error plot for Model 3B. These changes are associated with erroneous VO measurements. Models 2 and 3A benefited from errors in this particular environment, whereas

Model 3B was immune to them. Such robustness is more desired, and we believe that Model 3B can provide better position estimates if there is a proper terrain identification/classification component.

The variance in position estimation for most of the models remained small, leading to error growing beyond the confidence margin at frequent VO measurements. It needs to be investigated in more detail if such behaviour is overfitting due to many indirect slip measurements (VO) or a limitation of the analysis done in one dimension. As the update period between VO measurements increases, so does the variance, which may indicate its relation with VO measurement covariance R_{VO}.

4.4.3 Additional Considerations

4.4.3.1 Using Wheel Current for Slip Measurement

Some researchers [25, 77, 78] indicate that wheel current can be used to estimate wheel slip mainly because the slip is a function of motors' torque proportional to motors' current. Encouraged by their results, we decided to include this measurement in Model 1. However, rather than performing a detailed analysis of wheel-soil interactions, the robot drove again on the same terrain and thus obtained the motors' current and ground-truth slip were then correlated in the form of a linear function. The function was used for post-processing the initial trajectory analysed in section 4.4.2. Covariance of the linear function fit was used as the measurement covariance. The resulting wheel slip measured based on the motors' current is presented as the magenta plot in Fig. 4.10. Notice that it follows the same shape as the ground-truth slip (green line); however, it is more prominent in magnitude. It is similar to what was performed in [25] but limited in scope.

The results of fusing motor wheel current with WO and VO are promising and are presented in Fig. 4.11 as cyan bars. This approach is more immune to errors in VO measurements than Model 0 and original Model 1. It is likely that good performance of using wheels current to estimate slip comes from the high frequency at which they are available (the same rate as WO). Nevertheless, models that use slip profiles were still better, even though their estimates were less frequent.

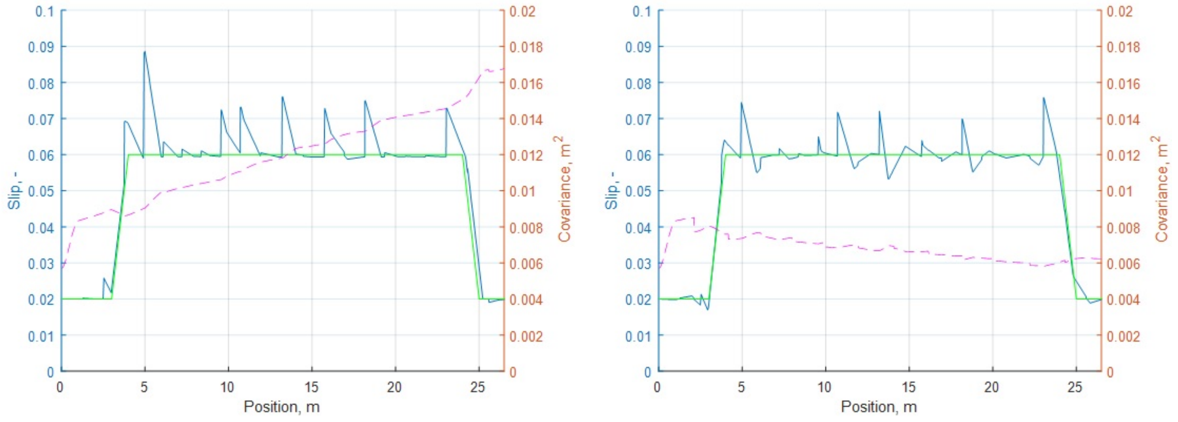
While the potential of this approach seems encouraging, this result must be interpreted with caution. While one drive was used to calibrate the slip pseudo-measurement from current and a second drive to evaluate it, these were over the same track, so it is not yet possible to eliminate other correlating factors – something that is correctly taken into account in [22, 53] where WO is only used for traction control, but not for pose estimation.

4.4.3.2 Slip Profile Localisation

Simulation results indicated an interesting observation regarding Model 3B. It seems that in Fig. 4.7(c) the system recognises the profile and, as a result, reduces the covariance of position estimation. Fig. 4.13 presents slip profiles (estimated - blue, simulated for measurements - green) with the covariance of position estimation (dashed magenta) for Models 3A and 3B. Note that even for Model 3A, the covariance decreases (or reduces its growth rate) at locations where the change in slip is present. For Model 3B, after initial growth, covariance decreases almost for the entire traverse. It indicates as if the slip was being recognised from the profile. This behaviour resembles a Simultaneous Localisation and Mapping (SLAM) [79] or Terrain-Reference Positioning [80]. There is an open possibility of using slip profiles to improve localisation, which is in line with observations from [77, 81], where the vision-based information is correlated with what the robot can feel under its wheels.

4.5 Summary

This chapter compares various models that fuse information about the expected wheel slip with WO and VO to improve the localisation. There is a definite enhancement in position estimation when slip can be predicted, especially at low rates of VO. However, careful consideration in selected measurements needs to be made to avoid unwanted correlation, leading to the overfitting in EKF. The solution is validated using real-world data for odometry measurements and a simulated vision-based classifier for slip predictions. An interesting observation is made where



(a) Model 3A (b) Model 3B

Figure 4.13: Slip profiles against the variance of position estimation for Models 3A and 3B with T_{VO} = 3 s: solid blue line – estimated wheel slip, solid green line – slip profile used for the simulated vision-based classifier, magenta dashed line – variance of the position estimation. Note a decrease in position estimation's variance for Model 3B.

the filter recognises the slip profile, which may open the possibility of slip-based SLAM.

Future work should replace a simulated vision-based classifier with a simple solution that would consider the actual images from the cameras. The proposed system should then be validated on more complex trajectories where the terrain type, and hence predicted slip, varies. An option to extend the functionality further with motors current is also promising, as long as any correlation is correctly interpreted. Finally, it might be worthwhile to explore also options of detecting only changes in the slip rather than actual values. A reactive VO scheduling tool may benefit from knowing expected changes in slip to acquire more images for VO around those areas.

CHAPTER 5

WHEEL SLIP ESTIMATION IN MOTION IN A PLANE

A large portion of robotic challenges require the system to operate on at least a two-dimensional (2D) plane with three degrees of freedom (2D position and rotation). This chapter investigates how to Model 2D localisation with integrated slip estimation. It begins by discussing mathematical models for 2D plane motion in section 5.1.1. Then, two candidate models are defined in section 5.1.2 and compared in section 5.1.3. Next, section 5.2 presents the analysis of how changing process noise statistics impact the system performance. It is a starting point for introducing the adaptation as defined in chapter 3. Then, the adopted reactive VO trigger strategies to the 2D problem are introduced. With the above method successfully applied to a 2D problem, there are high expectations that it would also work for a three-dimensional problem. The final conclusions for this chapter are outlined in section 5.3. In this chapter, Wheel Odometry (WO) is referred to as both measuring the drive wheels' velocity and steering wheels' angle.

5.1 Introduction

5.1.1 Background

The kinematic model for a motion on a 2D plane without taking any slip into consideration is defined as

$$p_x(t) = p_x(0) + \int_0^t v_x(\tau)\cos\psi(\tau)d\tau - \int_0^t v_y(\tau)\sin\psi(\tau)d\tau \tag{5.1a}$$

$$p_y(t) = p_y(0) + \int_0^t v_x(\tau)\sin\psi(\tau)d\tau + \int_0^t v_y(\tau)\cos\psi(\tau)d\tau \tag{5.1b}$$

$$\psi(t) = \psi(0) + \int_0^t \omega(\tau)d\tau \tag{5.1c}$$

Where p_x and p_y are location coordinates in the global frame, v_x and v_y are linear velocities in the platform's local frame, ψ and ω are orientation and the angular rate of the vehicle.

The robotic platform used in this research is a four-wheel drive, four-wheel steering Ackermann vehicle, as seen in Fig. 2.10. The Ackermann steering geometry is well-known and established, for example, in [82]. Formulas are often provided only for front-steering chassis, such as cars, and typically may deal with a single slip parameter (body slip, rather than per wheel slip). They are often also simplified to what is known as a bicycle model, where only one front and one rear wheel are modelled. This simplification comes with a few key assumptions which may not always be well-known. Firstly, the mass distribution on the vehicle has to be symmetric on its lateral axis. Otherwise, the centre of the mass would be offset from the vehicle's centreline, leading to constant side slip. It would be evident at higher velocities when the dynamics' impact on the system cannot be neglected. Secondly, the steering wheels on the platform are physically connected via trapezoidal steering linkage. With it, the outer wheel travels at a wider radius, thus requiring the vehicle to have a differential or independent drive. Without it, the platform would skid, leading to increased sideslip.

Formulas for Ackermann steering geometry for individual steering wheels, which applies to this project, are more complex as they should be treated individually. In general, for the Ackermann steering platform, the key relationship between i'th wheel's angular velocity and its steering angle is given by the platform's angular velocity, ω, around the centre of Ackermann steering radius

$$\omega(t) = \frac{\rho \Omega_i(t) \tan\phi_i(t)}{l_i + B_i \tan\phi_i(t)} \tag{5.2}$$

Where ρ is the wheel's radius, Ω_i and ϕ_i are angular velocity and steering angle of the i'th wheel respectively (both can be measured using encoders), and l_i and B_i are the coordinates of the i'th wheel in the chassis' local frame. Note how the component $B_i \tan\phi_i$ is zeroed in the bicycle model. Also, because of $\tan\phi_i$, (5.2) is invalid when wheels are turned at +/- 90 degrees. Finally, (5.2) does not include the wheel slip angle, which, if included, would shift the origin of the steering arc. Fig. 5.1 provides an overview of the ideal Ackermann steering geometry from which the geometric relation (5.2) is derived. Note that (5.2) holds for any wheel in the system, which implies that wheels on the inside of the Ackermann turn have bigger steering angles but lower angular speeds.

Estimating a vehicle's position in two dimensions means it is subject to more than one slip. In the 1D case, only the longitudinal slip was considered. When driving on a plane, the vehicle can also have a lateral slip (moving to the side), also known as sideslip, and some form of angular slip, which, in the car domain, can be understood as oversteer and understeer. Oversteer is a shorter turning radius than the ideal, whereas understeer is the opposite. Both lateral and angular slips could be neglected in low velocity for a typical car-like Ackermann platform, however, when a robot has independent steering wheels, these slips may also capture any wheels misalignment. For example, lateral slip may occur when the platform drives on the sandy slope and slips sideways, but also when all wheels are turned slightly into one direction. The lateral slip is

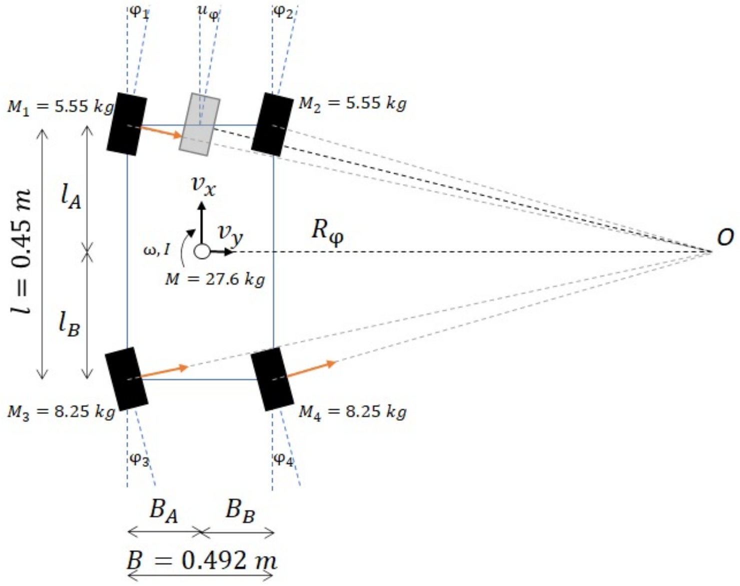


Figure 5.1: Ideal Ackermann steering geometry for the platform used in this project. Note the masses measured at each wheel (M_i), which sum to the value seen in the centre (M). Assuming front steering and rear steering is perfectly synchronised (wheels are independent), the mass distribution would allow for a simplified bicycle model. Also, note that the graphic is out of scale to make it easier to read.

defined as

$$\lambda_y(t) = \arctan \frac{v_y(t)}{v_x(t)} \tag{5.3}$$

The platform used in the research was operated at low velocity, and therefore it was not subject to either oversteer or understeer. However, the angular slip may still occur. For example, in a situation when driving in a straight line, wheels on one side encounter different terrain friction. The platform would skid until all wheels were again subject to the same friction. On the other hand, if, for example, only the front wheels of an independent steering platform were misaligned in one direction, the vehicle would always travel in an arc (non-zero $\tan \phi_i$ in (5.2)). With this in mind, all types of slip should be estimated. Because the platform is rigid, there is no active suspension, and it operates on low velocities, it can be safely assumed that only one lateral and angular slip value for the entire body needs to be estimated. In [83] authors omit the lateral slip altogether, stating that at the low velocity, it can be neglected. It is a valid point for a typical car chassis. However, in a robot with four independent wheels, the lateral motion, similarly to angular slip, may capture any misalignment in the wheel mount. When it comes to a longitudinal slip, it could be debated whether there should be only one value for the entire platform or one for each wheel individually. A single longitudinal slip is easier to observe using VO alone, however, if wheel stiffness parameters were known and additional wheel measurements were available to support per-wheel slip estimation (e.g. torque), it might be beneficial to model more slip values.

Because all drive wheels are tightly speed controlled in the platform used for this research, their velocity measurements would add little to no extra information above the known control input. It would then be nearly impossible to estimate individual slip values. As such, to avoid unobservability, a single longitudinal slip is used.

Finally, we decided to neglect dynamics because the platform operates at low velocities (0.2 - 0.4 m/s). There are examples of Kalman Filter including the dynamic system [83–85], however, in all these cases, the platforms were either operating at higher velocity. Furthermore, most of these applications include an accelerometer which allows measuring the second derivative of the platform's position. Only VO and WO are used in this project, which means that only relative displacement and wheels angles can be measured. Kalman Filter can still estimate the second derivative (acceleration), however, measurements errors can lead to large and erroneous estimates of the acceleration. It, in consequence, may lead to a poor prediction of the next iteration and system instability.

5.1.2 Models Definition

There are different ways to Model 2D sensor fusion and position estimation systems. In this section, two solutions are outlined with a common high-level overview as depicted in Fig. 5.2. Both models estimate chassis velocities using input data (velocities in the local frame of reference). The kinematic model uses these velocities to provide vehicles' position and orientation in the global

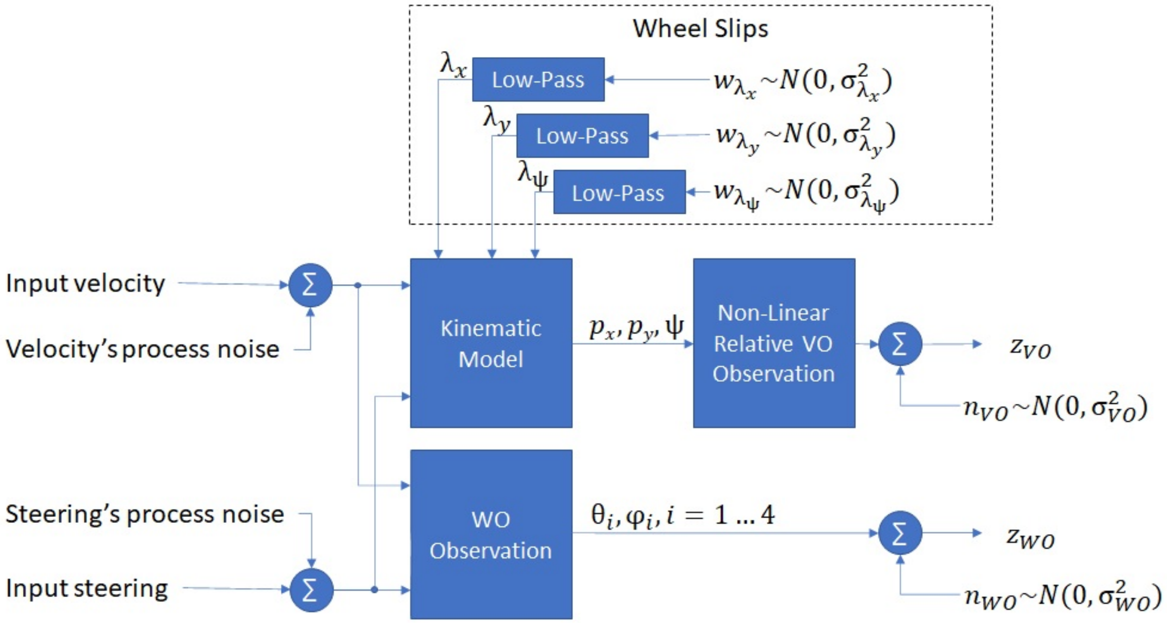


Figure 5.2: High-level architecture of 2D Model. The only difference between Model 1 and Model 2 is in the inputs: Model 1 has them per wheel, whereas in Model 2 they refer to chassis. Also, in Model 2 the WO observation model is non-linear and includes Ackermann steering geometry.

frame. Wheel slip (λ_x - longitudinal; λ_y - lateral; λ_ψ - angular) is added to vehicle's kinematic model and therefore is unobservable from WO perspective. However, WO measurements contribute to identifying control input's uncertainties (velocity and steering process noises). Similarly to the 1D model, wheel slip (all three types) is modelled as a low-pass filter.

1. In the first method, control input is provided to each wheel individually: drive rate u_{Ω_i} and steering angle u_{ψ_i}. As such, their angular rates have to be averaged to provide chassis' linear and angular velocities. Similarly to the 1D model defined in chapter 2, only VO and WO measurements are used to correct the state estimation. Because of that, only a single chassis' slip parameters are estimated. This solution could be treated as a general approach because it does not depend on the Ackermann steering geometry.
2. In the second method, control input is used to command the entire chassis: desired linear velocity u_v and turn curvature u_c. The Ackermann steering geometry is included in the filter, but rather than modelling it inside the state transition function, it forms part of the non-linear WO observation model.

In chapter 2, the model in continuous time was discretised by calculating integrals and applying mathematical theorems to obtain the final equations. Because 2D models are more complex than 1D, they are discretized using trapezoidal approximation. In addition, in the 1D model, the uncertainty of the control input was modelled as a random walk. It was treated as the angular acceleration of the wheel. For the 2D case, to simplify it further, the control input's uncertainty is modelled as white noise. The non-zero mean value is assumed for process noises modelling slip when discussing adaptation in section 5.2.

5.1.2.1 2D Model 1 - Per Wheel Control

This model is controlled via per wheel input. For chassis motion estimation, the weighted average of the motions of each wheel is used. We assumed the platform to be a particle system with all its mass located at wheels. Per wheel masses M_i are used as weights (note mass values in Fig. 5.1). It is then assumed that the chassis slip model corrects any errors in that approach. The averaging is defined as

$$Mv_x(t) = \rho \sum_i M_i \Omega_i(t) \cos\phi_i(t) \tag{5.4a}$$

$$Mv_y(t) = \rho \sum_i M_i \Omega_i(t) \sin\phi_i(t) \tag{5.4b}$$

$$I\omega(t) = \rho \sum_i M_i \Omega_i(t) \big(l_i \sin\phi_i(t) - B_i \cos\phi_i(t)\big) \tag{5.4c}$$

Where M is the total mass of the vehicle and I is the mass moment of inertia of the vehicle, which, with the assumption that the whole mass is located only in the platform's wheels (particle system with four mass particles), can be calculated as

$$I = \sum_i M_i(l_i^2 + B_i^2) \tag{5.5}$$

Combining all together, the state vector with the state transition function, excluding augmented states for better readability, is defined in its discrete form (discretised with a time step Δt) as

$$p_x(k+1) = p_x(k) + \frac{\Delta t}{2}\left(v'_{xx}(k+1) - v'_{xy}(k+1)\right) \tag{5.6a}$$

$$p_y(k+1) = p_y(k) + \frac{\Delta t}{2}\left(v'_{yx}(k+1) + v'_{yy}(k+1)\right) \tag{5.6b}$$

$$\psi(k+1) = \psi(k) + \frac{\Delta t}{2}\left(\omega(k+1) + \lambda_\psi(k+1) + \omega(k) + \lambda_\psi(k)\right) \tag{5.6c}$$

$$v_x(k+1) = \frac{\rho}{M}\sum_i M_i\Omega_i(k+1)\cos\phi_i(k+1) \tag{5.6d}$$

$$v_y(k+1) = \frac{\rho}{M}\sum_i M_i\Omega_i(k+1)\sin\phi_i(k+1) \tag{5.6e}$$

$$\omega(k+1) = \frac{\rho}{I}\sum_i M_i\Omega_i(k+1)\left(l_i\sin\phi_i(k+1) - B_i\cos\phi_i(k+1)\right) \tag{5.6f}$$

$$\lambda_x(k+1) = \left(1 - \frac{\Delta t}{T_0}\right)\lambda_x(k) + \frac{\Delta t}{T_0}w_{\lambda_x}(k) \tag{5.6g}$$

$$\lambda_y(k+1) = \left(1 - \frac{\Delta t}{T_1}\right)\lambda_y(k) + \frac{\Delta t}{T_1}w_{\lambda_y}(k) \tag{5.6h}$$

$$\lambda_\psi(k+1) = \left(1 - \frac{\Delta t}{T_2}\right)\lambda_\psi(k) + \frac{\Delta t}{T_2}w_{\lambda_\psi}(k) \tag{5.6i}$$

$$\theta_i(k+1) = \theta_i(k) + \frac{\Delta t}{2}\left(\Omega_i(k) + \Omega_i(k+1)\right) \tag{5.6j}$$

$$\Omega_i(k+1) = u_{\Omega_i}(k+1) + w_{\Omega_i}(k) \tag{5.6k}$$

$$\phi_i(k+1) = u_{\phi_i}(k+1) + w_{\phi_i}(k) \tag{5.6l}$$

Note that some states are defined as functions of other states' predictions $(k+1)$. This has been made only for better readability to encapsulate common terms. To make (5.6) easier to read, $v'_{xx}(k+1)$, $v'_{xy}(k+1)$, $v'_{yx}(k+1)$, and $v'_{yy}(k+1)$ were defined as

$$v'_{xx}(k+1) = v_x(k+1)\left(1-\lambda_x(k+1)\right)\cos\psi(k+1) + v_x(k)\left(1-\lambda_x(k)\right)\cos\psi(k) \tag{5.7a}$$

$$v'_{xy}(k+1) = v_y(k+1)\left(1-\lambda_y(k+1)\right)\sin\psi(k+1) + v_y(k)\left(1-\lambda_y(k)\right)\sin\psi(k) \tag{5.7b}$$

$$v'_{yx}(k+1) = v_x(k+1)\left(1-\lambda_x(k+1)\right)\sin\psi(k+1) + v_x(k)\left(1-\lambda_x(k)\right)\sin\psi(k) \tag{5.7c}$$

$$v'_{yy}(k+1) = v_y(k+1)\left(1-\lambda_y(k+1)\right)\cos\psi(k+1) + v_y(k)\left(1-\lambda_y(k)\right)\cos\psi(k) \tag{5.7d}$$

To simplify the state, in (5.7) we assumed a small-angle approximation for $\psi(k+1) - \psi(k)$, which reduces trigonometric terms to

$$\cos\psi(k+1) \approx \cos\psi(k) - \frac{\Delta t}{2}\left(\omega(k+1) + \lambda_\psi(k+1) + \omega(k) + \lambda_\psi(k)\right)\sin\psi(k) \tag{5.8a}$$

$$\sin\psi(k+1) \approx \sin\psi(k) + \frac{\Delta t}{2}\left(\omega(k+1) + \lambda_\psi(k+1) + \omega(k) + \lambda_\psi(k)\right)\cos\psi(k) \tag{5.8b}$$

Note also that in (5.7) both λ_x and λ_y are added to the kinematic model in a similar fashion to the 1D problem as $(1-\lambda)$. Two linear velocities model plane motion, therefore, each is scaled with their respective slip in a manner that was established in chapter 2. The angular slip λ_ψ is treated as an added bias to the angular velocity ω when calculating the platform's rotation ψ. This approach is because a scaling factor would not capture constant angular offset when not steering. A zero rate would still be scaled to zero, therefore, a bias is required. Furthermore, the angular slip λ_ψ is assumed unobservable for ω, hence it is present only in (5.6c) rather than in (5.6f).

In this model, θ_i and ϕ_i are directly measured using encoders on each wheel in both steering and driving axes, thus the linear WO observation function is given as

$$z_{\theta_i}(k) = \theta_i(k) + n_{\theta_i}(k) \tag{5.9a}$$

$$z_{\phi i}(k) = \phi_i(k) + n_{\phi i}(k) \tag{5.9b}$$

5.1.2.2 2D Model 2 - Chassis Control

In this approach, control input commands the chassis, therefore, it controls the kinematic model directly. From Fig. 5.1, the commanded steering curvature is defined as

$$u_c(t) = \frac{\tan u_\phi(t)}{l_A} = \frac{1}{R_\phi(t)} \tag{5.10}$$

Where u_ϕ is the steering angle of an ideal steering wheel located in the front of the platform on its longitudinal axis (l_A from the centre of rotation), and the R_ϕ is the radius of the Ackermann turn for the vehicle's centre of gravity. Because the radius is infinite when driving in a straight line, the curvature is a better way of modelling the Ackermann geometry, especially when linearising around the zero steering angle.

Eq. (5.3) defines lateral slip, $\lambda_y(t)$, as ratio of $v_y(t)$ to $v_x(t)$. It can be rewritten, assuming small-angle approximation, so that $v_y(t)$ is a function of $v_x(t)$, which, on the other hand, is the sum of control input, $u_v(t)$, and input's uncertainty (white process noise), $w_v(t)$.

$$v_y(t) = v_x(t)\lambda_y(t) \tag{5.11a}$$

$$v_x(t) = u_v(t) + w_v(t) \tag{5.11b}$$

Lastly, the angular velocity of the vehicle is taken from its circular motion (note added commanded curvature's uncertainty as $w_c(t)$)

$$\omega(t) = \frac{v_x(t)}{R_\phi(t)} = \big(u_v(t) + w_v(t)\big)\big(u_c(t) + w_c(t)\big) \tag{5.12}$$

Finally, the state vector with its transition function in its discrete form (following the same discretization workflow as in section 5.1.2.1) can be defined as

$$
\begin{aligned}
p_x(k+1) &= p_x(k) + \frac{\Delta t}{2}\big(v'_{xx}(k+1) - v'_{xy}(k+1)\big) && \text{(5.13a)}\\
p_y(k+1) &= p_y(k) + \frac{\Delta t}{2}\big(v'_{yx}(k+1) + v'_{yy}(k+1)\big) && \text{(5.13b)}\\
\psi(k+1) &= \psi(k) + \frac{\Delta t}{2}\big(v_x(k+1)c(k+1) + \lambda_\psi(k+1) + v_x(k)c(k) + \lambda_\psi(k)\big) && \text{(5.13c)}\\
v_x(k+1) &= u_v(k+1) + w_v(k) && \text{(5.13d)}\\
c(k+1) &= u_c(k+1) + w_c(k) && \text{(5.13e)}\\
\lambda_x(k+1) &= \left(1 - \frac{\Delta t}{T_0}\right)\lambda_x(k) + \frac{\Delta t}{T_0} w_{\lambda_x}(k) && \text{(5.13f)}\\
\lambda_y(k+1) &= \left(1 - \frac{\Delta t}{T_1}\right)\lambda_y(k) + \frac{\Delta t}{T_1} w_{\lambda_y}(k) && \text{(5.13g)}\\
\lambda_\psi(k+1) &= \left(1 - \frac{\Delta t}{T_2}\right)\lambda_\psi(k) + \frac{\Delta t}{T_2} w_{\lambda_\psi}(k) && \text{(5.13h)}\\
\theta_i(k+1) &= \theta_i(k) + \frac{\Delta t}{2\rho}\big(v_x(k+1) + v_x(k)\big) && \text{(5.13i)}
\end{aligned}
$$

Similarly to the previous model, $v'_{xx}(k+1)$, $v'_{xy}(k+1)$, $v'_{yx}(k+1)$, and $v'_{yy}(k+1)$ were defined to simplify (5.13) and are given as

$$
\begin{aligned}
v'_{xx}(k+1) &= v_x(k+1)\big(1-\lambda_x(k+1)\big)\cos\psi(k+1) + v_x(k)\big(1-\lambda_x(k)\big)\cos\psi(k) && \text{(5.14a)}\\
v'_{xy}(k+1) &= v_x(k+1)\lambda_y(k+1)\sin\psi(k+1) + v_x(k)\lambda_y(k)\sin\psi(k) && \text{(5.14b)}\\
v'_{yx}(k+1) &= v_x(k+1)\big(1-\lambda_x(k+1)\big)\sin\psi(k+1) + v_x(k)\big(1-\lambda_x(k)\big)\sin\psi(k) && \text{(5.14c)}\\
v'_{yy}(k+1) &= v_x(k+1)\lambda_y(k+1)\cos\psi(k+1) + v_x(k)\lambda_y(k)\cos\psi(k) && \text{(5.14d)}
\end{aligned}
$$

A non-linear WO observation model for steering angle can be built from Fig. 5.1, where the individual wheel's steering angle is defined as

$$
z_{\phi_i}(k) = \arctan\frac{l_i}{R_\phi(k) - B_i} + n_{\phi_i}(k) = \arctan\frac{l_i}{\frac{1}{c(k)} - B_i} + n_{\phi_i}(k) = \arctan\frac{l_i c(k)}{1 - B_i c(k)} + n_{\phi_i}(k) \quad \text{(5.15)}
$$

The observation for θ_i remains unchanged and as defined in (5.9a). With this approach, four independent sensors indirectly measure each control input with associated uncertainty, reducing the measurement error's impact.

5.1.3 Models Comparison and Discussion

Similarly to the 1D problem, one calibration trajectory in a car park was used to estimate VO measurement error for 2D motion. First, raw VO imagery was captured to estimate the six degrees of freedom (x, y, z, roll, pitch, yaw) error for different T_{VO} and corresponding covariance matrices. Any mean values from the errors were neglected. It is believed that bias could arise from stereo camera miscalibration and lack of perfect synchronisation between VO and ground truth. Then, only variances of p_x, p_y, and ψ were used to build R_{VO} for 2D motion; any cross-correlation that might have been present when analysing errors is neglected. It is dictated by the fact that the VO algorithm [33] uses RANSAC to solve the perspective-three-point problem [86] (further augmented by using M-estimator [87]), which provides independent estimates for each degree of freedom. Non-zero cross-correlation may have been a result of how the rover was operated. A possible improvement could be made by estimating R_{VO} using photo-realistic simulations with a perfect ground-truth so that VO results are isolated from the environment's impact and rover's control system.

The main metric used when comparing both models was the Euclidean distance error. It allows plotting a single error metric for 2D movement with its 3σ confidence margin defined as

$$3\sigma = 3\sqrt{P_{xx} + P_{yy}} \tag{5.16}$$

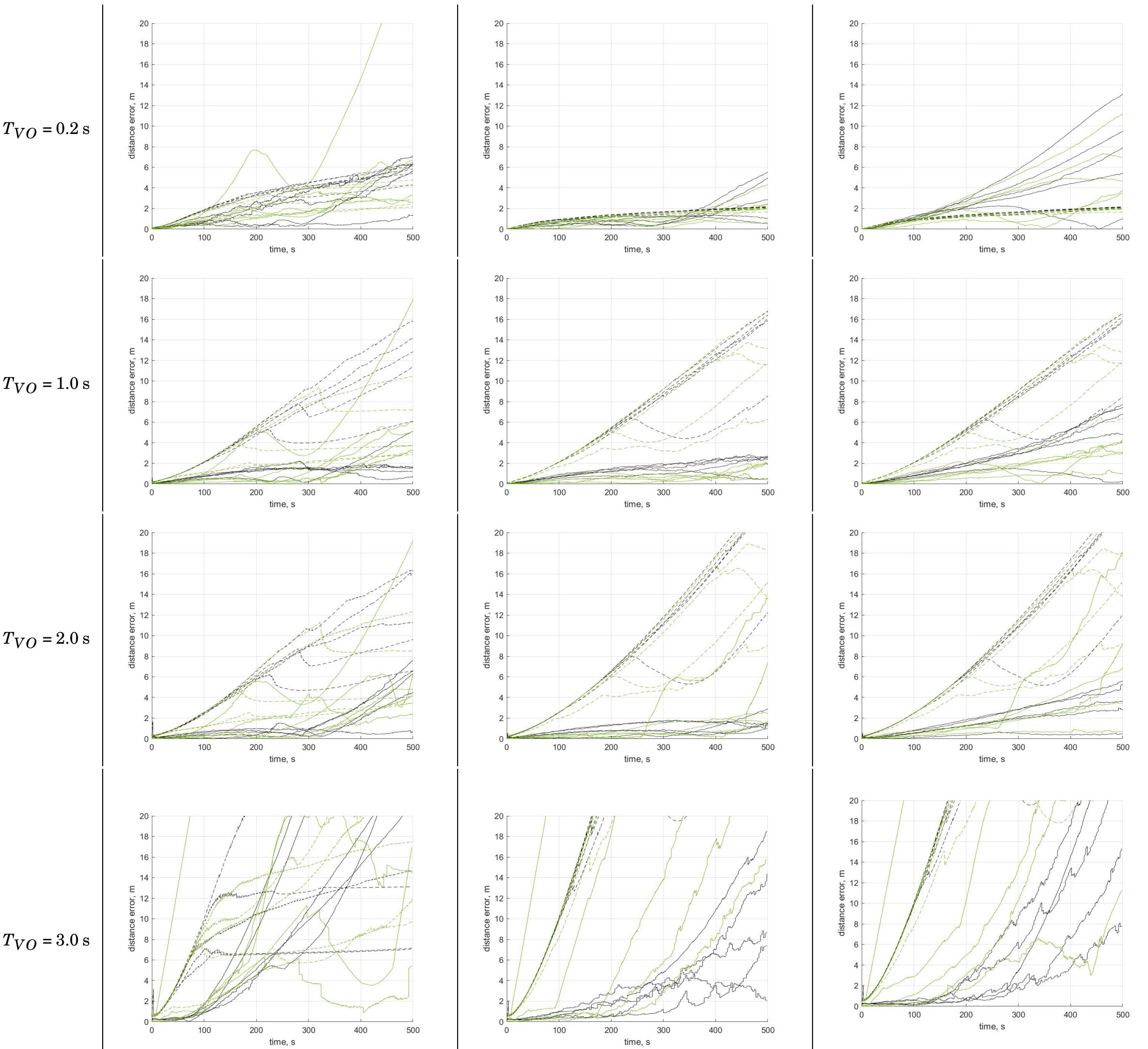
Where P_{xx} and P_{yy} are variances of p_x and p_y position estimates respectively.

For the comparison, only trajectories from the Hengrove dataset were used (described in section 2.3.1). To make results representative, time constants related to longitudinal and lateral slips were set to 20 s (the same value as for T_0 in the one-dimensional EKF defined in chapter 2), whereas the time constant associated with the low-pass filter driving the angular slip was set to 1 s. Process noise variances were set as presented in Table 5.1. Typically, higher values result in a noisier P matrix which could be seen as a saw-tooth in 3σ confidence margin for position estimation. The variance on control inputs for Model 1 may seem quite high, however, in the case of $\sigma^2_{\Omega_i}$, it was calculated by assuming a unit standard deviation of a linear velocity impact on the platform (what is equivalent to σ_v for Model 2), scaled by an inverse of wheels radius $\rho = 0.112$ m to convert it into the angular rate. As for the $\sigma^2_{\phi_i}$, it was selected based on trials and errors. Low values result in high trust in control input, making little use of any measurement updates. On the other hand, higher $\sigma^2_{\phi_i}$ results in the higher trust being placed in measurements, at which point wheel slip estimates govern a balance between WO and VO for orientation estimation.

To compare both models, it is best to present their results side by side, as seen in Fig. 5.3. At first glance, it becomes clear that Model 2 outperforms Model 1. Some merit may be given to Model 1 when frequent VO updates are available (top row). In those settings, only one trajectory is completely erroneous, but for the rest Model 1 correctly tracks the convariance in position estimation. The same cannot be said about Model 2, where errors from a few trajectories grow beyond 3σ confidence margin. In general, it is assumed that the poor performance of Model 1 is

Table 5.1: Initial process noise variances for 2D models comparison.

	$\sigma^2_{\Omega_i}$	$\sigma^2_{\phi_i}$	σ^2_v	σ^2_c	$\sigma^2_{\lambda_x}$	$\sigma^2_{\lambda_y}$	$\sigma^2_{\lambda_\psi}$
Model 1	81	100	N/A	N/A	0.25	0.49	1
Model 2	N/A	N/A	1	1	0.25	0.49	1



(a) Model 1

(b) Model 2

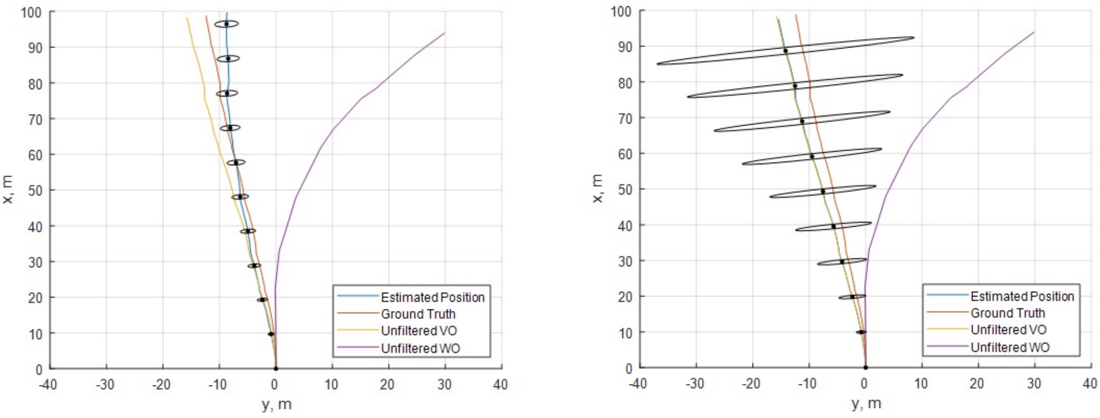
(c) Model 2 with non-zero cross-correlation in R_{VO}

Figure 5.3: Comparison of two models' distance errors using trajectories from the Hengrove dataset. Black plots represent trajectories on the tarmac and green on the grass. Dashed lines represent the 3σ confidence margin. Note that all errors are positive as they are 2D Euclidean distances.

related to having four individual steering wheels. Any wheels misalignment confuses the filter very much. Also, having one control input per wheel does not allow for any averaging of the noise, which happens in Model 2, where WO's observation smooths the impact of any misalignment. On the other hand, Model 2 does not do a good job in tracking the covariance. Not only for a frequent VO as mentioned earlier, but also for less frequent as seen in the second and third rows. The errors are quite small, but the covariance continues to grow exponentially. Fig. 5.3(c) presents Model 2 with non-zero R_{VO} cross-correlation and is only provided to illustrate impact erroneous covariance matrix may have.

Note how some trajectories seem as if they bounce back from the zero distance error, for example, as seen in some black trajectories in Fig. 5.3(b) top row. It results from the platform starting to turn into the opposite side of the ground-truth path and crossing it. What is interesting, at the intersection, the error is almost zero, which indicates accurate tracking of total distance travelled, which is in line with observations from chapter 2. One such trajectory is presented in Fig. 5.4(a), where black ellipses indicate a 3σ error margin around highlighted points. On the same plot, at the beginning, the estimated position path follows the path of unfiltered VO closely. It may suggest that having only VO measurements would be a better solution, especially for fast VO acquisition (in Fig. 5.4, T_{VO} = 0.2 s). However, when no WO measurements are provided to better estimate noises on the control inputs and to provide a comparison for slip estimation, the 3σ error margin is much bigger, as shown in Fig. 5.4(b). What is interesting, when using frequent VO measurements alone, 3σ margins look similarly to those seen in Fig. 5.3(b) middle rows as opposed to the top row (T_{VO} = 0.2 s). Further investigation is required to understand this behaviour better.

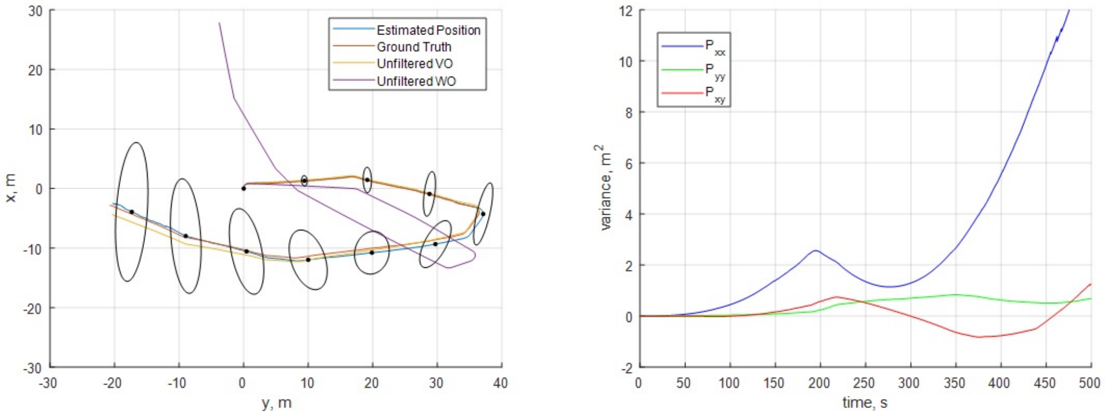
Another observation can be made when inspecting the decrease in the 3σ margin on some trajectories. For example, in the second row in Fig. 5.3(b) there is one green trajectory where the margin goes visibly down around 200^{th} second and then starts to grow exponentially from around 260^{th} second. It is a result of the platform turning. The example trajectory with plots of P_{xx}, P_{yy}, and P_{xy} variances are presented in Fig. 5.5. In this trajectory, the rover started to drive forward on X-axis for one metre, then turned right and continued on the Y-axis until the U-turn. Similarly to Fig. 5.4, the ellipsoidal 3σ margin grows on the vehicle's local frame's lateral axis - X in this example. This is reflected by the growth of variance P_{xx} seen in Fig. 5.5(b). When turning, the platform starts to move on the X-axis, which results in a reduction of P_{xx} and an increase in P_{yy}. Also, the P_{xy} starts to change to reflect the turn, which is also seen as a change in ellipsoidal shape in Fig. 5.5(a). Once the platform is traversing along the Y-axis again, the P_{xx} resumes growing. It seems that non-zero curvature improves the localisation performance, however, it is not immediately obvious why. The model is fully observable thus, it may be related to wheel misalignment or measurement covariance matrices. It would be an exciting topic to explore in more detail using more trajectories. Nonetheless, further investigation is beyond the scope of this project.



(a) Fusion of VO and WO

(b) Only VO measurements

Figure 5.4: An example trajectory for Model 2 with $T_{VO} = 0.2$ s. On the left, note how the estimated position (blue) intersects with the ground truth (red). Unfiltered VO and WO estimations are presented to better understand the raw information from sensors. Black ellipses represent 3σ errors around the estimates. On the right, WO measurements are not fused. Notice how the estimated position follows the path of unfiltered VO, but 3σ errors are greater than when fusing both VO and WO.



(a) Example trajectory for Model 2

(b) Uncertainties of the example trajectory

Figure 5.5: An example trajectory for Model 2 with $T_{VO} = 1$ s with a U-turn on the left. The covariance of position estimation grows on the lateral axis which is reflected in the shape of the ellipsoidal 3σ margin during the turn, and in the variances plot on the right.

To summarise, in the case of a platform with independent wheels, where only VO and WO measurements are available, a simple kinematic Model 2 outperforms Model 1 implementing per wheel control. In the following sections, when discussing the adaptive EKF and reactive VO scheduling, only Model 2 is used.

5.2 Adaptive EKF and Reactive VO Scheduling for 2D Model

This section begins with an analysis of how changing each parameter related to process noises impact the model's accuracy. The aim is to establish if the adaptive solution would impact the accuracy in the first place. Then, the same Sage-Husa [68, 69] algorithm as presented in section 3.2.2 is implemented, tested, and discussed. Finally, a simple reactive VO solution is presented to assess its feasibility for the 2D motion problem.

5.2.1 Parameters Sensitivity

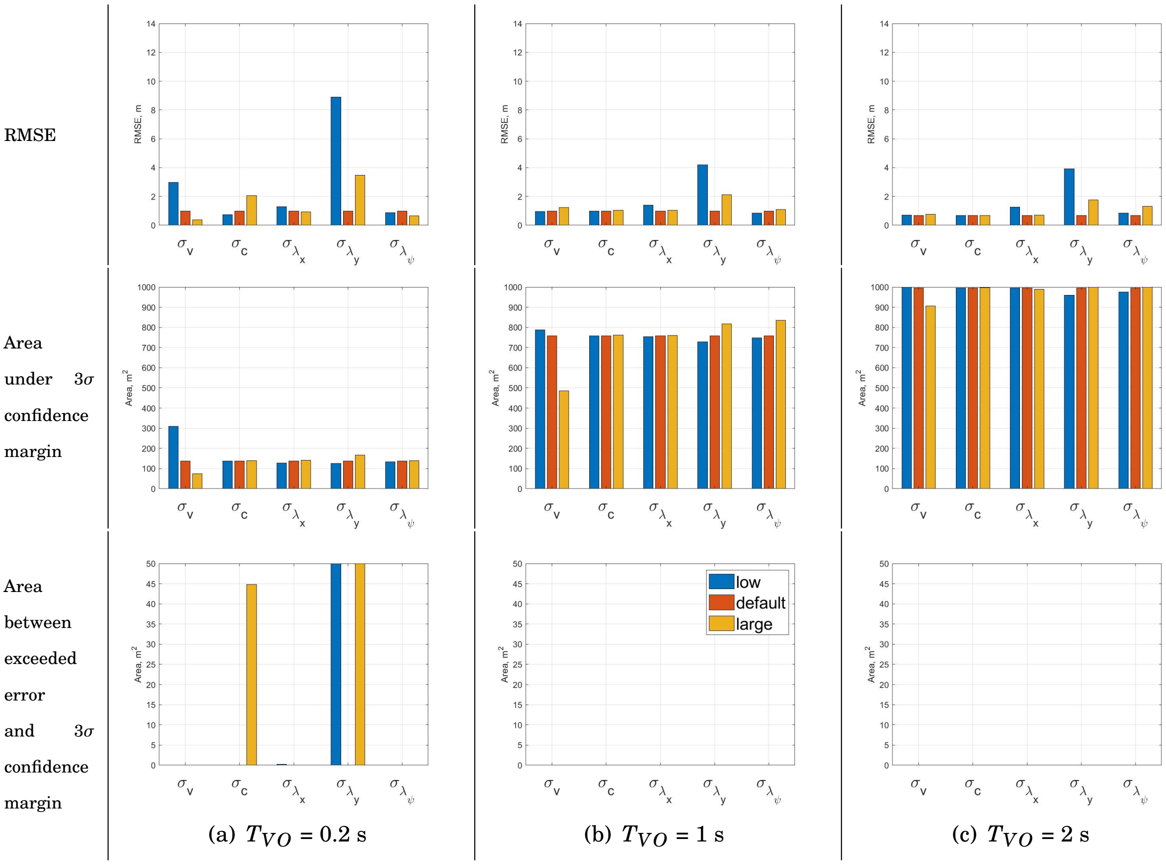
The first task is to manually change process noise parameters and assess how they impact the filter's position estimation performance. If the model's accuracy does not change when varying these parameters, an adaptive EKF would only add computational overhead without any gain in the quality of the estimation. The parameters listed in Table 5.1 for Model 2 are selected as the starting point. To have a clear view of the impact these parameters may cause, they are changed to either a low value, 0.1, or a high value, 10. To better isolate the impact of varying parameters on the quality of the position estimation, this analysis is performed for only one straight trajectory on the tarmac from the Hengrove dataset. The analysis was performed under T_{VO} = 0.2, 1, and 2 s, as the process noise may have a different impact on position estimation depending on the frequency of VO measurements.

Fig. 5.6 shows the results of changing standard deviation of each process noise using metrics defined in section 1.6. Middle bars, red, are always the same as they represent default parameters. When performing the analysis, only one noise statistic was changed at a time to give a clear indication of its impact on the filter's performance.

Below is a list of several highlights of this comparison

- The lowest RMSE (top row) was for a system fusing VO every two seconds (Fig. 5.6(c)), with the system fusing VO every second (Fig. 5.6(b)) providing only a slightly bigger RMSE. This behaviour needs to be investigated in the future to determine why it is not the case, as one would expect lower RMSE for frequent VO measurements. We believe it results from an insufficient number of WO measurements in between VO, which does not offer enough information to estimate the increased number of wheel slip parameters correctly. The size of the 3σ confidence margin of position estimation (middle row) is related to VO measurement covariance matrix R_{VO} as expected. The more frequent VO, the smaller the covariance matrix and the smaller the margin. However, the margin is too small when operating frequent VO leading to error in position estimation exceeding the confidence margin.

- σ_v impacts the 3σ confidence margin of the position estimation. The smaller value, the larger the confidence margin and vice versa. It also impacts RMSE, however, in a different way depending on T_{VO}: for frequent VO, smaller σ_v increases RMSE, whereas,



(a) T_{VO} = 0.2 s (b) T_{VO} = 1 s (c) T_{VO} = 2 s

Figure 5.6: Parameters sensitivity for Model 2 for varying process noise statistics showing metrics for errors in position estimation. Noise statistics on individual plots going from left to right: σ_v, σ_c, σ_{λ_x}, σ_{λ_y}, and σ_{λ_ψ}. Each time only one standard deviation was changed compared to default parameters as outlined in Table 5.1. In each plot, the middle bars are always the same. Vertical axis limits truncate some bars. Note the legend in the centre-bottom plot, where blue - lower values than the default, red - default values, and yellow - increased values.

for infrequent VO, it increases the error. However, the impact of σ_v diminishes as T_{VO} increases.

- σ_c seems to impact only RMSE for frequent VO. However, what is not captured in these metrics is the shape of the covariance of other state estimates. In this case, the bigger σ_c, the noisier is the variance in orientation, $P_{\psi\psi}$.

- The longitudinal slip has a similar impact on the 2D model to the 1D model. Also in the case of the 2D model, lower σ_{λ_x} results in an overestimation of the distance travelled, best seen in Fig. 5.6(a). For low values, the filter trusts WO more, leading to increased RMSE, which confirms observations from chapter 2. The change in the error for smaller σ_{λ_x} is not substantial, as this analysis was performed using a tarmac trajectory that already offers very little wheel slip.

- The lateral slip has the most impact on the shape of the trajectory. It relates to side motion which may come from misaligned wheels. For low σ_{λ_y}, the estimated position follows the unfiltered WO path more closely, leading to increased RMSE (note how unfiltered WO path compares to ground truth in Fig. 5.4 and Fig. 5.5). Interestingly, too high σ_{λ_y} also impacts the shape of the trajectory. At that point, any small changes in the steering may be enlarged by the filter resulting in a different direction of movement.

- σ_{λ_ψ} impacts the system differently depending on T_{VO}. For frequent VO in Fig. 5.6(a), both smaller and bigger σ_{λ_ψ} offer lower RMSE. However, as T_{VO} increases, this relation changes so that in Fig. 5.6(c), the smallest RMSE is for the default parameter. As mentioned earlier, these metrics do not capture the shape of covariance and as such large σ_{λ_ψ}, similarly to σ_c, results in noisier the variance in orientation, $P_{\psi\psi}$.

Overall, it is possible to conclude that changes to noise statistics do not always impact the system in the same way, but they rather depend on T_{VO}. Tuning these noise statistics is better left for an adaptive EKF rather than trying to manually define a set of parameters for a given T_{VO}. Especially with varying time between VO measurements (reactive VO), the impact of incorrectly defined noise statistics may not be seen right away but after some time.

5.2.2 Adaptive EKF

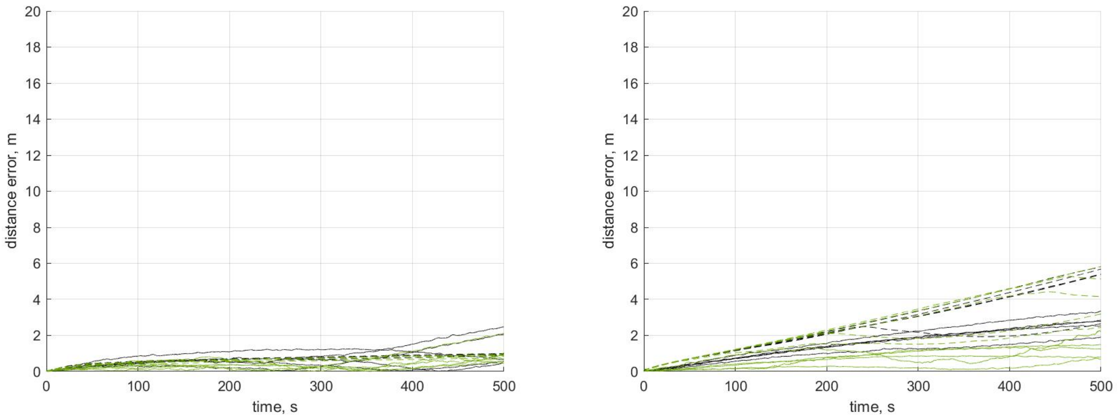
The Sage-Husa algorithm is used for the 2D problem as selected in section 3.2.2 to see if it will work with only minor changes. For the 1D problem, a simulation campaign was designed to try and find the best set of weights for R_{VO}, R_{WO}, and Q adaptation. However, in this case a set of parameters was selected arbitrarily based on several trials and errors, namely N_{WO} = 100, N_{VO} = 10000, and N_Q = 10000. The parameters are far from optimal, but they show whether the idea of adaptive EKF and later reactive VO scheduling can be applied to higher dimensions. More extensive simulations should be designed in the future, considering the possibility of different weights within WO and VO, for example, different weights for each degree of freedom in the VO estimate.

The following additional considerations and assumptions were also taken into account:

- All measurement noises are unbiased and therefore have zero mean value. In chapter 3, it was shown that a non-zero mean value for VO may improve its performance for bigger T_{VO}. However, an additional sensor to complement VO is required to verify that properly. Without it, adapting VO impacts both wheel slip estimation and VO's mean residual, making one unobservable.

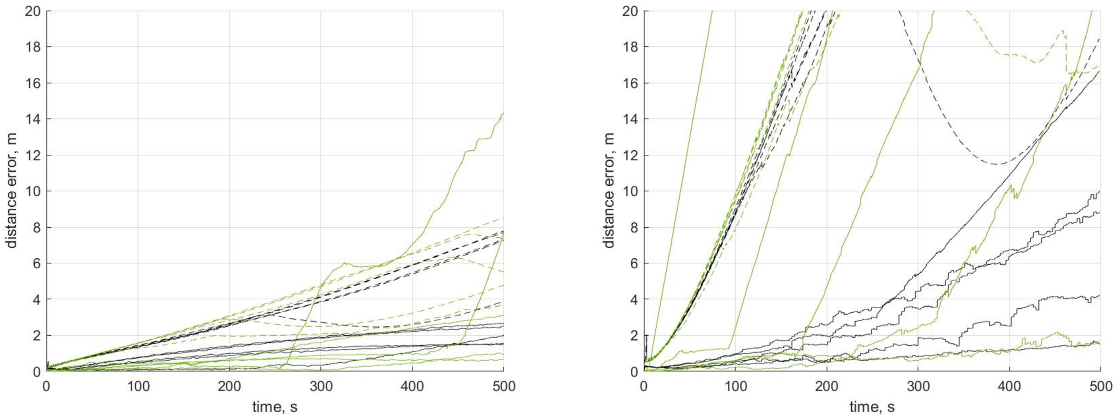
- No cross-correlation in measurement noise covariance matrix is allowed for both R_{VO} and R_{WO}. For WO, each encoder measurement is independent, making this assumption valid at any time. In the case of VO, the delta motion is estimated for all degrees of freedom. It would make sense to allow cross-correlation in R_{VO}, however, early results indicated worse outcome for EKF (section 5.1.3 and Fig. 5.3(c)). As a consequence, during the R-adaptation step, all values outside the main diagonal in R_{VO} and R_{WO} are set to zero. Also, in case when VO and WO measurements arriving at the same time, there is no cross-correlation between R_{VO} and R_{WO}.

- For R_{WO}-adaptation, only the measurement noise related to steering angle is adapted. The position estimation performance was much better when the measurements of wheels rotation were not adapted. In the future, it might be advantageous to have separate adaptive gains for steering and wheels' rotation, however, here, we decided to adopt a more



(a) Errors in distance for T_{VO} = 0.2 s

(b) Errors in distance for T_{VO} = 1.0 s



(c) Errors in distance for T_{VO} = 2.0 s

(d) Errors in distance for T_{VO} = 3.0 s

Figure 5.7: Distance errors for an adaptive 2D model. Black plots are for trajectories on the tarmac and green are for trajectories on the grass. Dashed lines indicate 3σ confidence margin. Bad results for the last model with T_{VO} = 3 s are believed to be associated with wrong orientation estimation. Turns can happen rapidly and thus infrequent VO may not fully estimate how far the robot has turned.

straightforward approach.

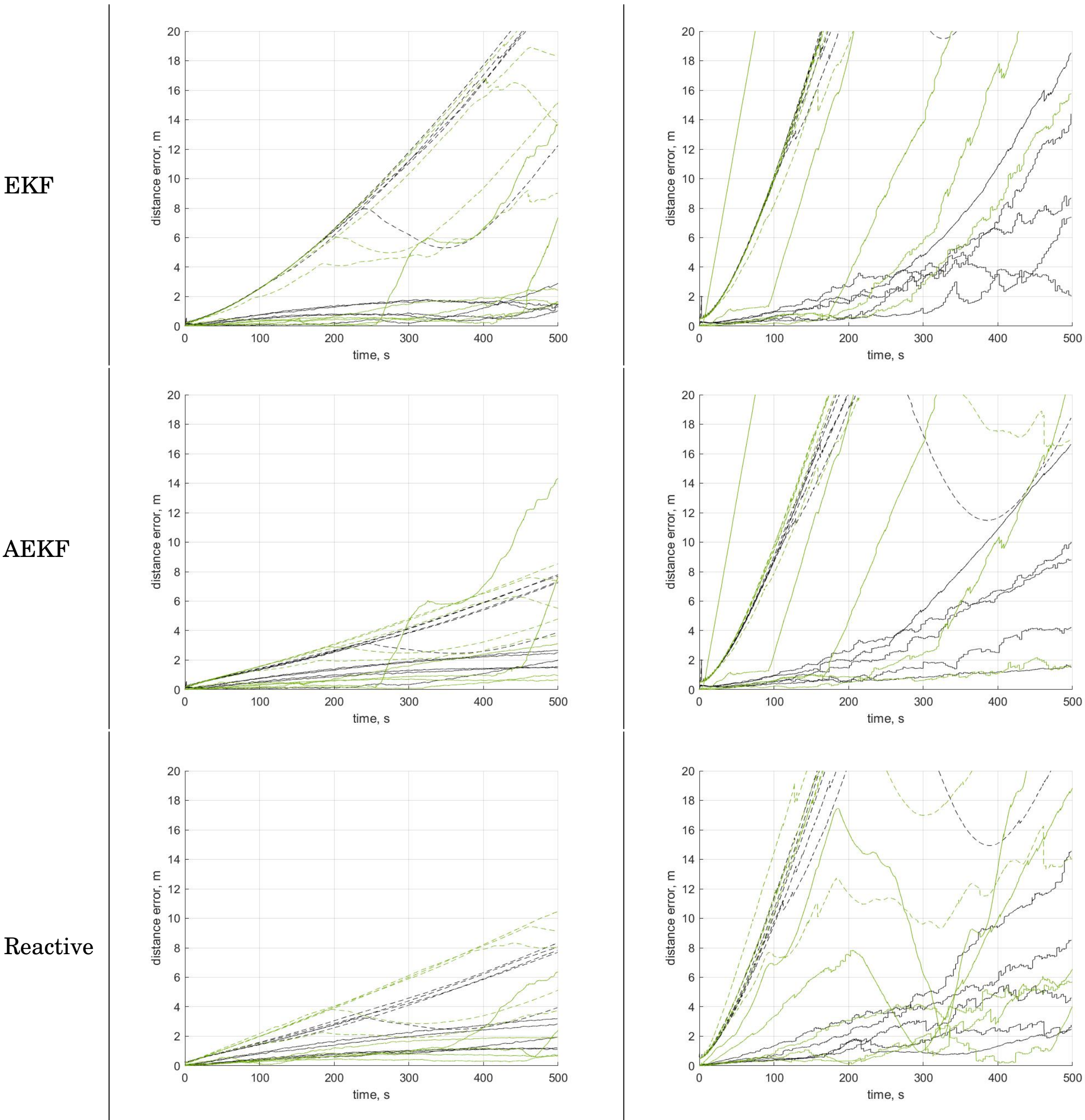
- All process noises are biased and have a non-zero mean value, estimated during the adaptation step. Non-zero mean is expected for wheel slip but can also be valid for control input uncertainty, for example, when wheels constantly report different velocity than commended.
- In the process noise, a cross-correlation between all noises is allowed except for w_{λ_ψ}. The uncertainties on control inputs, w_v and w_c, are assumed to be correlated as the platform may reduce its speed when driving at an angle to reduce the stress on wheels mounts. The correlation between the control input uncertainty and longitudinal, w_{λ_x}, and lateral slip, w_{λ_y}, is assumed on the same basis as the 1D model. For example, the platform may slip when subjected to a sudden change in control velocity. Finally, both longitudinal and lateral slips describe the terrain and are assumed to be correlated. However, the angular slip acts as a bias and mainly captures the wheels' miscalibration. Because of that, we decided not to correlate it with other process noises.

The results of applying an adaptation to Model 2 are presented in Fig. 5.7. Starting with Fig. 5.7(a), notice how errors and confidence margin are much smaller. There are still trajectories for which errors grew outside the confidence margin, but their magnitude is 2-3 times smaller. Fig. 5.7(b) shows an excellent improvement in the tracking of covariance in position estimation. The errors remain similar to EKF, but the filter gives more confident estimates. It is similar in Fig. 5.7(c) with the exception of two grass trajectories. They were also problematic for the EKF, however, in the case of the adaptive filter, uncertainties do not bound them. Finally, there seems to be little to no effect of the adaptive filter in Fig. 5.7(d). With large adaptive gains and low VO frequency, the filter adapts very slowly based on available information.

5.2.3 Reactive VO Scheduling

With adaptive EKF proved to be working for the 2D problem, the next step is to investigate reactive VO scheduling. As before, the aim is to try the same approach as defined in the chapter 3 with some adjustments, namely:

- The strategy which checks the variance of process noise that drove 1D slip estimation is extended to all slip noises and the cross-correlation between w_{λ_x} and w_{λ_y}. In other words, additional VO is scheduled whenever there is an increase by 1.6% in $\sigma^2_{\lambda_x}(k)$, $\sigma^2_{\lambda_y}(k)$, $\sigma^2_{\lambda_x\lambda_y}(k)$, and $\sigma^2_{\lambda_\psi}(k)$ compared to $\sigma^2_{\lambda_x}(k-1)$, $\sigma^2_{\lambda_y}(k-1)$, $\sigma^2_{\lambda_x\lambda_y}(k-1)$, and $\sigma^2_{\lambda_\psi}(k-1)$ respectively.
- One of the trigger strategies is an unknown VO error. To statistically validate VO measurement, a squared Mahalanobis distance [72] is calculated, as defined in (3.8). It follows a Chi-square distribution with the number of degrees of freedom equal to the degrees of freedom provided by the VO estimate itself. In the 1D problem, the VO was valid with



(a) $T_{VO} = T_{VO_{MAX}} = 2$ s

(b) $T_{VO} = T_{VO_{MAX}} = 3$ s

Figure 5.8: Distance errors for 2D reactive VO scheduling compared with EKF and AEKF. Black plots are for trajectories on the tarmac and green are for trajectories on the grass. Dashed lines indicate 3σ confidence margin. Note how reactive solution provides better position estimation for $T_{VO_{MAX}} = 2$ s. For 3 s all models are subject to the same increased errors that are believed to come from wrong orientation estimation.

95% probability if the Mahalanobis distance was not greater than 3.841 (one degree of freedom). In the 2D case, VO estimates delta motion in three degrees of freedom, therefore, the threshold is 7.815.

The results of applying reactive VO scheduling with $T_{VO_{MAX}}$ = 2 s and 3 s are presented in Fig. 5.8, together with EKF and Adaptive EKF (AEKF) for better readability. Inspecting Fig. 5.8(b), notice that one grass trajectory for which error exceeded 3σ confidence margin at the very beginning for both EKF (top row) and adaptive EKF (middle row) achieved lower error and is now within the confidence margin. Nevertheless, the performance of the reactive VO scheduling system with $T_{VO_{MAX}}$ = 3 s is not great. A solid performance of an adaptive filter is required for the reactive VO solution to offer visible improvements. However, note how reducing the maximum time between VO measurements to two seconds, as seen in Fig. 5.8(a) better illustrates the benefits of the reactive VO scheduling system. The reactive solution (bottom row) results in a lower error for grass trajectories while maintaining only a slightly bigger confidence margin compared to AEKF (middle row).

5.3 Summary

In this chapter, it has been shown that the solution for fusion of VO and WO with integrated slip estimation can be also applied to higher dimensions. With a limited number of sensors used, a simple mathematical model proved to be more accurate when tracking the 2D position over the more complex model that includes per wheel control. The model was successfully extended with the existing adaptive algorithm from the chapter 3 to demonstrate the reactive VO scheduling. The 2D model was not so well-tuned as the 1D model, but the solution is viable as a proof of concept.

Besides the parameters tuning, such as the gains for varying T_{VO} in the adaptive EKF, the position estimation for higher dimensions could greatly benefit from additional sensors, such as IMU. Wheels' steering was not too reliable in tracking the vehicle's orientation. Also, VO at a lower framerate struggled when moving in the arc. It is generally more vulnerable to errors during turns as the scenery may change drastically, preventing reliable feature tracking. As a result, even though the platform may correctly track the total distance travelled, it was, on some occasions, integrating its delta motion in the wrong direction. It is not too surprising, as all dead reckoning solutions suffer from this problem. There was hope that an input steering/curvature would fill this gap, but it was insufficient. An IMU could be a missing link in estimating the platform's orientation. Unfortunately, the only IMU was inside the RTK unit used for ground-truthing in the available setup.

Chapter 6

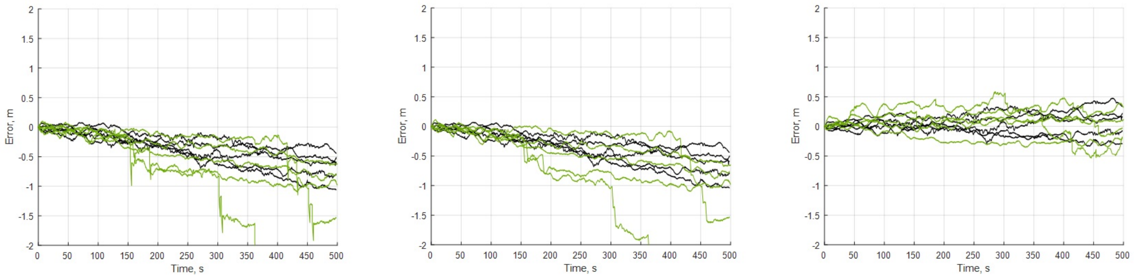
Results and Discussion

This chapter summarises and discusses all results from the perspective of the research question stated at the beginning. Section 6.1 lists the main contributions from each chapter. These are provided here for convenience before diving into the research question to which answer is provided in section 6.2 with an interesting discussion provided in section 6.3.

6.1 Thesis Summary

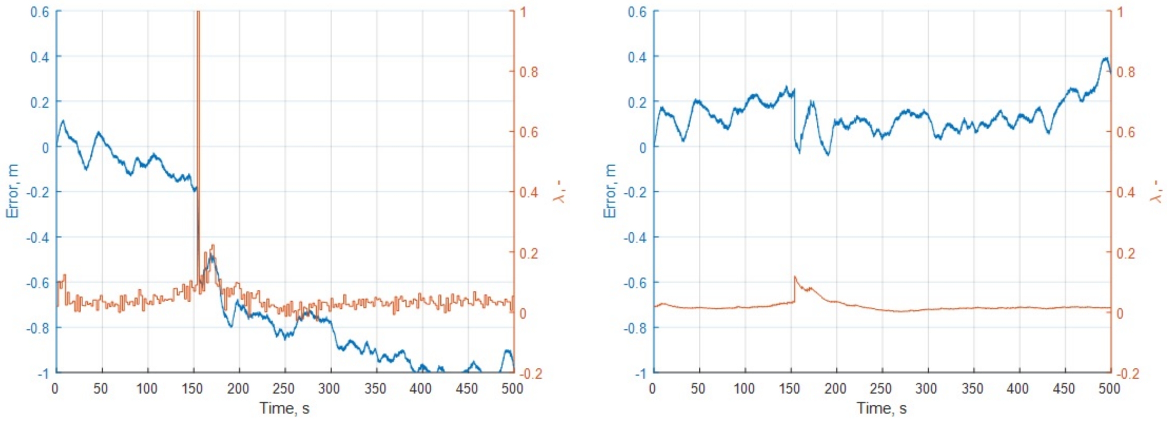
6.1.1 Model with Integrated Slip Estimation

In chapter 2, a model with Integrated Slip Estimation was defined, tuned, and analysed. Even though the model was verified only on two types of terrain, its performance was promising. Its



(a) Filter Free Estimation (b) Separate Slip Estimation (c) Integrated Slip Estimation

Figure 6.1: A comparison of errors in position estimation between different fusion models for T_{VO} = 2.0 s. Black are trajectories on the tarmac and green are trajectories on the grass. Note how the model with Integrated Slip Estimation provides lower errors for all trajectories.



(a) Pseudo-control u_λ from Separate Slip Estimation

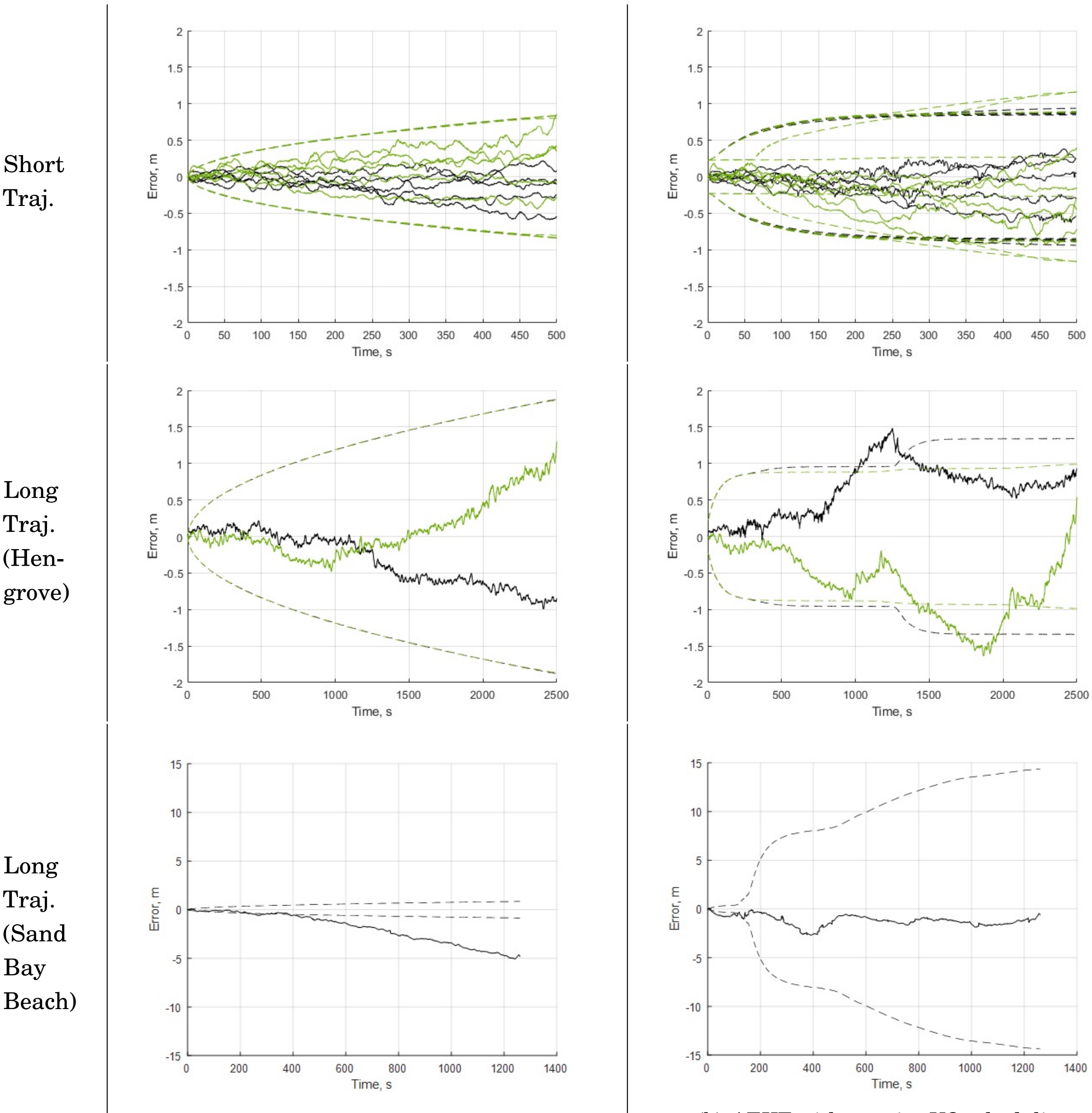
(b) λ from Integrated Slip Estimation

Figure 6.2: Single trajectory with overlaid error in position estimation (blue) and slip estimation (orange). Note how the system with Integrated Slip Estimation better filters wheel slip due to its low-pass filter. As a result, the error in position estimation is relatively small compared to the system with wheel slip treated as a pseudo-control input.

main advantage is modelling wheel slip as a low-pass filter driven by a process noise. The model not only gave better accuracy compared to other representative models (Fig. 6.1) when the VO update rate was relatively long, but it also offered the capability of filtering out incorrect VO measurements (Fig. 6.2).

6.1.2 Adaptive EKF and Reactive VO Scheduling

In chapter 3, the model was extended with the adaptive filter. Next, several trigger strategies were defined based on the outputs of the adaptive filter to determine when the subsequent VO measurements should happen. This reactive VO scheduling solution improved overall position estimation on all trajectories under investigation by scheduling new VO measurements where needed. This approach was tuned using several short trajectories (Fig. 6.3 top row) and later validated using long trajectories from two different datasets (Fig. 6.3 middle and bottom rows), including one where the rover drove at double its nominal speed (Fig. 6.3 bottom row). For the Sand Bay Beach case, a tradeoff between RMSE ans the number of VO measurements (Fig. 6.4) shows that the proposed solution maintains comparable localisation accuracy while using approximately seven times less VO measurements.



(a) EKF with fast VO

(b) AEKF with reactive VO scheduling

Figure 6.3: Errors in position estimation for EKF with fast VO acquisition (T_{VO} = 0.6 s for top two rows and 0.2 s for the bottom row) and AEKF with all reactive VO strategies ($T_{VO_{MAX}}$ = 3 s for top two rows and 2 s for the bottom row). Top two rows are using Hengrove dataset, hence black plots are for trajectories on the tarmac and green on the grass. Dashed lines represent 3σ confidence margin.

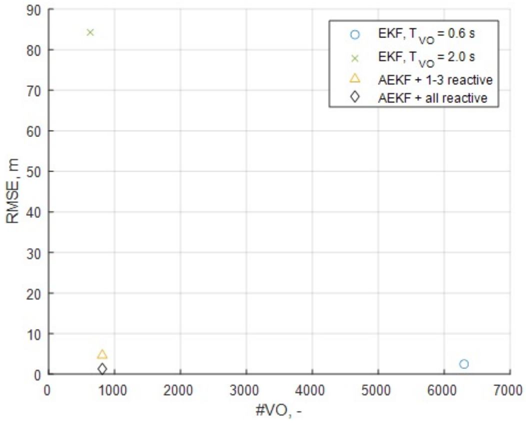


Figure 6.4: A tradeoff between RMSE and the number of VO measurements based on one long trajectory from the Sand Bay Beach dataset (double the nominal rover speed). An ideal solution would be confined to the bottom-left corner.

6.1.3 Wheel Slip Measurements and Prediction

Chapter 4 investigates how the information about the predicted wheel slip could be fused into the model for improved navigational accuracy. The model with Integrated Slip Estimation from chapter 2 is used as a baseline, with additional models defined to show various ways how the wheel slip can be treated. When using a slip profile in the model, an interesting observation is made when the model recognises slip values from the profile in measurements. It is indicated by a reduction of variance in position estimation which resembles SLAM (Fig. 6.5). It opens the possibility of using wheel slip as another source for position estimation.

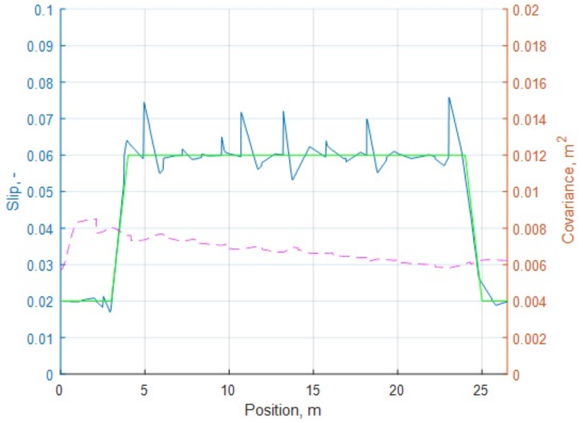
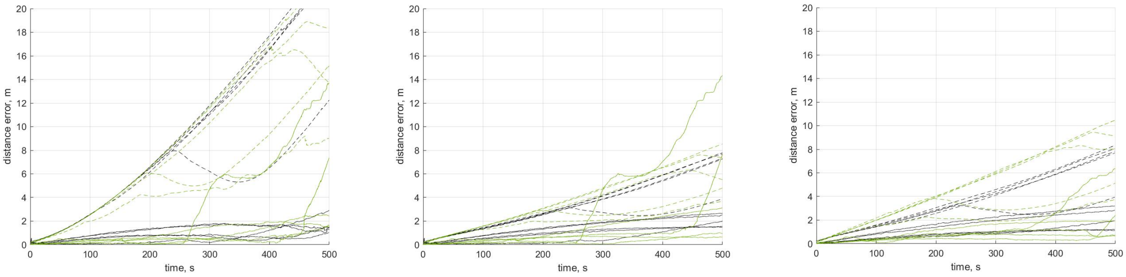


Figure 6.5: Slip profiles against the variance of position estimation for Model 3B with T_{VO} = 3 s: solid blue line – estimated wheel slip, solid green line – slip profile used for the simulated vision-based classifier, magenta dashed line – variance of the position estimation. Note a decrease in position estimation's variance.



(a) EKF (b) AEKF (c) AEKF + Reactive VO Scheduling

Figure 6.6: Distance errors for 2D reactive VO scheduling compared with EKF and AEKF for $T_{VO} = T_{VO_{MAX}} = 2.0$ s. Black plots are for trajectories on the tarmac and green are for trajectories on the grass. Dashed lines indicate 3σ confidence margin. Note how reactive solution provides better position estimation.

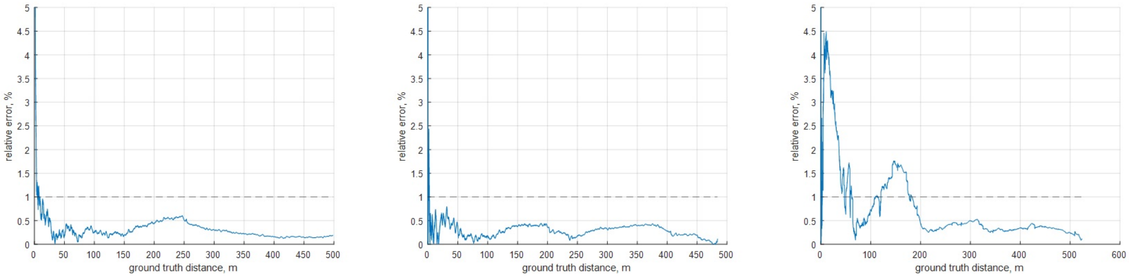
6.1.4 Motion in a Plane

All previous discussions were made for a one-dimensional problem. An initial attempt to introduce reactive VO scheduling, as defined in chapter 3, to a two-dimensional problem is presented in chapter 5. A simple model is defined, which utilises three different slip values: longitudinal, lateral, and angular. The system is evaluated using only one dataset from Hengrove Park. The results are promising given that all algorithms were used as-is (Fig. 6.6), however, there is room for improvement and further investigation. Suggestions in that matter are listed in the next section 7.2 but suffice to say that to achieve a good quality of results, more research in this area is required.

6.2 Can VO frequency be reduced without sacrificing navigational accuracy?

The answer to this question is provided solely on outputs from chapters 2 and 3. As indicated earlier, models from chapter 4 required a simulated slip profile, whereas chapter 5 provided only an initial attempt to use the reactive VO scheduling for two dimensions

Based on the tradeoff presented in chapter 3, adaptive EKF with reactive VO scheduling, on average, requires 4-7 times fewer VO measurements compared to EKF with VO triggered every 0.6 seconds. At the same time, the proposed solution led to nearly double the RMSE compared to EKF in Hengrove dataset and half the RMSE in the Sand Bay Beach dataset (whilst double the rover speed). In both cases, the error was often less than 1% of the distance travelled, which is often a minimum requirement. Fig. 6.7 shows the plots of the relative error. Aside from the beginning of each trajectory when the filter initialises, the relative error is usually below the 1% margin. Only in the Sand Bay Beach trajectory the error exceeds the margin at some point, but



(a) Hengrove, tarmac, 0.2 m/s (b) Hengrove, grass, 0.2 m/s (c) Sand Bay Beach, 0.4 m/s

Figure 6.7: Relative absolute error to ground truth distance for each long trajectory. Dashed lines mark 1% threshold.

even then, the error is minimised as the drive continues. With this in mind, we assumed adaptive EKF with reactive VO scheduling to be an accurate method for 1D localisation.

With the system capable of maintaining good navigational accuracy, reducing VO measurements could free enough computational resources to execute other algorithms, such as SLAM for localisation and mapping with loop closure or machine-learning-based techniques to advance the rover's terrain recognition capabilities further.

However, is it possible to go one step further and assess if the adaptive EKF with reactive VO can lead to reduced power usage for a rover?

6.3 Could AEKF with Reactive VO Lead to Reduced Power Usage for SFR?

To begin with, it is essential to note that because SFR development is still in an early stage, there is limited publicly-available documentation describing its systems and power budget. This section is based on two sources of information:

- the ExoMars rover – SFR aims to re-use as many ExoMars components and subsystems as possible. For example, the same Localisation/Navigation cameras will be used on SFR as on ExoMars.
- ESA's MarsFAST CDF Study Report [88] – even though it explains early concepts, some information is relevant.

SFR is expected to drive around 200-300 m per sol (Martian day). Given that solar panels will power it, its operations may be limited to only around noon. To cover a large distance in a short time, SFR will have to drive fast. Originally it was assumed to drive at 8 cm/s, compared to 1.1 cm/s for ExoMars. However, there are indications that the drive speed has been since reduced to 6.67 cm/s. This decision is to maintain a similar distance between consecutive stereo pairs for VO. In the case of ExoMars, VO will perform every 10 s [21]. With 1.1 cm/s that gives 11 cm separation. In the case of SFR, VO is expected to run every 2 s, which would lead to a 13.34 cm distance between VO measurements.

When looking at hardware power consumption, the following details were extracted:

- Stereo Camera – direct impact on VO
- LEON4 – co-processor responsible for VO estimation
- Overall locomotion and Guidance, Navigation, and Control (GNC) during traverse – baseline

Table 6.1 summarises power usage of listed items. The camera's peak power can be further confirmed on Neptec UK's website[1] (ExoMars camera manufacturer). However, there is no indication about power usage in a standby mode, nor is any information on how entering standby mode could further optimise power usage. The table shows that cameras and CPU power usage are negligible when traversing from the overall system perspective. It is also unknown if cameras can enter standby mode on-demand between image acquisition. Also, it is unknown if the CPU can use less power when not utilised. Performing operations on CPU is commonly known as "free resource", as it is always on, and there is no publicly available information to indicate whether Leon4 can reduce its power usage when not on full load. What is unknown at this stage is whether running VO requires a cooling system for the CPU. Mars's atmosphere may have different heat dissipation properties. Should active cooling be required, it could affect the overall power consumption.

After an informal discussion with some engineers, the following has been assumed:

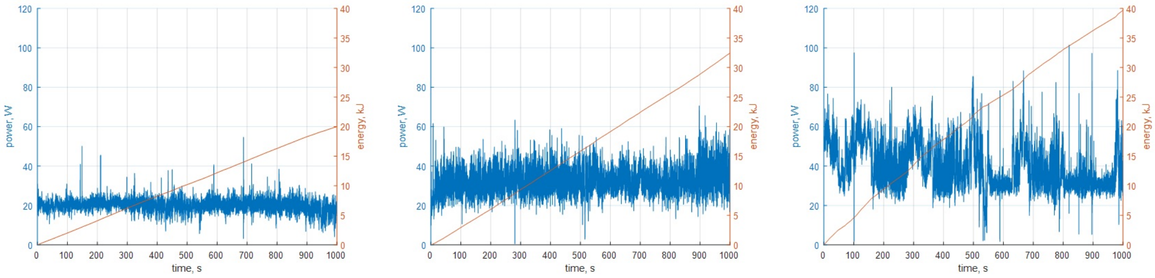
- The stereo camera operates all the time. There is a low-power mode, but it is only used at night. Entering the operational mode may require several onboard procedures like sensors warming up.
- CPU is powered all the time and most likely does not change power draw.
- At the time of writing this thesis, the power needed for SFR's traverse is estimated to be around 170 W, which is 2.5 times more than expected. Any saving in powering cameras would be negligible from the overall locomotion and GNC power perspective.

Table 6.1: Power usage for selected SFR subsystems

Item	Initial Power Usage	Revisited Power Usage
Localisation Camera Peak Power	< 5.0 W [89]	< 2.5 W [90]
Localisation Camera Stand-by Mode	< 1.5 W [89]	< 2.0 W [90]
LEON4 Co-Processor	N/A	< 3.0 W [91]
Overall Traverse	70.84 W [92], 79 W [88]	170 W

---

[1] https://neptec.com/missions/exomars-mars-mission/ - the website is no longer available as Neptec was acquired by MDA, which does not provide information about ExoMars cameras.



(a) Hengrove, tarmac, 0.2 m/s (b) Hengrove, grass, 0.2 m/s (c) Sand Bay Beach, 0.4 m/s

Figure 6.8: Total motors power and incremental energy for 1000 s long traverse section. The blue line indicates current power in W, the red line is incremental energy usage in kJ.

As indicated in Fig. 6.4, reactive VO can achieve comparable accuracy for fewer images than ExoMars and SFR baseline. However, the current avionics are not designed to take advantage of it as they are tailored to the existing workflow. There is a minimal difference in cameras' standard and standby power modes, and any savings there would not impact the overall power budget during the traverse. It is somehow confirmed in a separate study [30] where it was shown that for the same amount of energy, a representative rover traversed about 2380 m when using VO and vision-based maps for navigation compared to a 2700 m blind drive. In that work, the authors recognise that energy used for VO constitutes about 24.5% of the total addition (the rest goes to map building). It could be interpreted as VO costing the rover only about 78 m of the total distance (nearly 3% of the blind drive distance).

There are still two other possible advantages of using reactive VO. Firstly, it would free up the co-processor, allowing leveraging other compute-intense algorithms to be executed on the go. Based on the experiments, there are reasons to believe that the VO trigger time could be doubled to 4 s enabling other algorithms to use spare co-processor time for other tasks.

The second advantage of using reactive VO might be counter-intuitive. Because it allows for a greater distance between images, assuming constant SFR VO frame rate, higher rover speed could be achieved. It, hypothetically, could lead to some energy savings. Carefully designed experimentation would be needed to confirm or disprove it, but a crude example would be petrol cars with the best efficiency at a speed of 55-65 miles per hour. To back this hypothesis with results from the trials, Fig. 6.8 provides motors power and incremental energy usage for 1000 s long intervals on the tarmac, grass, and from the Sand Bay Beach where the rover drove at double speed. Total energy consumed during the traverses is listed in Table 6.2. Note that these power and energy values take into account only locomotion - they exclude sensors and computation.

At first glance, driving at double the speed may double the energy used. However, note that trajectory on the grass with the same speed as on tarmac uses 62% more energy. It shows that more power is needed to negotiate challenging terrain rather than drive faster. In Sand Bay Beach, the rover was subject to concrete, grass, and sand at double the speed, which, compared to

Table 6.2: Total energy usage after 1000 s drive

Terrain	Nominal Speed [m/s]	Total Energy Used [kJ]	Increase Compared to Tarmac
Tarmac	0.2	20.012	1
Grass	0.2	32.441	1.62
Mixed beach	0.4	39.682	1.98

the trajectory on grass, increased power output only by 22%. It is only one example and should not be treated as general proof of the hypothesis, but it indicates an area that may be further explored. Of course, much depends on the locomotion system. Even though the rover used in this project has a representative suspension and chassis, its motors may not be.

To summarise, reactive VO on its own may not lead to reduced power usage. It may only free computational resources. However, because the results suggest that accurate localisation is still achievable with less frequent VO measurement (longer distance between consecutive stereo frames), a rover may favour faster travel speed. Eventually, it may lead to reduced overall power output.

CHAPTER 7

CONCLUSIONS AND FURTHER RECOMMENDATIONS

This chapter concludes the thesis by providing a reflection upon the project objectives and how they were achieved. It is done in section 7.1, whereas section 7.2 lists recommendations for future work.

7.1 Conclusions

With upcoming missions to explore Mars and Moon, more emphasis is on long-range navigation. For rovers to achieve that, they need more powerful computing capabilities and bigger onboard power storage units. However, we propose to revisit how sensors are controlled and how their information is fused in the hope of freeing some computational resources and potentially leading to reduced power usage.

This thesis investigates ways of reducing the number of VO measurements while maintaining navigational accuracy for a rover by integrating wheel slip estimation into EKF and inspecting how the wheel slip can be predicted. The main focus of this work was to consider only one dimension (total distance travelled). A two-dimensional model was also explored and even though the results were promising, it suggests that more research in this area is still required.

The main objective of this project was to assess if VO frequency can be reduced without sacrificing navigational accuracy. As discussed in the previous chapter, it is possible. However, it is important to note that it is based on results from only two field trials. More testing would be highly recommended, but so far the outcome of this project is very encouraging. This problem is further discussed to evaluate if the solution can translate into the reduced power consumption of a rover. As highlighted, it very much depends on the avionics of the robot, however, a hypothetical idea is presented which claims that a robot may use less energy by driving faster. A reduced VO frequency for a rover driving at a nominal speed could be translated into a nominal VO frequency with a faster drive.

In the course of this project, models for wheel slip measurement prediction were also evaluated. Even though the results did not contribute directly to the main objective, the outcome highlighted a possibility of a slip-based SLAM system. It could further improve localisation accuracy, especially when driving faster.

Finally, the main objective, with a reduced scope (i.e. without actual analysis of the number of VO measurements used and utilising only one dataset) was also assessed in 2D motion in a plane. This work aimed to analyse if the proposed model can be scaled into higher dimensions. This has been done successfully, albeit more work is required to fully validate the solution.

7.2 Future Work Recommendations

This section lists three main 'leftovers' which could be pursued in the future by further strengthening the slip-aware localisation for wheeled vehicles.

1. Design a system for wheel slip prediction utilising image-based assessment of the terrain in front of a robot and relating it to wheels performance. One can imagine how the terrain is segmented into different types with associated tilt angles from depth estimation. This information could provide a more detailed prediction of wheel slip, which could be refined when the robot drives over that particular terrain type. By providing an in-depth analysis of the wheel's torque and power consumption, the system could learn to predict future wheel slip values better. A similar approach was presented in [77].

2. When analysing motion in a plane, the maximum time between VO measurements had to be reduced from 3 s to 2 s. We believe it was a consequence of not having enough image overlap during turns to correctly estimate the yaw angle. Including an IMU to provide orientation estimations in between VO measurements could lead to better overall performance. It would be interesting to include raw IMU measurements in EKF to estimate biases and angles and compare this approach with existing IMU filters that output unbiased orientation angles, such as the Madgwick filter [93].

3. A more in-depth analysis of adaptive filter for motion in a plane. In this project, there was one adaptive gain for the entire process noise, and one for each measurement noise. However, one could imagine having a scaling matrix where individual components scale individual noises (variances) and their cross-correlations. The same is applicable for VO and WO measurements where each degree of freedom could have separately-tuned gain for adaptive EKF.

APPENDIX A

DERIVATION OF FORMULAS FOR MODEL WITH INTEGRATED SLIP ESTIMATION

This annex provides equations for discretisation of $p(t)$, $\theta(t)$, $\Omega(t)$, and $\lambda(t)$ from the state vector defined in (2.7) as

$$x(t) = [p(t), \theta(t), \Omega(t), \lambda(t), m(t)]^T \tag{A.1}$$

States under consideration, based on Fig. 2.4, are defined in the continuous time as:

$$p(t) = \int_0^t \rho\Omega(\tau)\big(1-\lambda(\tau)\big)\,d\tau \tag{A.2a}$$

$$\theta(t) = \int_0^t \Omega(\tau)\,d\tau \tag{A.2b}$$

$$\Omega(t) = u_\Omega(t) + \int_0^t w_\alpha(\tau)\,d\tau, w_\alpha \sim N(\bar{w}_\alpha, \sigma_\alpha) \tag{A.2c}$$

$$\lambda(t) = K\big(1-e^{\frac{-t}{T_0}}\big)w_\beta(\tau), w_\beta \sim N(\bar{w}_\beta, \sigma_\beta) \tag{A.2d}$$

Note that the gain K for the low-pass filter in this work was set to one, however, for the completeness of this work and for the benefit of readers, an unknown value is assumed. Furthermore, it is assumed that

$$w_\alpha(k+1) = w_\alpha(k) \rightarrow \frac{dw_\alpha(\tau)}{d\tau} = 0 \tag{A.3a}$$

$$w_\beta(k+1) = w_\beta(k) \rightarrow \frac{dw_\beta(\tau)}{d\tau} = 0 \tag{A.3b}$$

Because (A.2a) contains non-linearity, it is the most difficult to discretise from all (A.2). Additionally, $p(k+1)$ requires other equations from (A.2) to have a known discrete form. As such, we begin with $\Omega(t)$ noting the assumption (A.3a). Its discrete form is

$$\begin{aligned}\Omega(k+1) &= u_\Omega(k+1) + \int_0^{k+1} w_\alpha(\tau)\,d\tau = u_\Omega(k+1) + \int_0^{k} w_\alpha(\tau)\,d\tau + \int_k^{k+1} w_\alpha(\tau)\,d\tau \\ &= u_\Omega(k+1) + \big(\Omega(k) - u_\Omega(k)\big) + \int_k^{k+1} w_\alpha(\tau)\,d\tau \\ &= \Omega(k) + u_\Omega(k+1) - u_\Omega(k) + \Delta t \frac{\bar{w}_\alpha(k) + \bar{w}_\alpha(k+1)}{2} \\ &= \Omega(k) + u_\Omega(k+1) - u_\Omega(k) + \Delta t \bar{w}_\alpha(k)\end{aligned} \tag{A.4}$$

Next we discretise $\theta(t)$ utilising (A.4)

$$\begin{aligned}\theta(k+1) &= \int_0^{k+1} \Omega(\tau)\,d\tau = \int_0^{k} \Omega(\tau)\,d\tau + \int_k^{k+1} \Omega(\tau)\,d\tau = \theta(k) + \int_k^{k+1} \Omega(\tau)\,d\tau \\ &= \theta(k) + \Delta t \frac{\Omega(k) + \Omega(k+1)}{2} = \theta(k) + \Delta t \frac{\Omega(k) + \Omega(k) + u_\Omega(k+1) - u_\Omega(k) + \Delta t \bar{w}_\alpha(k)}{2} \\ &= \theta(k) + \Delta t \Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \frac{\Delta t^2}{2} \bar{w}_\alpha(k)\end{aligned} \tag{A.5}$$

Because $\lambda(t)$ is an output from the low-pass filter, we use Z transform assuming Zero-Order Hold (ZOH) model. The transfer function of a low-pass filter in s-domain is given as

$$G(s) = \frac{K}{sT_0 + 1} \tag{A.6}$$

We then apply ZOH model and evaluate it using information from appropriate tables

$$G_{ZOH}(z) = (1 - z^{-1}) \mathcal{Z}\Big\{\frac{G(s)}{s}\Big\} = (1 - z^{-1}) \frac{K\Big(1 - e^{-\frac{\Delta t}{T_0}}\Big)}{(1 - z^{-1})\Big(z - e^{-\frac{\Delta t}{T_0}}\Big)} = \frac{K\Big(1 - e^{-\frac{\Delta t}{T_0}}\Big)}{\Big(z - e^{-\frac{\Delta t}{T_0}}\Big)} = \frac{\lambda(k)}{\bar{w}_\beta(k)} \tag{A.7}$$

Eq. (A.7) can be applied to evaluate $\lambda(k+1)$ noting that operator z indicates timestep at $k+1$

$$\lambda(k+1) = \lambda(k) e^{-\frac{\Delta t}{T_0}} + K\Big(1 - e^{-\frac{\Delta t}{T_0}}\Big) \bar{w}_\beta(k) \tag{A.8}$$

Finally, we discretise $p(t)$. It will be performed in several steps. The first step reuses some of equations from (A.5)

$$\begin{aligned}p(k+1) &= \rho \int_0^{k+1} \Omega(\tau)\big(1 - \lambda(\tau)\big)\,d\tau = \rho \int_0^{k} \Omega(\tau)\big(1 - \lambda(\tau)\big)\,d\tau + \rho \int_k^{k+1} \Omega(\tau)\big(1 - \lambda(\tau)\big)\,d\tau \\ &= p(k) + \rho \int_k^{k+1} \Omega(\tau)\big(1 - \lambda(\tau)\big)\,d\tau = p(k) + \rho \int_k^{k+1} \Omega(\tau)\,d\tau - \rho \int_k^{k+1} \Omega(\tau)\lambda(\tau)\,d\tau \\ &= p(k) + \rho \Delta t \Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \rho \frac{\Delta t^2}{2} \bar{w}_\alpha(k) - \rho \int_k^{k+1} \Omega(\tau)\lambda(\tau)\,d\tau\end{aligned} \tag{A.9}$$

Now, let us investigate the $\int_k^{k+1} \Omega(\tau)\lambda(\tau)\,d\tau$ element by using the integration by parts

$$\int_k^{k+1} \Omega(\tau)\lambda(\tau)\,d\tau = \int_k^{k+1} \frac{d\theta(\tau)}{d\tau} \lambda(\tau)\,d\tau = [\theta(\tau)\lambda(\tau)]_k^{k+1} - \int_k^{k+1} \theta(\tau) \frac{d\lambda(\tau)}{d\tau}\,d\tau \tag{A.10}$$

Next, we calculate $\frac{d\lambda(\tau)}{d\tau}$ noting (A.2d) and (A.3b)

$$\frac{d\lambda(\tau)}{d\tau} = K\frac{dw_\beta(\tau)}{d\tau} - K\frac{dw_\beta(\tau)e^{\frac{-\tau}{T_0}}}{d\tau} = -K\frac{dw_\beta(\tau)e^{\frac{-\tau}{T_0}}}{d\tau} = -Kw_\beta(\tau)(-\frac{1}{T_0})e^{\frac{-\tau}{T_0}} = \frac{K}{T_0}w_\beta(\tau)e^{\frac{-\tau}{T_0}} \tag{A.11}$$

With (A.11) defined, it is possible to evaluate the last element (integral) from (A.10). Additionally, because $\theta(\tau)$ is the total wheels angle, we can assume that this function is non-decreasing, therefore, we can apply Bonnet's Theorem

$$\int_k^{k+1} \theta(\tau)\frac{d\lambda(\tau)}{d\tau}\,d\tau = \frac{K}{T_0}\int_k^{k+1} \theta(\tau)w_\beta(\tau)e^{\frac{-\tau}{T_0}}\,d\tau = \frac{K}{T_0}\theta(k+1)\int_k^{k+1} w_\beta(\tau)e^{\frac{-\tau}{T_0}}\,d\tau \tag{A.12}$$

Let us again evaluate the last integral from (A.12) using the integration by parts and noting the assumption (A.3b)

$$\begin{aligned}\int_k^{k+1} w_\beta(\tau)e^{\frac{-\tau}{T_0}}\,d\tau &= [-T_0w_\beta(\tau)e^{\frac{-\tau}{T_0}}]_k^{k+1} - \int_k^{k+1}\frac{dw_\beta(\tau)}{d\tau}e^{\frac{-\tau}{T_0}}\,d\tau = T_0w_\beta(k)[-e^{\frac{-\tau}{T_0}}]_k^{k+1}\\ &= T_0w_\beta(k)\Big(e^{\frac{-k\Delta t}{T_0}} - e^{\frac{-(k+1)\Delta t}{T_0}}\Big) = T_0w_\beta(k)e^{\frac{-k\Delta t}{T_0}}\Big(1-e^{\frac{-\Delta t}{T_0}}\Big)\end{aligned} \tag{A.13}$$

Now we insert (A.13) into (A.12)

$$\begin{aligned}\int_k^{k+1} \theta(\tau)\frac{d\lambda(\tau)}{d\tau}\,d\tau &= \frac{K}{T_0}\theta(k+1)\int_k^{k+1} w_\beta(\tau)e^{\frac{-\tau}{T_0}}\,d\tau\\ &= \frac{K}{T_0}\theta(k+1)T_0w_\beta(k)e^{\frac{-k\Delta t}{T_0}}\Big(1-e^{\frac{-\Delta t}{T_0}}\Big) = K\theta(k+1)w_\beta(k)e^{\frac{-k\Delta t}{T_0}}\Big(1-e^{\frac{-\Delta t}{T_0}}\Big)\end{aligned} \tag{A.14}$$

Now we insert (A.14) to (A.10) noting (A.8) and (A.5)

$$\begin{aligned}\int_k^{k+1} \Omega(\tau)\lambda(\tau)\,d\tau &= [\theta(\tau)\lambda(\tau)]_k^{k+1} - \int_k^{k+1}\theta(\tau)\frac{d\lambda(\tau)}{d\tau}\,d\tau\\ &= \theta(k+1)\lambda(k+1) - \theta(k)\lambda(k) - K\theta(k+1)w_\beta(k)e^{\frac{-k\Delta t}{T_0}}\Big(1-e^{\frac{-\Delta t}{T_0}}\Big)\\ &= \theta(k+1)\lambda(k)e^{-\frac{\Delta t}{T_0}} + \theta(k+1)K\Big(1-e^{-\frac{\Delta t}{T_0}}\Big)\bar{w}_\beta(k)\\ &\quad - \theta(k)\lambda(k) - K\theta(k+1)w_\beta(k)e^{\frac{-k\Delta t}{T_0}}\Big(1-e^{\frac{-\Delta t}{T_0}}\Big)\\ &= \theta(k+1)\lambda(k)e^{-\frac{\Delta t}{T_0}} - \theta(k)\lambda(k) + \theta(k+1)Kw_\beta(k)\Big(1-e^{\frac{-\Delta t}{T_0}}\Big)\Big(1-e^{\frac{-k\Delta t}{T_0}}\Big)\\ &= \theta(k+1)\lambda(k)e^{-\frac{\Delta t}{T_0}} - \theta(k)\lambda(k) + \theta(k+1)\Big(1-e^{\frac{-\Delta t}{T_0}}\Big)\lambda(k)\\ &= \theta(k+1)\lambda(k)e^{-\frac{\Delta t}{T_0}} - \theta(k)\lambda(k) + \theta(k+1)\lambda(k) - \theta(k+1)e^{\frac{-\Delta t}{T_0}}\lambda(k)\\ &= \Big(\theta(k+1)-\theta(k)\Big)\lambda(k)\\ &= \Big(\theta(k) + \Delta t\Big(\Omega(k) + \frac{u_\Omega(k+1)-u_\Omega(k)}{2}\Big) + \frac{\Delta t^2}{2}\bar{w}_\alpha(k) - \theta(k)\Big)\lambda(k)\\ &= \Delta t\Big(\Omega(k) + \frac{u_\Omega(k+1)-u_\Omega(k)}{2}\Big)\lambda(k) + \frac{\Delta t^2}{2}\bar{w}_\alpha(k)\lambda(k)\end{aligned} \tag{A.15}$$

Finally, insert (A.15) to (A.9)

$$\begin{aligned}
p(k+1) &= p(k) + \rho\Delta t\Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \rho\frac{\Delta t^2}{2}\bar{w}_\alpha(k) - \rho\int_k^{k+1}\Omega(\tau)\lambda(\tau)d\tau \\
&= p(k) + \rho\Delta t\Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \rho\frac{\Delta t^2}{2}\bar{w}_\alpha(k) \\
&- \rho\Delta t\Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big)\lambda(k) - \rho\frac{\Delta t^2}{2}\bar{w}_\alpha(k)\lambda(k) \\
&= p(k) + \rho\Delta t\Big(1 - \lambda(k)\Big)\Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \rho\frac{\Delta t^2}{2}\Big(1 - \lambda(k)\Big)\bar{w}_\alpha(k)
\end{aligned} \tag{A.16}$$

To summarise, the discrete form of the state variables is defined as

$$\begin{aligned}
p(k+1) &= p(k) + \rho\Delta t\Big(1 - \lambda(k)\Big)\Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \rho\frac{\Delta t^2}{2}\Big(1 - \lambda(k)\Big)\bar{w}_\alpha(k) \\
\theta(k+1) &= \theta(k) + \Delta t\Big(\Omega(k) + \frac{u_\Omega(k+1) - u_\Omega(k)}{2}\Big) + \frac{\Delta t^2}{2}\bar{w}_\alpha(k) \\
\Omega(k+1) &= \Omega(k) + u_\Omega(k+1) - u_\Omega(k) + \Delta t\bar{w}_\alpha(k) \\
\lambda(k+1) &= \lambda(k)e^{-\frac{\Delta t}{T_0}} + K\Big(1 - e^{-\frac{\Delta t}{T_0}}\Big)\bar{w}_\beta(k)
\end{aligned} \tag{A.17}$$

BIBLIOGRAPHY

[1] NASA, “Mars 2020 Perseverance Rover.”
Available at https://mars.nasa.gov/mars2020/.

[2] CNSA, “Probe makes historic landing on Mars,” May 2021.
Available at http://www.cnsa.gov.cn/english/n6465652/n6465653/c6812005/content.html.

[3] ESA, “ExoMars suspended,” Mar. 2022.
ilable at https://www.esa.int/Newsroom/Press_Releases/ExoMars_suspended.

[4] B. K. Muirhead, A. Nicholas, and J. Umland, “Mars Sample Return Mission Concept Status,” in *2020 IEEE Aerospace Conference*, 2020.

[5] NASA, “NASA’s Lunar Exploration Program Overview,” Sept. 2020.
Available at https://www.nasa.gov/sites/default/files/atoms/files/artemis_plan-20200921.pdf.

[6] L. Duvet, F. Beyer, J. Delfa, and E. Zekri, “ESA Sample Fetch Rover: Heritage and Way Forward,” vol. 2071, (Berlin, Germany), Apr. 2018.

[7] M. D. Benedetti, M. Azkarate, and I. Mario, “PARAMETRIC PERFORMANCE CHARACTERIZATION OF VISUAL ODOMETRY FOR THE SAMPLE FETCH ROVER,” in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, p. 8, Oct. 2020.

[8] J. P. Grotzinger, J. Crisp, A. R. Vasavada, R. C. Anderson, C. J. Baker, R. Barry, D. F. Blake, P. Conrad, K. S. Edgett, B. Ferdowski, R. Gellert, J. B. Gilbert, M. Golombek, J. Gómez-Elvira, D. M. Hassler, L. Jandura, M. Litvak, P. Mahaffy, J. Maki, M. Meyer, M. C. Malin, I. Mitrofanov, J. J. Simmonds, D. Vaniman, R. V. Welch, and R. C. Wiens, “Mars Science Laboratory Mission and Science Investigation,” *Space Science Reviews*, vol. 170, pp. 5–56, Sept. 2012.

[9] NASA, “First Multiple-Sol Drive,” Feb. 2022.
Available at https://mars.nasa.gov/mars2020/mission/status/367/first-multiple-sol-drive/.

[10] M. Maimone, Y. Cheng, and L. Matthies, “Two years of Visual Odometry on the Mars Exploration Rovers,” *Journal of Field Robotics*, vol. 24, no. 3, pp. 169–186, 2007.

[11] W. Churchill, *Experience Based Navigation: Theory,Practice and Implementation*. PhD thesis, University of Oxford, Oct. 2012.

[12] S. Karachalios, M. Woods, S. Schwenzer, and L. Joudrier, “NOVELTY OR ANOMALY HUNTER: TOWARDS FLIGHT READY AUTONOMOUS SCIENCE USING STATE OF THE ART MACHINE & DEEP LEARNING,” in *Advanced Space Technologies in Robotics and Automation 2019*, (Noordwijk, The Netherlands), p. 7, May 2019.

[13] DARPA, “DARPA Subterranean Challenge,” 2021. Available at https://www.subtchallenge.com/.

[14] S. Khattak, H. Nguyen, F. Mascarich, T. Dang, and K. Alexis, “Complementary Multi–Modal Sensor Fusion for Resilient Robot Pose Estimation in Subterranean Environments,” in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1024–1029, Sept. 2020. ISSN: 2575-7296.

[15] M. Bloesch, M. Burri, H. Sommer, R. Siegwart, and M. Hutter, “The Two-State Implicit Filter Recursive Estimation for Mobile Robots,” *IEEE Robotics and Automation Letters*, vol. 3, pp. 573–580, Jan. 2018. Conference Name: IEEE Robotics and Automation Letters.

[16] M. Tranzatto, F. Mascarich, L. Bernreiter, C. Godinho, M. Camurri, S. Khattak, T. Dang, V. Reijgwart, J. Loeje, D. Wisth, S. Zimmermann, H. Nguyen, M. Fehr, L. Solanka, R. Buchanan, M. Bjelonic, N. Khedekar, M. Valceschini, F. Jenelten, M. Dharmadhikari, T. Homberger, P. De Petris, L. Wellhausen, M. Kulkarni, T. Miki, S. Hirsch, M. Montenegro, C. Papachristos, F. Tresoldi, J. Carius, G. Valsecchi, J. Lee, K. Meyer, X. Wu, J. Nieto, A. Smith, M. Hutter, R. Siegwart, M. Mueller, M. Fallon, and K. Alexis, “CERBERUS: Autonomous Legged and Aerial Robotic Exploration in the Tunnel and Urban Circuits of the DARPA Subterranean Challenge,” Tech. Rep. arXiv:2201.07067, arXiv, Jan. 2022. arXiv:2201.07067 [cs] type: article.

[17] F. Jenelten, J. Hwangbo, F. Tresoldi, C. D. Bellicoso, and M. Hutter, “Dynamic Locomotion on Slippery Ground,” *IEEE Robotics and Automation Letters*, vol. 4, pp. 4170–4176, Oct. 2019. Conference Name: IEEE Robotics and Automation Letters.

[18] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, D. Heřt, M. Petrlík, T. Báča, V. Spurný, F. Pomerleau, V. Kubelka, J. Faigl, K. Zimmermann, M. Saska, T. Svoboda, and T. Krajník, “DARPA Subterranean Challenge: Multi-robotic Exploration of Underground Environments,” in *Modelling and Simulation for Autonomous Systems*

(J. Mazal, A. Fagiolini, and P. Vasik, eds.), Lecture Notes in Computer Science, (Cham), pp. 274–290, Springer International Publishing, 2020.

[19] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, T. Azayev, D. Heřt, M. Petrlík, T. Báča, V. Spurný, V. Krátký, P. Petráček, D. Baril, M. Vaidis, V. Kubelka, F. Pomerleau, J. Faigl, K. Zimmermann, M. Saska, T. Svoboda, and T. Krajník, "System for multi-robotic exploration of underground environments CTU-CRAS-NORLAB in the DARPA Subterranean Challenge," Tech. Rep. arXiv:2110.05911, arXiv, Oct. 2021.
arXiv:2110.05911 [cs] type: article.

[20] M. Palieri, B. Morrell, A. Thakur, K. Ebadi, J. Nash, A. Chatterjee, C. Kanellakis, L. Carlone, C. Guaragnella, and A. Agha-mohammadi, "LOCUS: A Multi-Sensor Lidar-Centric Solution for High-Precision Odometry and 3D Mapping in Real-Time," *IEEE Robotics and Automation Letters*, vol. 6, pp. 421–428, Apr. 2021.
arXiv:2012.14447 [cs].

[21] L. Bora, R. Lancaster, B. Nye, C. Barclay, S. Rubio, and M. Winter, "EXOMARS ROVER CONTROL, LOCALISATION AND PATH PLANNING IN A HAZARDOUS AND HIGH DISTURBANCE ENVIRONMENT," p. 8, 2017.

[22] D. Helmick, Y. Cheng, D. Clouse, M. Bajracharya, L. Matthies, and S. Roumeliotis, "Slip compensation for a Mars rover," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005.

[23] S. Roumeliotis and J. Burdick, "Stochastic cloning: a generalized framework for processing relative state measurements," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, 2002.

[24] P. F. Muir and C. P. Neuman, "Kinematic modeling of wheeled mobile robots," *Journal of Robotic Systems*, vol. 4, pp. 281–340, Apr. 1987.

[25] K. Bussmann, L. Meyer, F. Steidle, and A. Wedler, "Slip Modeling and Estimation for a Planetary Exploration Rover: Experimental Results from Mt. Etna," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[26] A. Sakai, Y. Tamura, and Y. Kuroda, "An efficient solution to 6DOF localization using Unscented Kalman Filter for planetary rovers," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4154–4159, Oct. 2009.
ISSN: 2153-0866.

[27] K. Berntorp, "Joint Wheel-Slip and Vehicle-Motion Estimation Based on Inertial, GPS, and Wheel-Speed Sensors," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 1020–1027, May 2016.

Conference Name: IEEE Transactions on Control Systems Technology.

[28] V. X. Ha, C. Ha, and J. Lee, “Dual Estimation for the Path Tracking Control of a Four Track Wheel Skid-Steered Mobile Robot,” in *AIAA Guidance, Navigation, and Control (GNC) Conference*, (Boston, MA), American Institute of Aeronautics and Astronautics, Aug. 2013.

[29] S. Higa, Y. Iwashita, K. Otsu, M. Ono, O. Lamarre, A. Didier, and M. Hoffmann, “Vision-Based Estimation of Driving Energy for Planetary Rovers Using Deep Learning and Terramechanics,” *IEEE Robotics and Automation Letters*, vol. 4, pp. 3876–3883, Oct. 2019.
Conference Name: IEEE Robotics and Automation Letters.

[30] M. Woods, A. Shaw, I. Wallace, and M. Malinowski, “The Chameleon field trial: toward efficient, terrain sensitive navigation,” in *Advanced Space Technologies in Robotics and Automation 2015*, Dec. 2015.

[31] NASA, “Rover Brains.”
Available at https://mars.nasa.gov/mars2020/spacecraft/rover/brains/.

[32] NASA, “Video Friday: Perseverance Autonomy,” Apr. 2022.
Available at https://spectrum.ieee.org/video-friday-perseverance-autonomy.

[33] D. Townson, M. Woods, and S. Carnochan, “EXOMARS VISLOC – THE INDUSTRIALISED, VISUAL LOCALISATION SYSTEM FOR THE EXOMARS ROVER,” in *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, June 2018.

[34] M. Winter, S. Rubio, R. Lancaster, C. Barclay, N. Silva, B. Nye, and L. Bora, “DETAILED DESCRIPTION OF THE HIGH-LEVEL AUTONOMY FUNCTIONALITIES DEVELOPED FOR THE EXOMARS ROVER,” p. 8, 2017.

[35] C. Gaisler, “GR740 Quad-Processor LEON4FT System-on-Chip Overview,” Sept. 2018.
Available at https://gaisler.com/doc/gr740/GR740-OVERVIEW.pdf.

[36] C. Wang, Y. Liu, K. Zuo, J. Tong, Y. Ding, and P. Ren, “ac2SLAM: FPGA Accelerated High-Accuracy SLAM with Heapsort and Parallel Keypoint Extractor,” in *2021 International Conference on Field-Programmable Technology (ICFPT)*, pp. 1–9, Dec. 2021.

[37] S. Sabogal, A. George, and G. Crum, “ReCoN: A Reconfigurable CNN Acceleration Framework for Hybrid Semantic Segmentation on Hybrid SoCs for Space Applications,” in *2019 IEEE Space Computing Conference (SCC)*, pp. 41–52, July 2019.

[38] M. Avilés, M. Lourakis, G. Lentaris, I. Stamoulias, K. Maragos, D. Mora, and D. Soudris, “SPARTAN: VISION-BASED AUTONOMOUS NAVIGATION SYSTEM FOR FAST

TRAVERSAL PLANETARY ROVERS," in *International Symposium on Artificial Intelligence, Robotics, and Automation in Space*, p. 7, 2018.

[39] ESA, "ESA team blasts Intel's new AI chip with radiation at CERN," Nov. 2018.
Available at https://www.esa.int/Enabling_Support/Space_Engineering_Technology/ESA_team_blasts_Intel_s_ne

[40] ESA, "PhiSat-1 - Satellite Missions - eoPortal Directory."
Available at https://directory.eoportal.org/web/eoportal/satellite-missions/p/phisat-1.

[41] C. Adams, A. Spain, J. Parker, M. Hevert, J. Roach, and D. Cotten, "Towards an Integrated GPU Accelerated SoC as a Flight Computer for Small Satellites," in *2019 IEEE Aerospace Conference*, pp. 1–7, Mar. 2019.
ISSN: 1095-323X.

[42] W. S. Slater, N. P. Tiwari, T. M. Lovelly, and J. K. Mee, "Total Ionizing Dose Radiation Testing of NVIDIA Jetson Nano GPUs," in *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, pp. 1–3, Sept. 2020.
ISSN: 2643-1971.

[43] J. Delaune, R. Brockers, D. S. Bayard, H. Dor, R. Hewitt, J. Sawoniewicz, G. Kubiak, T. Tzanetos, L. Matthies, and J. Balaram, "Extended Navigation Capabilities for a Future Mars Science Helicopter Concept," in *2020 IEEE Aerospace Conference*, pp. 1–10, Mar. 2020.
ISSN: 1095-323X.

[44] R. Worley, Y. Yu, and S. Anderson, "Acoustic Echo-Localization for Pipe Inspection Robots," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, 2020.

[45] "Thames Tideway Tunnel – Benchmarking and Risk Assessment of Alternative Drive Strategies," Nov. 2013.

[46] "Hammersmith Flyunder Tunnel Feasibility Study - Tunnel and Geotechnical Engineering," Mar. 2014.

[47] "Silvertown Tunnel Preliminary Environmental Information Report Appendix 16.D River Wall Structural Condition Survey," 2015.

[48] R. F. Stengel, *Optimal Control and Estimation*.
Courier Corporation, Sept. 1994.
Google-Books-ID: byRgDwAAQBAJ.

[49] M. Malinowski, A. Richards, and M. Woods, "Power-aware Fusion of Visual and Wheel Odometry for Mobile Platforms," in *UKRAS20 Conference: "Robots into the real world" Proceedings*, pp. 111–113, EPSRC UK-RAS Network, May 2020.

[50] M. T. Malinowski, A. Richards, and M. Woods, “Fusion of Visual and Wheel Odometry with Integrated Slip Estimation,” in *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, Jan. 2021.

[51] M. T. Malinowski, A. Richards, and M. Woods, “Reactive Visual Odometry Scheduling Based on Noise Analysis using an Adaptive Extended Kalman Filter,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 294–299, Sept. 2021.
ISSN: 2153-0866.

[52] M. T. Malinowski, A. Richards, and M. Woods, “Wheel Slip Prediction for Improved Rover Localization,” in *AIAA SCITECH 2022 Forum*, American Institute of Aeronautics and Astronautics, Jan. 2022.
_eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2022-1080.

[53] D. Helmick, Y. Cheng, D. Clouse, L. Matthies, and S. Roumeliotis, “Path following using visual odometry for a Mars rover in high-slip environments,” in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, vol. 2, pp. 772–789 Vol.2, Mar. 2004.
ISSN: 1095-323X.

[54] J. Kim and J. Lee, “A kinematic-based rough terrain control for traction and energy saving of an exploration rover,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3595–3600, Oct. 2016.
ISSN: 2153-0866.

[55] A. Angelova, L. Matthies, D. Helmick, G. Sibley, and P. Perona, “Learning to predict slip for ground robots,” in *Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA)*, 2006.

[56] O. Toupet, J. Biesiadecki, A. Rankin, A. Steffy, G. Meirion-Griffith, D. Levine, M. Schadegg, and M. Maimone, “Terrain-adaptive wheel speed control on the Curiosity Mars rover: Algorithm and flight results,” *Journal of Field Robotics*, vol. 37, no. 5, pp. 699–728, 2020.

[57] Jayoung Kim and Jihong Lee, “Intelligent slip-optimization control with traction-energy trade-off for wheeled robots on rough terrain,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014.

[58] P. Ondrúška, C. Gurău, L. Marchegiani, Chi Hay Tong, and I. Posner, “Scheduled perception for energy-efficient path following,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[59] M. Lahijanian, M. Svorenova, A. A. Morye, B. Yeomans, D. Rao, I. Posner, P. Newman, H. Kress-Gazit, and M. Kwiatkowska, “Resource-Performance Tradeoff Analysis for

Mobile Robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1840–1847, 2018.

[60] Y. Hu and T. Furukawa, "Degenerate Near-Planar 3D Reconstruction from Two Overlapped Images for Road Defects Detection," *Sensors*, vol. 20, no. 6, 2020.

[61] D. Woodbury and J. Junkins, "On the Consider Kalman Filter," in *AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, Aug. 2010.

[62] R. Zanetti and R. Bishop, "Kalman Filters with Uncompensated Biases," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 1, pp. 327–335, 2012.

[63] R. Mehra, "Approaches to adaptive filtering," *IEEE Transactions on Automatic Control*, vol. 17, pp. 693–698, Oct. 1972.
Conference Name: IEEE Transactions on Automatic Control.

[64] J. Wang, "Stochastic Modeling for Real-Time Kinematic GPS/GLONASS Positioning," *NAVIGATION*, 1999.

[65] K. Myers and B. Tapley, "Adaptive sequential estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 21, no. 4, pp. 520–523, 1976.

[66] T. J. Klein, M. A. Lewis, J. Jeka, and T. Kiemel, "Postural control in a bipedal robot using sensory reweighting," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2053–2058, May 2011.
ISSN: 1050-4729.

[67] Y. Zhang, H. Chen, W. Xu, T. C. Yang, and J. Huang, "Spatiotemporal Tracking of Ocean Current Field With Distributed Acoustic Sensor Network," *IEEE Journal of Oceanic Engineering*, vol. 42, pp. 681–696, July 2017.
Conference Name: IEEE Journal of Oceanic Engineering.

[68] A. P. Sage and G. W. Husa, "Algorithms for sequential adaptive estimation of prior statistics," in *1969 IEEE Symposium on Adaptive Processes (8th) Decision and Control*, 1969.

[69] A. P. Sage and G. W. Husa, "Adaptive filtering with unknown prior statistics," *Joint Automatic Control Conference*, 1969.

[70] W. Yan, Q. Ding, J. Chen, Y. Liu, and S. S. Cheng, "Needle Tip Tracking in 2D Ultrasound Based on Improved Compressive Tracking and Adaptive Kalman Filter," *IEEE Robotics and Automation Letters*, vol. 6, pp. 3224–3231, Apr. 2021.
Conference Name: IEEE Robotics and Automation Letters.

[71] S. Zhang, “An Adaptive Unscented Kalman Filter for Dead Reckoning Systems,” in *2009 International Conference on Information Engineering and Computer Science*, pp. 1–4, Dec. 2009.
ISSN: 2156-7387.

[72] “Reprint of: Mahalanobis, P.C. (1936) "On the Generalised Distance in Statistics.",” *Sankhya A*, Dec. 2018.

[73] D. Helmick, A. Angelova, and L. Matthies, “Terrain Adaptive Navigation for planetary rovers,” *Journal of Field Robotics*, vol. 26, no. 4, pp. 391–410, 2009.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.20292.

[74] C. Cunningham, M. Ono, I. Nesnas, J. Yen, and W. L. Whittaker, “Locally-adaptive slip prediction for planetary rovers using Gaussian processes,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5487–5494, May 2017.

[75] M. Endo, S. Endo, K. Nagaoka, and K. Yoshida, “Terrain-Dependent Slip Risk Prediction for Planetary Exploration Rovers,” *Robotica*, pp. 1–14, 2021.
Publisher: Cambridge University Press.

[76] K. Skonieczny, D. K. Shukla, M. Faragalli, M. Cole, and K. D. Iagnemma, “Data-driven mobility risk prediction for planetary rovers,” *Journal of Field Robotics*, vol. 36, no. 2, pp. 475–491, 2019.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21833.

[77] J. Schwendner, S. Joyeux, and F. Kirchner, “Using Embodied Data for Localization and Mapping,” *Journal of Field Robotics*, vol. 31, no. 2, pp. 263–295, 2014.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21489.

[78] L. Ojeda, D. Cruz, G. Reina, and J. Borenstein, “Current-Based Slippage Detection and Odometry Correction for Mobile Robots and Planetary Rovers,” *IEEE Transactions on Robotics*, vol. 22, pp. 366–378, Apr. 2006.

[79] M. Dissanayake, P. Newman, S. Clark, H. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 229–241, June 2001.
Conference Name: IEEE Transactions on Robotics and Automation.

[80] R. E. Zelenka, “Design and analysis of a Kalman filter for terrain-referenced positioning and guidance,” *Journal of Aircraft*, vol. 31, pp. 339–344, Mar. 1994.

[81] C. A. Brooks and K. Iagnemma, “Self-supervised terrain classification for planetary surface exploration rovers,” *Journal of Field Robotics*, vol. 29, no. 3, pp. 445–468, 2012.
_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21408.

[82] J. Y. Wong, *Theory of Ground Vehicles, 4th Edition*.
John Wiley & Sons, Aug. 2008.

[83] L. Marín, M. Vallés, A. Soriano, A. Valera, and P. Albertos, "Event-based localization in ackermann steering limited resource mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 19, no. 4, pp. 1171–1182, 2014.

[84] A. Albagul and Wahyudi, "Dynamic modeling and adaptive traction control for mobile robots," in *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, vol. 1, pp. 614–620 Vol. 1, Nov. 2004.

[85] C. C. Ward and K. Iagnemma, "A Dynamic-Model-Based Wheel Slip Detector for Mobile Robots on Outdoor Terrain," *IEEE Transactions on Robotics*, vol. 24, pp. 821–831, Aug. 2008.
Conference Name: IEEE Transactions on Robotics.

[86] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Commun. ACM*, vol. 24, pp. 381–395, June 1981.
Place: New York, NY, USA Publisher: Association for Computing Machinery.

[87] P. Torr and A. Zisserman, "Robust computation and parametrization of multiple view relations," in *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pp. 727–732, Jan. 1998.

[88] "MarsFAST CDF Study Report," Oct. 2014.

[89] E. Astrium, "ExoMars Rover Vehicle: Rover Vehicle Navigation Cameras Requirements Specification," 2010.

[90] M. Winter, C. Barclay, V. Pereira, R. Lancaster, M. Caceres, K. McManamon, N. Silva, D. Lachat, and M. Campana, "EXOMARS ROVER VEHICLE: DETAILED DESCRIPTION OF THE GNC SYSTEM," p. 8, 2015.

[91] J. Andersson, "Advances in European Processor Technology," Dec. 2019.

[92] A. Merlo, J. Larranaga, and P. Falkner, "SAMPLE FETCHING ROVER (SFR) FOR MSR," in *Advanced Space Technologies in Robotics and Automation 2013*, May 2013.

[93] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, pp. 1–7, June 2011.
ISSN: 1945-7901.