Full State History Cooperative Localisation with Complete Information Sharing

Lachlan Toohey

A thesis submitted in fulfillment
of the requirements of the degree of
Doctor of Philosophy

THE UNIVERSITY OF
SYDNEY

Australian Centre for Field Robotics
School of Aerospace, Mechanical and Mechatronic Engineering
The University of Sydney

February 2016

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the University or other institute of higher learning, except where due acknowledgement has been made in the text.

Lachlan Toohey

29 February 2016

Abstract

Lachlan Toohey
The University of Sydney

Doctor of Philosophy
February 2016

Full State History Cooperative Localisation with Complete Information Sharing

This thesis presents the development of a novel decentralised localisation method for multiple robots based on techniques from state of the art single vehicle localisation methods. We extend upon these methods and enable reduced bandwidth requirements whilst employing local solutions using state of the art single vehicle algorithms. Unlike many other state of the art approaches, this method does not enforce a specific communication topology or require complex tracking of fused information to avoid double counting and inconsistencies in the estimate. The methods for inclusion of shared data match basic methods for graph optimisation enabling quick implementation without special cases and giving greater flexibility of application.

There are four key contributions in this thesis. The first is a novel method for splitting the multiple vehicle problem into sections that can be iteratively transmitted in independent packets with an upper bound on bandwidth per inter robot observation based on the dimension of the state space. This is achieved through delayed elimination of states involved in intervehicle observations that we name external states. Local observations are placed in subgraphs, each of which contains all states from one intervehicle observation until the next. Any observations that connect to an external state are placed in either the subgraph before or after the target external state - but not both. Internal states, which are all states not involved in intervehicle observations, can then be eliminated from each subgraph and the joint probability of the start and end external states can then be shared between vehicles and combined with other subgraphs and further eliminated to yield the solution to the entire graph when intervehicle observations are included. Once a subgraph is transmitted

the linearisation cannot be changed, however a local optimiser can relinearise and the local linearisation point can inform selection of linearisation points for subsequent graphs.

The second contribution is exploitation of variable reordering within these packets to enable simple handling of delayed observations that target an existing historical state such as may occur with visual loop closures. We identify the calculations required to give the conditional probability of the delayed historical state and the existing external states chronologically before and after it. This can then be combined with previously transmitted probability information. This reduces the recalculation to updating the factorisation of a single subgraph and is independent of the time since the observation was made.

The third key contribution is a method and conditions for insertion of states into existing packets that does not require relinearisation and re-elimination nor invalidation of previously transmitted information. We derive the conditions that enable this method and our fourth contribution is the presentation of two motion models that describe real world systems and conform to the derived conditions. Together this permits handling of the general out of sequence case where remote platforms may be accumulating new states at relatively low frequencies and be unaware that they were observed. Like the second contribution it limits recalculation to a single subgraph. It does additionally require linearisation of the motion model factors either side of the newly inserted state along with updating the factorisation of the subgraph.

These methods are demonstrated in simulation to explore issues with a partially fixed linearisation that we use and on captured data from a small fleet of ground vehicles to demonstrate performance on real systems.

This thesis emphasises exact solutions after linearisation instead of approximations. The absence of approximation methods such as sparsification reduces concern surrounding conservativity or inconsistency of localisation estimates and simplifies general graph handling. It also emphasises iterative methods over batch methods for communication, in turn requiring less bandwidth to generate a complete state history that is updated at each interaction. This is important for cooperative localisation in contrast to cooperative mapping where odometry drift over time can be reduced by frequent interaction. This work results in predictable bandwidth usage, based on state space and intervehicle sensing rate and enables more precise estimates of bandwidth required for the localisation task. This then permits system design and offline planning for bandwidth allocation to localisation as well as planning or control tasks.

Acknowledgements

It took a long time to get here and it would not have been possible without the support of a large number of people – both directly involved in the work presented here, indirectly through general robotics discussions and those who just kept me going even if talk of optimisation, bandwidth and robots was meaningless.

Specifically thank you to my supervisors Stefan Williams and Oscar Pizarro who have made numerous suggestions about research direction and priorities and pushing for greater clarity in my explanations. Also thanks to the rest of the postgrads from the Marine Robotics Group and rest of the ACFR over the years for company on field trips, lunch trips and general fun around the lab - Don, Dan B, Ari, Ash, Dan S, Lash, Michael, Dush, Troy, Bryan and Peter as well as our techs Christian and Andy.

Also thanks to my parents Michael and Lynda for supporting me since birth and enabling and encouraging me to push myself.

And finally to the woman who was at the start of candidature my girlfriend, then halfway through fiancée and now wife – for your patience whilst I completed this, planning a wedding I said I'd be done in time to help out with and making life that bit more fun.

Cat, I thank you most of all.

Contents

List of Figures

List of Tables

List of Algorithms

List of Symbols

$\oplus$	Group operator for variables.
exp	Exponential map to calculate Lie Group from Lie Algebra or manifold from tangent coordinates.
F_i	Set of frontal variables in a conditional probability distribution being conditioned by S_i that is $p(F_i \mid S_k)$.
$f_i(\Theta_i)$	Vector valued error metric.
$g_i(\Theta_i)$	Probability density function for metric f_i such that $g_i(\Theta_i) \propto \exp -\frac{1}{2} f_i^T(\Theta_i)\, \mathbf{\Lambda}_i^{-1} f_i(\Theta_i)$.
$\mathbf{J}$	Jacobian matrix.
$\mathbf{J}_{ij}$	Element of Jacobian matrix representing $\frac{\partial f_i}{\partial \theta_j}$.
$\mathbf{\Lambda}_i$	Covariance matrix for error metric f_i.
log	Logarithm map to calculate Lie Algebra from Lie Group or tangent coordinates from manifold coordinates.
$p_j'^i$	Compressed packet j of sensor data on vehicle i.
p_j^i	Packet j of sensor data on vehicle i.

$\mathbb{R}$	the set of real numbers.
S_i	Set of variables connected to factors connected to θ_i or set of frontal variables F_i used in conditional distributions.
θ_i	Variable i in group/manifold form.
$\delta\theta_i$	Update for variable i in $\mathbb{R}^n$.
θ_i^0	Linearisation point for Variable i in group/manifold form.
Θ_i	Set of variables used in metric f_i in group/manifold form.
x_j^i	State of vehicle i at time j.
z_j^i	Combined measurements on vehicle i at time j.

Acronyms

ACFR	Australian Centre for Field Robotics.
AUV	Autonomous Underwater Vehicle.
BA	Bundle Adjustment.
CL	Cooperative Localisation.
OWTT	One Way Travel Time.
SAM	Smoothing and Mapping.
SfM	Structure from Motion.
SLAM	Simultaneous Localisation And Mapping.
VO	Visual Odometry.

Chapter 1

Introduction

The aim of this thesis is to develop methods that permit asynchronous state updates between agents and handles low bandwidth and unreliable communication channels.

1.1 Background and Motivation

Interest in multiple robot deployments stretch back further than 20 years with early research in system architectures, task allocation and communication protocol [5, 26] with a description of some of this early work given by Cao et al. [11]. Modern probabilistic localisation and mapping originated at a similar time for single robot systems [19, 66, 67] based on Kalman Filtering [40]. In this work the challenge was to estimate position relative to a starting point without global information such as GPS or a compass may provide. Observation of landmarks then permitted generation of a map which was used to constrain growth in uncertainty of robot poses, and of the landmarks themselves with much interest in the consistency and long term convergence properties [7, 17, 28, 35]. This problem, and the solutions to it are known as Simultaneous Localisation and Mapping (SLAM). Later work relaxed assumptions about the lack of global information (such as GPS and compass) and their incorporation in the state estimates is now commonplace for many applications with interest in large scale or cheap high noise sensors [3, 33, 65, 69].

Application and extension of these probabilistic single robot methods to the multiple robot case did not occur immediately although the concept of stationary landmarks was used

by Kurazume et al. [43] and Rekleitis et al. [61]. In these works the total group was divided into multiple teams with one team held in location as landmarks whilst the other robots moved with these roles alternating as the entire group moved around the region. An approach with all vehicles moving simultaneously based on Kalman filtering was described by Sanderson [64] although some noise parameters were ignored. Modern probabilistic localisation for multiple robot systems arrived soon after with a decentralised information filtering approach [55] and a Kalman filtering approach [63]. It was realised that tracking of information fusion was important to prevent double counting where information from a single observation is fused multiple times. Usage of a channel filter [55] was proposed to counter this within decentralised information/data filtering approaches. In these and many following works the landmarks and maps of SLAM were ignored in favour of direct observations of other robots giving rise to Cooperative Localisation as a distinct field, although there is significant overlap with developments in the area of single and multiple robot SLAM techniques. Subsequent work incorporated the mapping portion of SLAM [78] without direct observations. Alternate Monte Carlo (or Particle filtering) approaches were also developed [68] to better handle multi-modal and non-Gaussian uncertainty that arises from nonlinear models.

The desire to incorporate constraints to historic states in single vehicles through laser scan matching [49] or stereo image correspondence [21] led to the retention of historic states and investigation of the Full SLAM problem where a sparse information matrix representing the complete trajectory of the vehicle is used [22, 50, 75]. Identification of this with nonlinear least squares and graphical models such as discussion in Frese and Hirzinger [25] led to methods such as $\sqrt{SAM}$ [15] and subsequent adaptation to online usage [31, 37, 39, 42].

Extension of these methods to multi-robot deployments has gained attention, although present issues with bandwidth, reliability of communication channels, avoiding dependence on individual robots and computation tractability. Increasing the number of robots in the system tends to increase the concern with all these issues. Delays in communication or observation of remote systems have led to delayed state methods [8, 46] that bridge the gap between filtering and smoothing approaches where a small history is retained. DDF-SAM [13, 14] uses landmarks to create a map on each platform that is then shared with landmarks being associated on each platform, although it cannot handle direct inter-robot observations. Handling direct observations has been shown with group of methods based on

One-Way-Travel-Time (OWTT) [73, 76] that exploit both the information imbalance with surface ships and submarines such that information is needed primarily in one direction only and the dual nature of their acoustic modems as a communication and (one-way) range device when coupled with synchronised low drift clocks. Methods developed here cover delayed state and full graph approaches. The use of an 'origin' state that is used to base packets off resulted in a tolerance to packet loss although the creation method would result in double counting of information if used in a bidirectional network. More recent work based on this sensing methodology [74] yielded a solution based on sparsification or approximate marginalisation that uses odometry composition and decomposition operations relative to local coordinate frames along with approximate priors. In this method any packet can be lost without the system failing and a similar graph structure for solutions is used as developed here. Their solution however is restricted to minimal (x, y) states.

Alternative methods that do not rely as heavily on a known 'origin' state exist [59]. Here states between two consecutive inter-robot measurements are marginalised and a sparse approximation of the dense information matrix over the two states and landmarks is generated and transmitted.

The structure of the solutions to multi-robot deployment localisation (and mapping) is generally related to the specific constraints the problem faces. Where cases like underwater robotics have very limited bandwidth over acoustic modems other scenarios on the ground or in the air have much higher bandwidth. Again underwater, indoor or extra-terrestrial problems lack the use of a Global Navigation System that they can independently position against (such as GPS, GLONASS and Gallileo) and must rely on landmarks or relayed information from robots with this globally visible information.

The centralised techniques that many of these are based off provide a target standard for decentralised methods. These solutions utilise all available information with no approximations beyond linearisation which is repeated to reduce the ill-effects of poor initial linearisation point selection. Recent prominent techniques such as iSAM2 or $\sqrt{SAM}$ give examples of the reference standards for online and offline centralised techniques.

1.2 Problem Statement

The difficult problems in cooperative localisation concern double counting of sensor information, bandwidth limits, unreliable communication, delayed communication, tractability of computation and handling asynchronous and hereogeneous observations.

We aim to enable solutions nearly equivalent to centralised solutions for multiple robots in an online decentralised manner. We are not seeking to handle swarms of hundreds of robots but a local team in the order of five to ten. The solution should, on a given robot, contain all local variables and have nearly identical marginal and conditional probabilities. The method should avoid replication of calculation by solving locally before sharing as much information as possible whilst still permitting fusion with solutions from other vehicles. The method should be generic and permit handling of all Gaussian noise models for intra and inter-robot observations. The method should minimise the bandwidth required and handle arbitrary delays for limited range or high latency communications. Resilience against dropped packets is also desired but not required.

1.3 Thesis Contributions

This thesis makes four novel contributions to the state of the art in multi-vehicle decentralised localisation under bandwidth limited communication constraints. These contributions include:

- The development of a method for information fusion that allows iterative in-sequence construction and transmission of a summary graph containing only the states involved in intervehicle measurements for each vehicle.

  This method eliminates the risk of double counting of measurements and permits arbitrary communication patterns. While existing methods either create chains of linked packets using marginal and conditional probabilities on each vehicle which require careful manipulation [8] or restrict vehicles from retransmitting [73], the approach proposed here allows packets to be transmitted between vehicles without introducing inconsistencies in the final estimate. The resulting estimates are exact up to the local linearisation that takes place on each vehicle.

- The extension of this method to facilitate the inclusion of out of sequence intervehicle measurements without retransmission through promoting an existing state that is sufficiently close to the desired time. Existing methods require a combination of retransmission of existing data, increasing complexity of data tracking, or recalculation across the interval between the original measurement time and the current time using a limited delay horizon.
- A further extension to facilitate inserting a new state where an existing state does not exist and the vehicle model obeys certain conditions. Existing methods require recalculation of all data since the new observation or information cancellation and re-elimination since the observation time of the out of sequence measurement.
- The analysis of vehicle models that conform to the insertion requirements for out of sequence measurements.

These contributions are validated using a combination of simulation experiments and data collected by teams of ground vehicles.

1.4 Outline

The remainder of this thesis is organised as follows.

Chapter 2 presents background information related to the problem. It provides an overview of the state of the art in areas of state estimation including applications to single vehicle localisation, SLAM and cooperative localisation. Additionally it includes background on some of the less known mathematical methods that are used for state representation and metric calculation.

Chapter 3 describes the mathematical foundation for the iterative in-sequence construction and transmission of a summary graph. It validates these developments using both simulated, multi-vehicle scenarios as well as using real data collected during the deployment of a team of robots.

Chapter 4 extends this formulation to allow for the inclusion of out-of-sequence intervehicle measurements as well as intravehicle measurements resulting from loop closure events. This is again validated in both simulated scenarios and real data collected from a team of robots.

Chapter 5 summarises the main contributions of the thesis and provides an outline of directions for future work.

Chapter 2

Background and Related Work

This chapter will cover background knowledge required for understanding the details of later chapters as well as a survey of related literature. The first section will cover factor graphs and nonlinear estimation including a section on state representation. The discussion of related literature in the second section, covering alternate localisation methods and applications of these methods, will be framed by the terminology and methods described in the first section.

2.1 Factor Graphs and Nonlinear Optimisation

The basis of many localisation methods for single and multiple robots in recent literature has utilised nonlinear optimisation techniques and weighted nonlinear least squares in particular. Nonlinear least squares seeks to find the state/value that minimises the scalar error calculated from the error function describing the problem. We begin with a relatively informal description of describing localisation problems and the scalar error function before getting more rigorous and developing more completely the mathematics and models we use to describe it.

The scalar error function is the product of smaller parts of the problem that describe sensor observations and vehicle motion. Both types of these functions are derived from conditional probability distributions and are similar in form. Sensor observations, such as vehicle orientation from a compass, coarse precision from GPS, speed from wheel encoders

or inter-vehicle observations such as range from laser, RADAR or stereo cameras can be treated as the conditional probability of an observation given the true state. As is typical in localisation a Gaussian distribution is used to represent this probability distribution as in (2.1) where h is a function to predict the sensor observation from the state. A similar conditional probability distribution is used to represent the vehicle model in (2.2) where the probability of the next state is given conditioned on the previous state. This function represents the difference between the estimated next state x_j^i and the predicted next state $h(x_{j-1}^i)$.

$$p\left(z_j^i \mid x_j^i\right) = \det 2\pi\mathbf{\Lambda}_i{}^{-\frac{1}{2}} \exp -\frac{1}{2}\left(z_j^i \oplus h(x_j^i)^{-1}\right)^T \mathbf{\Lambda}_i^{-1}\left(z_j^i \oplus h(x_j^i)^{-1}\right) \tag{2.1}$$

$$p\left(x_j^i \mid x_{j-1}^i\right) = \det 2\pi\mathbf{\Lambda}_i{}^{-\frac{1}{2}} \exp -\frac{1}{2}\left(x_j^i \oplus h(x_{j-1}^i)^{-1}\right)^T \mathbf{\Lambda}_i^{-1}\left(x_j^i \oplus h(x_{j-1}^i)^{-1}\right) \tag{2.2}$$

$$\tag{2.3}$$

These probability distribution functions can be combined to give a total probability for all states and sensor observations through multiplication of these functions. The expected value of the vehicle states over time is the mean of the combined distribution which, for Gaussian distributed probability density functions is the argument that gives the maximal value. It is this expected value that we seek and is the aim of solving in nonlinear least squares based methods.

Typically the probability distribution function is modified to produce an easier to compute and solve equation by discarding normalising terms to get a likelihood function and then taking the logarithm and further discarding constant terms which do not change where the minimal (after taking the logarithm) value occurs in the state space. At this point the more familiar least squares format appears.

Observations of the structure of the probability distribution function and derived least squares error function led to formation of a graph structure known as a factor graph where these functions, or factors in the graphical model terminology, are connected via edges to variables if they depend on the value of that variable. The connectivity of this graph corresponds to the sparsity pattern of a Jacobian matrix that is computed whilst solving the nonlinear least squares problem.

Methods for localisation developed recently have utilised graphical models to describe the formulation of and the solution to these problems and enable insight into the structure of

the sparse matrices that are the mathematical representation of the problems. In particular the factor graph has been used for the formulation of localisation and mapping problems and construction of the Jacobian and Information matrices [15] and a relatively recently developed model termed the Bayes Tree [38] for representing the upper (or lower) triangular square root inverse matrix that is used for back substitution in the solution to these problems after variable elimination. The underlying methods are applicable to Simultaneous Localisation And Mapping (SLAM), Structure from Motion (SfM), Visual Odometry (VO) and numerous other optimisation problems however we will focus on applications relating to localisation (with mapping to a lesser extent).

Factor graphs are bipartite graphs with variable nodes θ_i that represent state estimates and function or factor nodes $g_j(\Theta_j)$ that are likelihood functions that depend upon a subset of the state estimates. The total factor graph g is the product of all these factors $\prod_j g_j(\Theta_j)$. Probability distributions which are normalised likelihood functions are not required as the focus is to find the maximum likelihood state estimate which is independent of normalisation.

We use a factor graph for the cooperative localisation problem similar to the one depicted in Figure 2.2. Here the variables θ are robot states x_j^i which denote the state of robot i at time step j on that robot. We use different hues to designate states from different robots and additionally vary the brightness to denote with faded colour 'internal' states that are not connected through an observation to another vehicle and brighter colour 'external' states that are connected via a factor to another vehicle. We additionally include landmark states l_x where multiple digits indicate multiple landmarks were observed simultaneously and have been bundled for convenience. These states are also variables and are used in map building and landmark free loop closures (visual loop closures or similar where the landmark is eliminated before the factor is included in the global graph). The factors, depicted as small black circles, are probability distributions or likelihood functions that denote the likelihood of an observation z_j given connected variables Θ_j. These distributions arise from sensor models for local state sensors and relative state sensors between vehicles or vehicles and landmarks. Examples of local state sensors include position through global navigation systems, orientation from compasses or velocity through wheel encoders or Doppler measurements. Relative state sensors include visual, laser, RADAR or acoustic ranging and bearing observations of remote targets or pair matching for visual loop closures and laser range scan matching.

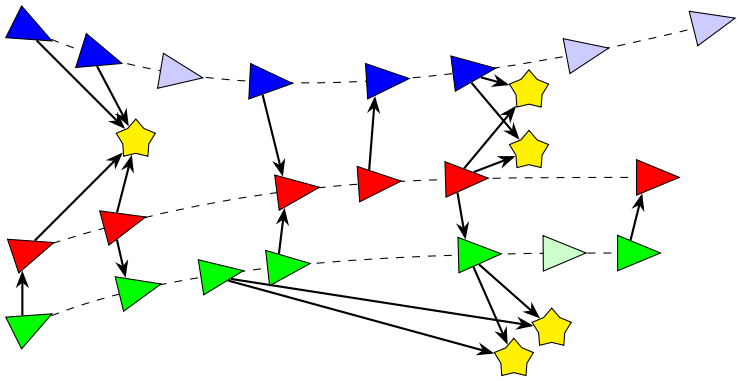


Figure 2.1 – A generic three vehicle system over time. Vehicle states are coloured by vehicle and are red, blue or green with landmarks in yellow. Odometry links connect vehicle poses and intervehicle observations are indicated by arrows denoting the target of the observation.

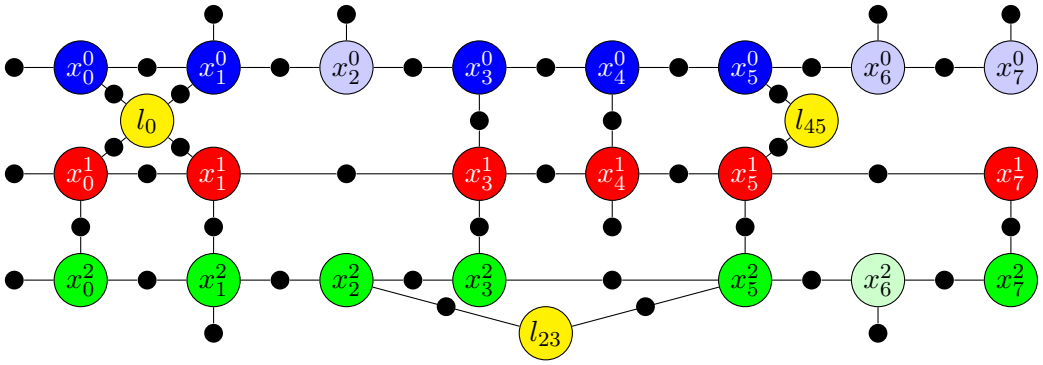


Figure 2.2 – A generic three vehicle factor graph. State variables are coloured by vehicle and either red, blue or green for each vehicle with landmarks in yellow. Factors, which represent observations relating to vehicle and landmark states are the black circles.

Finally vehicle models may also be used to create a link between states over time using knowledge of system dynamics.

More rigorously a factor graph is a graph $FG = (\mathcal{G}, \Theta, \mathcal{E})$ with two node types and edges that connect only between nodes of different types. Factor nodes $g_j \in \mathcal{G}$ represent the likelihood functions and variable nodes $\theta_i \in \Theta$ represent the states. Edges $e_{ij} \in \mathcal{E}$ represent dependence of a factor function g_j on a variable θ_i. Θ_j is the set of variables a factor g_j depends upon. The graph is the product of all factors and the value represents the likelihood of a set of variable values. The aim is to find the maximum likelihood value of all variables, which is the same as finding the global maximum of $g(\Theta)$. η is a normalisation constant used to scale the graph to a probability distribution. This problem can be easily reposed into a more familiar (nonlinear) least squares form as in (2.8) when assuming that

likelihood functions are Gaussian distributions containing a vector error function $f_j(\Theta_j)$ and covariance matrix Λ_j.

$$p(\Theta) = \eta g(\Theta) \tag{2.4}$$

$$p(\Theta) = \eta \prod_j g_j(\Theta) \tag{2.5}$$

$$\Theta^* = \arg\max_{\Theta} \prod_j g_j(\Theta_j) \tag{2.6}$$

$$= \arg\max_{\Theta} \prod_j \exp\left(-f_j(\Theta_j)^T \Lambda_j^{-1} f_j(\Theta_j)\right) \tag{2.7}$$

$$= \arg\min_{\Theta} \sum_j \left(f_j(\Theta_j)^T \Lambda_j^{-1} f_j(\Theta_j)\right) \tag{2.8}$$

The number of rows m_i in the factor g_j is the dimension of the observation, and corresponds to the number of rows and columns in the square covariance matrix Λ_j representing the uncertainty in the factor.

Finding the minimum of the nonlinear least squares problem is done through iteration over linearised versions of the problem with each iteration updating linearisation points and ideally approaching the global minimum. Algorithms like Gauss-Newton, Levenberg-Marquardt and conjugate gradient can be used to calculate the update each iteration with different advantages to each [16, 30]. We will use Gauss-Newton here for demonstration purposes.

2.1.1 Linearisation

Linearisation of these functions is performed using first order Taylor expansion of the error metrics f_j around linearisation points Θ^0. Expanding the error term in (2.8) we find

$$\Theta^* = \arg\min_{\Theta} \sum_j \left(\left(f_j(\Theta_j^0) + \sum_i \frac{\partial f_j(\Theta_j^0)}{\partial \theta_i}\left(\theta_i - \theta_i^0\right)\right)^T \Lambda_j^{-1} \left(f_j(\Theta_j^0) + \sum_i \frac{\partial f_j(\Theta_j^0)}{\partial \theta_i}\left(\theta_i - \theta_i^0\right)\right)\right) \tag{2.9}$$

We can then rearrange to optimise over the change in state $\Delta\Theta$ relative to the linearisation point where $\Delta\Theta = \left(\Theta - \Theta^0\right)$

$$\Delta\Theta^* = \arg\min_{\Delta\Theta} \sum_j \left(\left(f_j(\Theta_j^0) + \sum_i \frac{\partial f_j(\Theta_j^0)}{\partial \theta_i} \delta\theta_i \right)^T \Lambda_j^{-1} \left(f_j(\Theta_j^0) + \sum_i \frac{\partial f_j(\Theta_j^0)}{\partial \theta_i} \delta\theta_i \right) \right) \tag{2.10}$$

This can be further simplified to collate the summations to a single Jacobian matrix, covariance matrix and error metric vector for the entire graph

$$\Delta\Theta^* = \arg\min_{\Delta\Theta} \left(f(\Theta^0) + J\Delta\Theta \right)^T \Lambda^{-1} \left(f(\Theta^0) + J\Delta\Theta \right) \tag{2.11}$$

where f is the combined error metric vector, J the global Jacobian matrix whose rows correspond to the metric functions and the columns to variables they depend upon, Λ is the global block diagonal covariance matrix and $\Delta\Theta$ combines the individual $\delta\theta_i$ into a single vector. Given the structure of many problems, the Jacobian J is sparse as each error metric depends only on a small number of variables.

2.1.2 State Representation

Whilst methods derived from the Kalman Filter, which will be covered briefly in the next section, including the EKF, EIF, SEIF and DDF - historically solved for the state directly this required careful handling throughout the estimator code for cases like rotations where the state values are non-Euclidean and Euclidean distance metrics do not match the true distances in these spaces. The simplest example of this is calculating the robot's orientation over time. A local sensor such as a compass will provide absolute estimates in the $\pm\pi$ range (or equivalently $[0, 2\pi)$). Accumulation through odometry of circular motion will result in the orientation estimate exceeding $\pm\pi$ resulting in incorrectly large innovations when fusing new compass measurements. This can be compensated through methods including adding $\pm 2k\pi$ to the compass observation or other workarounds that result in a normalised difference in the $\pm\pi$ range. These can involve careful manipulation of the values and updating the offset as estimates of values change. The choice of solving for the new estimate directly includes the linearisation points in the error metric (through the $\frac{\partial f_j}{\partial \theta_i}\theta_i^0$ terms). The

alternative of solving for $\delta\theta_i$ has only the error metric and so depends only upon the error value not the underlying values as well.

Treatment of the states as a manifold, or more precisely an n-dimension differentiable manifold, enables calculation of derivatives and further creation of a tangent space at each point on the manifold. The tangent space is important as the space is n-dimensional Euclidean, matching the results of optimisation from least squares. Furthermore vector values in the tangent space can be mapped back to the manifold, relative to the linearisation point, enabling composition with the updates in a consistent manner. The mapping from tangent space to the manifold is known as the exponential map exp and the reverse mapping is the logarithm map log. In this way the update equation to add the delta back to the linearisation point is adjusted. What is $\theta_i = \theta_i + \delta\theta_i$ for Euclidean states becomes $\theta_i = \exp_{\theta_i^0}(\delta\theta_i) = \theta_i^0 \oplus \exp_0(\delta\theta_i)$ representing movement by $\delta\theta_i$ in the tangent space of the manifold at θ_i^0 and alternatively as the composition of the linearisation point θ_i^0 with the movement in the tangent space around the group identity $\exp_0(\delta\theta_i)$.

Another set of mathematical structures that enable simpler and more accurate handling of non-Euclidean variables are Lie Groups and their related Lie Algebra [12]. Introductions and examples of applications to robot dynamics [57] and later robot localisation [2, 31, 39, 48] can be readily found. Lie groups are continuous transformation groups that are also n-dimensional differentiable manifolds. Groups require a binary group operation, a null value or identity, an inverse for each operation, associativity for operations and closure in that the result of the group operation on group members must also be a group member. Lie groups additionally assert that the group operation and inversion are differentiable.

Of particular interest to many robotics applications are the matrix Lie Groups $SO(2)$, $SO(3)$, $SE(2)$ and $SE(3)$ the former pair of which handle rotations and the latter pair rotations and translations in 2 and 3 dimensional Euclidean space. The numerical representation of these are the standard square rigid transformation matrices. Whilst these matrices can be useful for calculating transformations they do not immediately provide n dimensional Euclidean values that are desired for the solving process. The tangent space to a Lie Group at the identity is known as the Lie Algebra and for the matrix Lie Groups is a square matrix of the same size with n degrees of freedom and each dimension is in $\mathbb{R}$. The exponential and logarithm maps are the matrix exponential and logarithm maps. Whilst these are impractical to calculate in the general case for each of these matrix Lie

Groups there are closed form equations that avoid calculating the full matrix exponential and logarithms. When using Lie Groups and Algebra the update equation can be rewritten as $\theta_i = \theta_i \oplus \exp_0(\delta\theta_i)$ where $\oplus$ is the group operation.

A simple example of a matrix Lie Group and its Lie Algebra is the group of rotations in the plane $SO(2)$ shown in (2.12) and translations and rotations in the plane $SE(2)$ shown in (2.13) which is slightly more complex but shows the dependence of translation on orientation.

$$\begin{bmatrix} \cos\omega & -\sin\omega \\ \sin\omega & \cos\omega \end{bmatrix} = \exp \begin{bmatrix} 0 & -\omega \\ \omega & 0 \end{bmatrix} \tag{2.12}$$

$$\begin{bmatrix} \cos\omega & -\sin\omega & \frac{u\sin\omega - v(1-\cos\omega)}{\omega} \\ \sin\omega & \cos\omega & \frac{v\sin\omega + u(1-\cos\omega)}{\omega} \\ 0 & 0 & 1 \end{bmatrix} = \exp \begin{bmatrix} 0 & -\omega & u \\ \omega & 0 & v \\ 0 & 0 & 0 \end{bmatrix} \tag{2.13}$$

2.1.3 Solving

Given the weighted linear least squares problem in (2.11) we can further simplify by incorporating the covariance matrix information into the Jacobian matrix and combined error metric to create an unweighted linear least squares problem as in (2.15). If we use the Cholesky decomposition of the inverse covariance matrix we get $\left(\Lambda^{-1}\right)^{T/2}\left(\Lambda^{-1}\right)^{1/2} = \Lambda^{-1}$. We can then split the components and pre-multiply the Jacobian matrix and error vector. This permits us, without loss of generality to incorporate the covariance term into the Jacobian matrix and error vector and drop it from the terms. Here we will denote the change forms with '$'$' but drop it again following for simplification of notation.

$$\Delta\Theta^* = \arg\min_{\Delta\Theta} \left(\Lambda^{-\frac{1}{2}}\left(f(\Theta^0) + J\Delta\Theta\right)\right)^T \Lambda^{-\frac{1}{2}}\left(f(\Theta^0) + J\Delta\Theta\right) \tag{2.14}$$

$$= \arg\min_{\Delta\Theta} \left(f'(\Theta^0) + J'\Delta\Theta\right)^T \left(f'(\Theta^0) + J'\Delta\Theta\right) \tag{2.15}$$

At this point application of Gauss-Newton, Levenberg-Marquardt or other linear solvers is possible to obtain a final solution for $\Delta\Theta$. We temporarily denote the linearised function of $\Delta\Theta$ we are minimising as $F(\Delta\Theta)$. In the case of Gauss-Newton the derivative of F with

respect to $\Delta\Theta$ is calculated and set to equal 0.

$$F(\Delta\Theta) = f(\Theta^0)^T f(\Theta^0) + f(\Theta^0)^T J\Delta\Theta + \Delta\Theta^T J^T f(\Theta^0) + \Delta\Theta^T J^T J\Delta\Theta \tag{2.16}$$

$$\frac{\partial F}{\partial \Delta\Theta} = 2f(\Theta^0)^T J + 2\Delta\Theta^T J^T J = 0 \tag{2.17}$$

If we transpose and rearrange the equation we arrive at the normal linear least squares problem.

$$J^T J\Delta\Theta = -J^T f\left(\Theta^0\right) \tag{2.18}$$

Solving (2.18) involves inverting $J^T J$, which is also known as the Information matrix $\mathcal{I}$, and multiplying both sides on the left by this inverse to obtain the solution for $\Delta\Theta$. Direct instantiation of the dense inverse of $\mathcal{I}$ can be avoided using the Cholesky decomposition of $\mathcal{I}$. The Cholesky decomposition calculates the square root upper triangular matrix R and its transpose such that $R^T R = \mathcal{I} = J^T J$. This then permits a two step process with forward and backward substitution and avoids direct inversion of $\mathcal{I}$.

$$J^T J\Delta\Theta = -J^T f(\Theta^0) \tag{2.19}$$

$$R^T y = -J^T f(\Theta^0) \tag{2.20}$$

$$R\Delta\Theta = y \tag{2.21}$$

An alternative approach is to avoid construction of $\mathcal{I}$ altogether using the QR decomposition that calculates the same R where $J = QR$. Here Q is a unit determinant orthogonal matrix such that $QQ^T = I$. QR decomposition can take place through a number of methods such as Givens rotations and the Gram-Schmidt algorithm. These algorithms construct Q as the product of many smaller unit determinant orthogonal matrices Q_i. Construction of Q itself can be avoided if each transformation is simultaneously applied to both sides of (2.18). Finally back substitution yields the solution in (2.23).

$$Q^T J\Delta\Theta = -Q^T f(\Theta^0) \tag{2.22}$$

$$R\Delta\Theta = -Q^T f(\Theta^0) \tag{2.23}$$

The delta update $\Delta\Theta$ is then added to the linearisation point Θ^0 and the steps from linearisation through solving are repeated until convergence where the update $\Delta\Theta \approx \mathbf{0}$.

As an aside column ordering chosen for construction of J defines elimination ordering and influences computational cost by controlling sparsity of $\mathcal{I}$ and R. Previous work in localisation and mapping has used a minimal computation ordering heuristic such as COLAMD or METIS [1]. We will cover elimination ordering further in the next section on Graphical Approaches.

2.1.4 Graphical Approaches

Whilst a factor graph represents the initial problem it does not assist in representation of the final solution. The process of Variable Elimination targets a single variable at a time in an ordering known as the elimination order and converts the factor graph to another graphical model known as a Bayesian Network [60] through inducing directional dependence in the edges based on order of elimination. Different orderings will generally result in different Bayesian Networks. This network encodes the conditional probabilities of a single variable upon a number of other variables in a directed acyclic graph. Whilst acyclic there may still be multiple paths from one node to another complicating inference.

In the case of Bayesian Networks resulting from variable elimination the network is chordal which permits conversion to a tree structure, merging the multiple paths that can exist between nodes in the Bayesian Network. The tree structure we convert to here is the Bayes Tree [38]. This structure retains the directed nature of the Bayesian Network but instead of encoding conditional probabilities of single variables relative to others it encodes the conditional probability of a set of 'frontal' variables upon the 'separator' variables. Separator variables also appear in the node's parent as either frontal or separator variables and all variables appear exactly once as a frontal variable in a complete tree. The directed edges follow the flow of information in the solution from the root to the leaves mapping the back substitution process. An example of a Bayes Tree is shown in Figure 2.3 and was created from the factor graph in Figure 2.2 with a randomly chosen ordering. Mathematically the Bayes Tree is represented as the product of conditional probabilities and can be directly

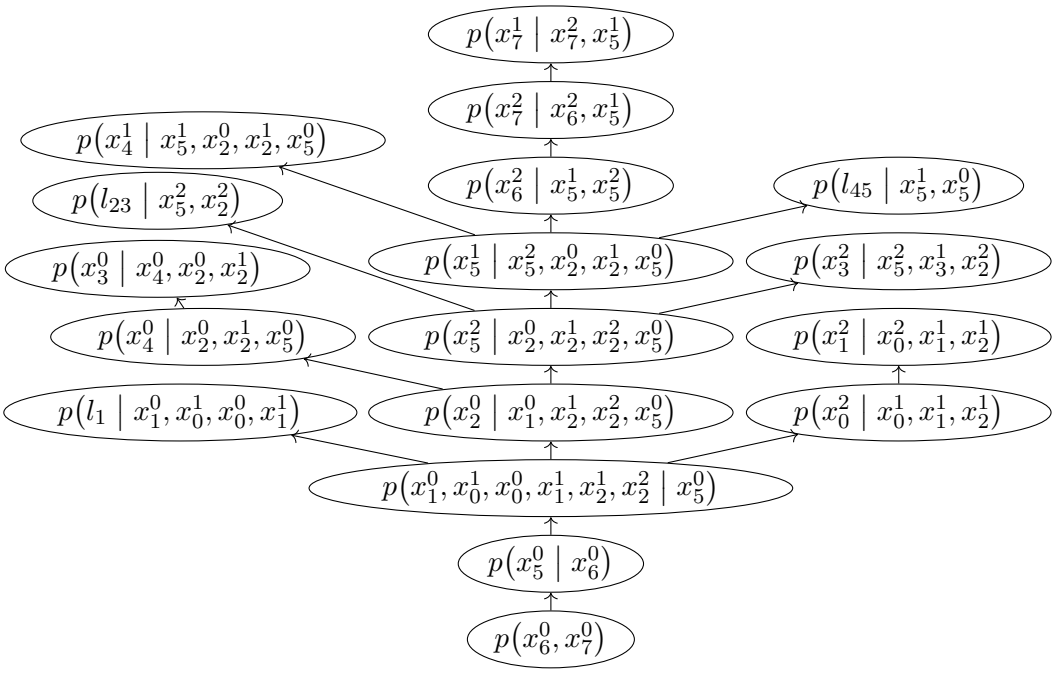


Figure 2.3 – A Bayes tree generated from a random elimination ordering for the factor graph in Figure 2.2. Larger cliques involve greater computation as each is a dense matrix and vector that requires inversion. Sparsity is greater when there are less variables in each clique. Computationally optimal orderings will aim to minimise clique size.

mapped to rows of the upper triangular matrix R that is calculated from QR or Cholesky decomposition methods given in the previous section.

$$p(\Theta) = \prod_k p(F_k \mid S_k) \tag{2.24}$$

Variable elimination can be understood within the graphical models as an iterative process that converts factor graphs to Bayesian Networks and then Bayes Trees [38]. The process takes the next variable in the elimination order and removes it along with all factors adjacent to it from the factor graph. The factors are combined into a single element and the conditional probability of the target variable dependent upon the variables adjacent to the removed factors is calculated. The conditional probability is added to the Bayes Network and the remaining components of the factor returned to the factor graph. The new factor is connected to all variables adjacent to all factors removed and results in 'filling-in'. Once the process is complete the Bayesian Network can be converted to a Bayes Tree by building in

reverse elimination order. Starting from the node representing the last eliminated variable, nodes are taken from the Bayesian Network and added to the tree. Thus the first eliminated variables make the leaves and the last eliminated variable is part of the root node.

Using this graphical model approach to variable elimination we can return to the linear algebra approach and replicate it. The process of removing the factors surrounding the target variable of dimension n from the factor graph is equivalent to isolating the rows of the Jacobian matrix that contain non-zeros in the target n columns. Extraction of the conditional then involves performing QR decomposition upon this isolated section, clearing under the diagonal for n columns. The top n rows are then the conditional probability and can be separated out, whilst the remaining rows return to the Jacobian. Givens Rotations and Householder Reflections are relatively stable methods for calculating the QR decomposition that also permit this partial process. Recovery of the delta can take place through backsubstition using the solution matrix.

2.2 Related Literature

Accurate estimation of state is an important problem for robotic and human controlled systems. Methods for state estimation have varied in robustness and requisite sensors over time although now are typically based on statistical methods that incorporate sensor uncertainty.

2.2.1 Filters

The origin for many of the methods now used in robotics is the Kalman Filter [40]. Seminal work by Durrant-Whyte [19], Smith et al. [66], Smith and Cheeseman [67] introduced the Kalman Filter to robotics as a solution to the SLAM problem, replacing methods that overestimated uncertainty and lacked statistical bounds, instead using absolute min/max bounds [10].

The Kalman Filter was designed as a method to handle weighted incorporation of sensor data from a multitude of independent observations over time. The filter was shown to be optimal for linear systems with Gaussian noise models, neither over or under estimating uncertainty. Output of the filter consists of a mean state estimate vector and a covariance

matrix representing the multi-variate Gaussian uncertainty around the mean state vector. Joint covariance with previous state estimates is not calculated with the focus on the latest state and the nature of the Kalman Filter as a recursive filter. The filter itself consists of two sets of calculations. The first set is state prediction using a dynamic model to project the state forward in time. The second set updates the state estimate through incorporation of sensor data obtained at that time. Fusion in this manner permits the uncertainty to be less than that of a single observation, especially if the dynamic motion model does not introduce significant noise.

Whilst the prediction stage is relatively fast to compute, the inclusion of sensor information involves numerous matrix inversions and an increased count of matrix multiplications. To this end an alternative representation of the filter was proposed for usage in slowly changing systems, or where a significant amount of information was to be fused at any single time. This representation, known as the Information Filter, uses the inverse covariance matrix or information matrix and the information vector, which is equal to the information matrix multiplied by the mean state estimate vector from the covariance form. In information form the state update from new sensor measurements includes a single inversion of the sensor covariance matrix which is smaller than the covariance matrix and often constant for a given sensor. State prediction is however more expensive, involving matrix inversions that are reliant on the previous information matrix.

Further alternate formulations using factored versions of the information matrix or covariance matrix have been proposed such as the Square Root Information Filter [51]. This results in gains in numerical accuracy or computation speed, although literature on usage of these is relatively sparse.

The methods shown so far are for linear systems and models which do not properly represent many systems of interest, such as vehicles moving in a plane or volume or nonlinear observation models such as range and bearing. A relatively straightforward extension of the Kalman Filter to handle this is the Extended Kalman Filter which uses the first order Taylor expansion of state prediction and sensor observation functions to provide the process and sensor model matrices from their Jacobians [27]. This is readily translated to the Information form of the filter, with the caveat that the mean state vector must still be used for the state prediction and sensor observation functions. This requires inversion of the Information matrix to recover the state.

All these variations are based on recursive filters which use different operations for state predictions and odometry and for fusion of sensor information about the current state. Whilst this enabled rapid calculation of the current state and uncertainty in the mean it limited the ability to incorporate loop closures or links to other states and results in growth in uncertainty over time unless some global source of absolute position information is present. Many of the early robotics applications assumed no global frame existed or was available and developed methods with augmented stationary states - or landmarks, that were features that the robot could identify in the environment and navigate relative to. This did come with the limitation that the landmarks needed to be readily identifiable and repeatedly observed - ideally from a large range of locations in the environment.

2.2.2 Smoothers

Seminal work of Lu and Milios [49] opted to retain historic pose information to permit computation of relative links between each of these poses derived from matching laser scan data. This was the first case of global optimisation in the robotics literature. Global optimisation of linear(ised) systems however had long existed within the literature under the term smoothing [27]. Subsequent research increased understanding of the robotics SLAM problem and noticed similarities to problems in other domains such as SfM and Bundle Adjustment (BA) [71].

Formulation of the full SLAM problem as nonlinear least squares and using graphical models permitted greater understanding of the structure of the problem. This also invited methods that repeat the optimisation until convergence to fully exploit the retained information. The required recalculation combined with the increased size of the matrices involved made it an offline problem. First attempts used 'springs' or elastic methods [18, 29] and lacked the attention to sensor and process modelling of the filtering literature but posed the problem as - or noted the relation to - least squares. A more complete mapping to nonlinear least squares and sparsity was shown in GraphSLAM [69]. An online implementation called Exactly Sparse Delayed-State Filters using the structure of a single iteration of a nonlinear least squares solution was developed for 'view-based' SLAM, where landmarks are not explicitly constructed in the state space [22].

Both GraphSLAM and Exactly Sparse Delayed-State Filters used the information matrix as their general storage and accumulation method. An alternate representation of the data

using the matrix square root of the information matrix was proposed by Dellaert and Kaess [15]. This forms the basis of the method we introduced at the start of this chapter and provides an explicit mapping between linear algebra and graphical models. The matrix square root can be found through Cholesky decomposition of the information matrix or through QR decomposition of the Jacobian matrix (after incorporation of the noise models). This form retains the general sparsity of the information matrix and is often used as part of solving a system of linear equations or matrix inversion. Others have implemented variants of this with a strong emphasis on speed of calculation [42]. Methods for fast inclusion of new data in the square root form resulted in iSAM [37] and reusing previous linearisation values and partial reordering when iterating gave iSAM2 [39]. Additionally a new graphical model to describe the conditional probabilites that the square root form represents in a tree structure was developed [38].

2.3 Localisation

Here we review research examining estimation in the context of robotics. We cover developments in localisation and mapping for a single vehicle before considering work in multiple vehicles which parallels the single vehicle case in many instances.

2.3.1 Single Vehicle

Initial solutions to single vehicle localisation can be seen in the Kalman Filter. The earliest robotics solutions used a constrained initial pose at the origin of the state space [10, 67] and landmarks were used as the basis for navigation with interest in convergence properties [17] and inconsistencies [36].

Efforts to handle larger problems led to approaches creating smaller submaps that delayed fusion to keep a small local problem [20, 24, 45, 77] or forcing sparsity of the covariance matrix [58, 70]. The latter approaches were not conservative[1] and sometimes resulted in reduced uncertainty in the initial algorithms although later developments corrected this [72]. Alternate problems in SLAM where laser scans or monocular or stereo images were directly matched led to greater interest in the Full SLAM problem where historic states are

[1] We use the definition of conservative to indicate that the new approximate covariance ellipsoid contains the original exact covariance ellipsoid derived from their respective matrices

retained. The first example was Lu and Milios [49] which also led to the first smoothing solutions. Later examples of the problem used delayed states with the Kalman Filter [21] although realisation of sparsity of the information matrix for the same problem led to improvements [22, 50].

Around the mid 2000's interest in smoothing solutions where the state and map estimates are optimised until convergence gained greater interest as computing power and memory increased and algorithms improved. This led to methods like GraphSLAM [69], $\sqrt{\text{SAM}}$ [15], iSAM [37], T-SAM [56] and HOG-Man [31]. These approaches all recognised the equivalence with nonlinear least squares but varied in methods for representing and updating the problem. GraphSLAM and $\sqrt{\text{SAM}}$ both used batch solutions with the latter favouring the square root of the Information matrix for representation of the problem. iSAM and subsequent extensions used the square root form enabling online updates, variable reordering and relinearisation in real time [39]. T-SAM and HOG-Man instead favoured an hierarchical approach using submaps, where pieces of the problem are solved before being grouped and solved at higher levels, borrowing from earlier methods with filtering approaches. Authors of both collaborated on an improved method using elements of both approaches [32].

Additionally smoothing based methods transitioned to solving for a vector of 'deltas' or differences to the linearisation point as in both $\sqrt{\text{SAM}}$ and HOG-Man based methods. When combined with a change to use manifold or Lie Algebra methods this improved handling of error metrics based on rotation groups $SO(N)$ and pose spaces $SE(N)$ through avoidance of singularities where N is 2 or 3 for planar or full 6-DOF motion.

2.3.2 Cooperative Localisation

Methods for localisation of multiple robots have generally paralleled single robot solutions although they present additional problems in communication range, bandwidth and reliability along with the increased number of states to optimise over.

Very early implementations for cooperative localisation (as distinct from the 'and mapping') split the team of robots into two groups where one group could move for a step and the other was held in place as landmarks [43]. Observations were still sent to a centralised location for incorporation in the Kalman Filter. A number of efforts for multiple robot Kalman Filters used shared cross correlation terms and required synchronised communications to

update each robot [63], or maintained a number of filter banks exponential in the number of systems [6] to enable combination with remote systems estimates and avoid fusion of observations multiple times.

Ultra-low bandwidth usage systems have also been proposed using quantisation of the innovation to amounts as small as a single bit per update [52, 62]. This required using identical estimators on all systems and so limited the ability to use local information at full resolution. A hybrid system that removed this limitation was developed [54] but still retained the fixed linearisation and updating of multiple estimators.

Nettleton et al. [55] identified the information form as being more suitable for multiple robot estimators with early work utilising a channel filter to cache information and transmit in a small batch. Later work with delayed states permitted prevention of repeated fusion through usage of a local solver that transmitted its estimate and uncertainty to a central solver that could be replicated on each platform [8]. The solvers could then subtract out information they had previously received before fusing the updated result. This did retain some of the complexity of the Kalman Filter methods but reduced the storage required to be proportional only to the number of vehicles as opposed to the square of the number of vehicles and enabled delayed observations to be fused.

Investigation of nonlinear least squares approaches and adapting Smoothing and Mapping (SAM) approaches has provided C-SAM [4] which linked separate $\sqrt{\text{SAM}}$ solvers on an intervehicle observation. The method did not investigate bandwidth reduction and instead focussed on solving the relative transform between maps and identifying landmarks included in both vehicles' sets.

DDF-SAM [13, 14] also came from $\sqrt{\text{SAM}}$ but ignored intervehicle observations. Instead the each vehicle shares the marginal probability of a set of common landmarks which are associated separately. Periodically new maps are shared requiring the old maps to be removed to avoid fusing observations multiple times. They used anti-factors to cancel previously shared information from the factor graph before fusing the updated information enabling a single solution to be used and reduce repeated computation.

Another approach is based on condensed measurements [32] where the total graph is divided and solved directly relative to an origin local to each section. The condensed measurements are then made up of approximated measurements of each desired state relative to the sec-

tions origin. Lazaro et al. [44] applied this to the multiple vehicle case to create a single condensed graph per vehicle/vehicle pair which is periodically retransmitted with updates included. The approximation used to condense measurements approximates the true distribution but does not guarantee the new uncertainty is equal to or greater than the actual distribution.

Exploitation of heterogeneous robot teams where there is large disparity in information sources available to each robot have led to asymmetric solutions where information is transmitted in one direction only and can use constant bandwidth per intervehicle observation [76]. They also utilise a one way range measurement made possible with the use of highly accurate, low drift synchronised clocks and known transmission times. This measurement medium also doubles as the communication channel which transmits the broadcast shared information. An extension to handle dropped packets was shown in [73]. It is not clear if agents can retask and periodically change the direction of information flow if the disparity of information changes or if the system would need to restart and discard old data. Additionally the method is highly specialised to the given sensor/communication combination and less useful in a more general setting where ranges may be calculated from a round trip time.

Communication schemes for transfer of information have also been proposed where all measurements and odometry were transferred between systems [46, 53]. These enable centralised equivalent solutions but with potentially significant bandwidth requirements depending on local sensing rates and if vehicles lose contact and need to transmit a large backlog.

2.4 Summary

There is a lack in the literature of exact and general methods based on a smoothing approach to localisation that focus on reducing communication bandwidth. Smoothing methods we surveyed either have restrictions on the communication topology or retransmit the whole map or approximation. Filtering methods in comparison limit opportunity to correct linearisation errors or data association where remote targets were incorrectly identified. We perceive a gap for an exact iterative smoothing approach to multiple robot localisation that does not restrict communication topology.

Chapter 3

Compression and Packetisation

Implementation of a cooperative localisation solution requires a system containing multiple robots, sensors that permit at least partial observation of relative state and a method to communicate data between all robots. Without any of these elements the system cannot optimise across multiple robots online. Any method that performs cooperative localisation should handle management of information to prevent 'double counting' and inconsistent estimates, consider fast solutions to permit online usage and work within the available bandwidth of the system to prevent delays in fusion. Additional design goals include matching the centralised solutions by finding the global minimum of the nonlinear optimisation problem, decentralisation to permit failure of any node and smart distribution of calculations to reduce repetition across robots and minimise bandwidth. Guidance for fast and accurate methods come from single and multiple robot SLAM literature and cooperative localisation literature where graph methods and nonlinear least squares have shown prominence.

This chapter covers the formulation of a method that enables a distributed and decentralised solution of the multi-vehicle problem. We have entitled the method Multiple Vehicle Smoothing and Mapping (MVSAM). In particular the method is based on partially solving a single iteration of the full history nonlinear optimisation problem, distributed elimination of variables local to each system before sharing the result to solve in a decentralised/multi-centralised manner. The solution is exact for a single iteration of nonlinear optimisation without approximations such as sparsification. The calculations performed are largely the same as the centralised version, with the main addition being a constrained elimination ordering and the addition of communication steps. Additionally this solution enables local

relinearisation whilst reusing the previously calculated results of variable elimination for remote state variables. We show that the communication bandwidth is limited, with the upper bound dependent on the state size and rate of intervehicle observations.

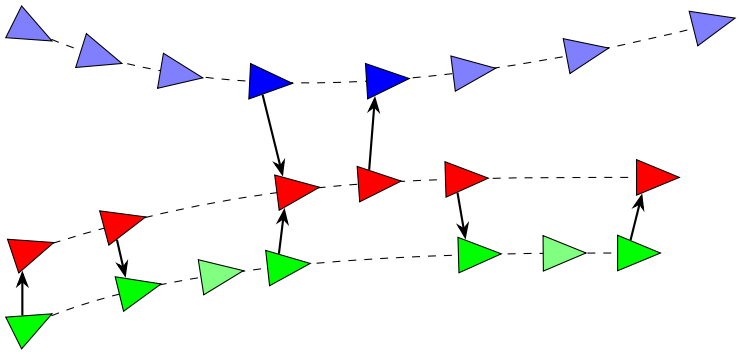
The contribution in this chapter is the novel linearisation ordering, combined with distributed partial elimination that arises from insights gained using the factor graph formulation of the problem.

3.1 Cooperative Localisation

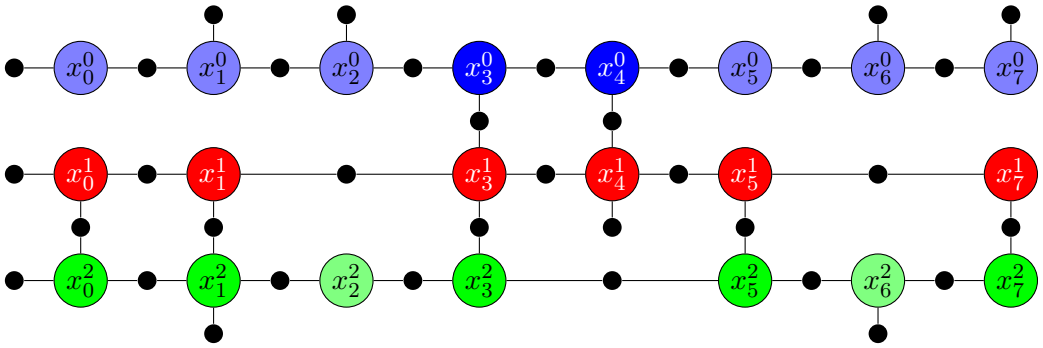
Figure 3.1b repeats the typical factor graph for the cooperative localisation problem from Figure 2.2 with landmark observations removed. It contains factors for local measurements that connect to a single state node, such as position, orientation and velocity observations, and local measurements that connect adjacent nodes that represent vehicle model or odometry links. When compared with single vehicle factor graphs, cooperative localisation factor graphs contain intervehicle measurements that connect two nodes on different robots which include observations of range and bearing between vehicles.

This structure has a long chain of variables local to a single vehicle and relatively sparse intervehicle connections. Computation load can be distributed through the use of variable orderings that eliminate local variables not connected to intervehicle factors, which we will refer to as internal variables, first. It also removes the requirement to transmit or include information about these internal variables that are not observed on any other robot. The factor graph and Bayes tree after elimination of internal variables using an ordering placing each internal state in chronological order is shown in Figure 3.1.

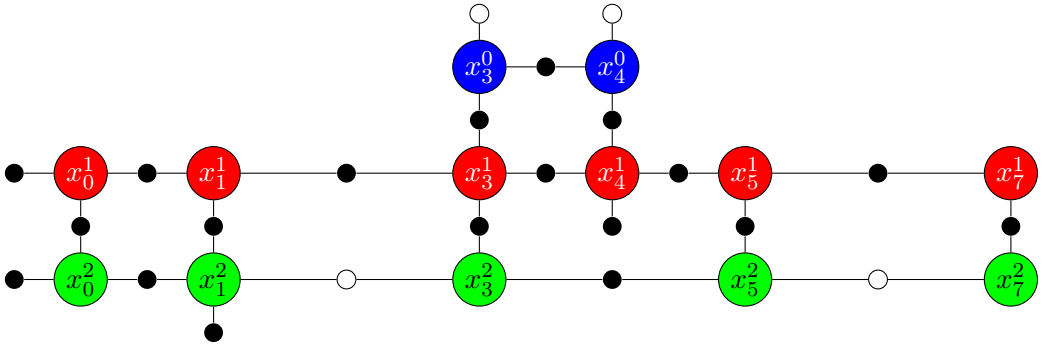
Three important ideas arise from these graphs. The first is that elimination of internal states does not depend on any information from other vehicles as is evident by the fact that none of the tree fragments contain variables from two vehicles. This allows the elimination of these internal variables to take place locally without correspondence. Elimination requires calculation of the Jacobian matrices and error of all factors connected to the target variable. Calculation of these requires fixing the linearisation of all variables connected to these factors. These linearisation points are then fixed for data shared with other robots. Update of the linearisation requires retransmission of newly re-eliminated factors. The estimate produced however is exact and matches a single iteration of the nonlinear optimisation



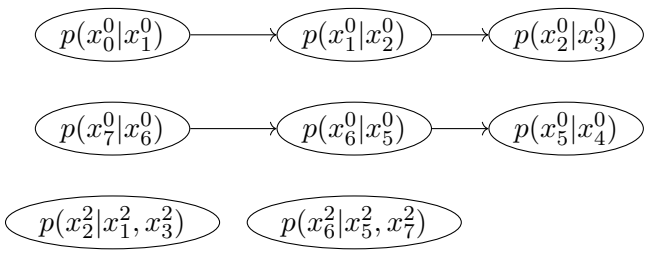
(a) The example vehicle system of Figure 2.1 with landmarks removed. The dashed lines represent vehicle paths with locations of each state marked with a triangle. Arrows between states denote inter-vehicle observations from source to target.



(b) Full factor graph showing all intra and inter-vehicle factors. We remove the landmark states and factors connected to them before elimination here as we are targeting a cooperative localisation solution.



(c) Partially eliminated factor graph. Factors returned from the elimination of other variables are in white.



(d) The leaves of a Bayes Tree generated after elimination of internal states.

Figure 3.1 – The factor graph before and factor graph and Bayes tree after elimination of the internal states. The most important result that can be seen is that the six sets of variables eliminated are not connected to each other. The external states which remain act as a separator permitting independent elimination.

process. Further iterations would differ due to the inability to relinearise data from remote systems.

The second idea is that elimination of states separated by an external state can also be done independently. Fragments are created for states between each intervehicle measurement the vehicle is involved in, and the separator of the root of each fragment refers to only the two states involved in intervehicle measurements either side of the eliminated internal variables. This separator means that no returned factors from the elimination of one fragment are used in calculation of a different fragment (but will be used for elimination for other variables in the same fragment). This permits elimination to take place over all internal variables since the last intervehicle measurement (or since the beginning when initialising).

The final idea that arises is that the conditional probabilities for the internal states do not need to be transmitted as they are only required for recovery of the update for those variables. The Bayes Tree shows that calculation of the update flows initially from the leaves to the root for calculation of the Jacobians and factors and returns from the root of the tree to the leaves for final calculation of the update. In this case the branches that contain only internal states of remote vehicles are not needed as the local states do not depend upon them.

3.1.1 Variable Elimination

Stepping through the manipulation of the factor graph's joint probability $g(\Theta)$ required for variable elimination we can show the ideas of the previous section. This requires extracting all elements that depend upon our target state x_k^j from the joint probability. We then manipulate to create the conditional probability of this state upon its connected variables S_i, known as the separator. This leaves a portion that does not depend on the eliminated variable which is then treated as part of the factor graph again whilst the conditional probability is now treated as part of a Bayes Tree.

To begin the variable elimination of an internal state x_k^j we gather all likelihood functions that depend upon this state variable separately to those that do not as in (3.2). We use the first order Taylor expansion to linearise these factors around our chosen linearisation point

to get (3.4).

$$g(\Theta) = \prod_i g_i(\Theta_i) \tag{3.1}$$

$$= \prod_{i|x_k^j \in \Theta_i} g_i(\Theta_i) \prod_{i|x_k^j \notin \Theta_i} g_i(\Theta_i) \tag{3.2}$$

$$\propto \prod_{i|x_k^j \in \Theta_i} \exp -\frac{1}{2} f_i(\Theta_i)^T \, \mathbf{\Lambda}_i^{-1} f_i(\Theta_i) \prod_{i|x_k^j \notin \Theta_i} g_i(\Theta_i) \tag{3.3}$$

$$\simeq \prod_{i|x_k^j \in \Theta_i} \exp -\frac{1}{2} \left(f_i\big(\Theta_i^0\big) + \sum_j \mathbf{J}_{ij} \delta\theta_j \right)^T \mathbf{\Lambda}_i^{-1} \left(f_i\big(\Theta_i^0\big) + \sum_j \mathbf{J}_{ij} \delta\theta_j \right) \prod_{i|x_k^j \notin \Theta_i} g_i(\Theta_i) \tag{3.4}$$

We can then rearrange the argument of the exponentials in the likelihood functions to extract the update for the target state x_k^j from variables in the separator S_i for each factor as in (3.5). We can combine all likelihood functions we have linearised into a single exponential with concatenated terms f_x, $\mathbf{J}_x$, $\mathbf{J}_s$ and $\mathbf{\Lambda}_x$ that give the function result vector at linearisation, Jacobian matrices and covariance matrix of the grouped factors. We also change to using $S_k = \bigcup_i S_i$ to denote the separator of the combined factor. We further show in (3.7) premultiplying the vector and matrix elements by the matrix square root of the inverse covariance matrix.

$$g(\Theta) \simeq \prod_{i,x_k^j \in \Theta_i} \exp -\frac{1}{2} \left(f_i\big(\Theta_i^0\big) + \mathbf{J}_{ix} \delta x_k^j + \mathbf{J}_{is} \delta S_i \right)^T \mathbf{\Lambda}_i^{-1} \left(f_i\big(\Theta_i^0\big) + \mathbf{J}_{ix} \delta x_k^j + \mathbf{J}_{is} \delta S_i \right) \prod_{i,x_k^j \notin \Theta_i} g_i(\Theta_i) \tag{3.5}$$

$$= \exp -\frac{1}{2} \left(f_x\big(\Theta_i^0\big) + \mathbf{J}_x \delta x_k^j + \mathbf{J}_s \delta S_k \right)^T \mathbf{\Lambda}_x^{-1} \left(f_x\big(\Theta_i^0\big) + \mathbf{J}_x \delta x_k^j + \mathbf{J}_s \delta S_k \right) \prod_{i,x_k^j \notin \Theta_i} g_i(\Theta_i) \tag{3.6}$$

$$= \exp -\frac{1}{2} \left(\mathbf{\Lambda}_x^{-\frac{1}{2}} \left(f_x\big(\Theta_i^0\big) + \mathbf{J}_x \delta x_k^j + \mathbf{J}_s \delta S_k \right) \right)^T \mathbf{\Lambda}_x^{-\frac{1}{2}} \left(f_x\big(\Theta_i^0\big) + \mathbf{J}_x \delta x_k^j + \mathbf{J}_s \delta S_k \right) \prod_{i,x_k^j \notin \Theta_i} g_i(\Theta_i) \tag{3.7}$$

We can transform this single factor into two elements, the first the conditional probability of x_k^j conditioned upon S_k and the second a factor dependent upon S_k but not x_k^j. To achieve this we need to modify the Jacobian coefficient of δx_k^j to be upper triangular through unit determinant orthogonal transforms $\mathbf{Q}$ where $\mathbf{Q}\mathbf{Q}^T = I$. This transform can be found via the QR algorithm, Gram-Schmidt or Householder reflections with $\mathbf{\Lambda}_x^{-\frac{1}{2}}\mathbf{J}_x = \mathbf{Q}\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{0}\end{smallmatrix}\right]$ and furthermore $\mathbf{Q}^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\mathbf{J}_x = \left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{0}\end{smallmatrix}\right]$ This separation of upper triangular and zero component is important to distinguish the conditional probability from the remnant factor. To splitting $\mathbf{R}$ from the $\mathbf{0}$ rows we divide $\mathbf{Q} = [\,\mathbf{Q}_x\ \mathbf{Q}_s\,]$ where the number of columns in $\mathbf{Q}_x$ equals the number of columns in $\mathbf{J}_x$. We insert $I = \mathbf{Q}\mathbf{Q}^T$ into the middle of the exponent before multiplying through and splitting the conditional from the remnant factor.

$$
\begin{aligned}
g(\Theta) \simeq \exp -\frac{1}{2}&\left(\left[\begin{smallmatrix}\mathbf{Q}_x^T\\ \mathbf{Q}_s^T\end{smallmatrix}\right]\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_x\delta x_k^j+\mathbf{J}_s\delta S_k\right)\right)^T\\
&\left[\begin{smallmatrix}\mathbf{Q}_x^T\\ \mathbf{Q}_s^T\end{smallmatrix}\right]\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_x\delta x_k^j+\mathbf{J}_s\delta S_k\right)\prod_{i,x_k^j\notin\Theta_i} g_i(\Theta_i) \qquad (3.8)\\
= \exp -\frac{1}{2}&\left(\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{0}\end{smallmatrix}\right]\delta x_k^j+\left[\begin{smallmatrix}\mathbf{Q}_x^T\\ \mathbf{Q}_s^T\end{smallmatrix}\right]\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\right)^T\\
&\left(\left[\begin{smallmatrix}\mathbf{R}\\ \mathbf{0}\end{smallmatrix}\right]\delta x_k^j+\left[\begin{smallmatrix}\mathbf{Q}_x^T\\ \mathbf{Q}_s^T\end{smallmatrix}\right]\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\right)\prod_{i,x_k^j\notin\Theta_i} g_i(\Theta_i) \qquad (3.9)\\
= \exp -\frac{1}{2}&\left(\begin{bmatrix}\mathbf{R}\delta x_k^j+\mathbf{Q}_x^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\\ \mathbf{0}+\mathbf{Q}_s^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\end{bmatrix}\right)^T\\
&\left(\begin{bmatrix}\mathbf{R}\delta x_k^j+\mathbf{Q}_x^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\\ \mathbf{0}+\mathbf{Q}_s^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\end{bmatrix}\right)\prod_{i,x_k^j\notin\Theta_i} g_i(\Theta_i) \qquad (3.10)\\
= \exp -\frac{1}{2}&\left(\mathbf{R}\delta x_k^j+\mathbf{Q}_x^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\right)^T\left(\mathbf{R}\delta x_k^j+\mathbf{Q}_x^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\right)\\
&\exp -\frac{1}{2}\left(\mathbf{Q}_s^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\right)\left(\mathbf{Q}_s^T\mathbf{\Lambda}_x^{-\frac{1}{2}}\left(f_x(\Theta_i^0)+\mathbf{J}_s\delta S_k\right)\right)^T\prod_{i,x_k^j\notin\Theta_i} g_i(\Theta_i)\\
& \qquad (3.11)
\end{aligned}
$$

At this point the remnant factor is returned to the graph to be used in future variable elimination and the conditional probability is retained separately. Calculation of the delta update δx_k^j occurs after calculation of all δS_k has taken place and these known values are substituted into the conditional probability that was extracted.

It is immediately apparent that provided the set of factors connected to two variables does not overlap then elimination can proceed in parallel. Given an internal first elimination ordering as we propose here the first condition immediately holds true as internal states do not have remote vehicle states in their separator and thus internal states do not require remote information for elimination. Furthermore sets of internal states separated by external states on a single vehicle can also be eliminated in parallel. This is less obvious as external states on the vehicle will be in the separators of multiple internal variables. However no factor directly connects two sets of internal states and thus no factor is involved in elimination of two or more sets of internal states. Finally these local conditional probabilities are not required on remote systems as internal variables do not appear in the separators of any external states and so are not required for recovery of the updates.

With these insights we return to the linear algebra perspective, with the block sparsity pattern of the matrices being shown in Figure 3.2. Here we combine the previously separated concepts of solution matrices (which represent conditional probabilities), and the Jacobian matrix (and its general representation of multi-variate likelihood functions) as the conditional probabilities can be seen as special cases of the generic likelihood functions that factors represent. This ready movement of nodes in the Bayes tree back to the factor graph was initially mentioned in [38] with regard to updating the Bayes tree with the arrival of new factors for the related factor graph and includes a single variable derivation of the conditional probability extraction using pseudo-inverses.

Our method extends this concept further to split the global factor graph into numerous smaller factor graphs in which no factor is replicated. This is depicted in Figure 3.3a with subgraphs being outlined by dashed lines. Notably no factor is included in multiple subgraphs and thus elimination can occur in parallel and no double counting can occur. The variables that occur in multiple subgraphs are the externally visible variables involved in intervehicle measurements and are placed last in the elimination order. Each subgraph is solved in parallel, using the same predetermined linearisation points for external variables which appear in multiple subgraphs. After elimination the elements at the root of the Bayes Tree relating to the external variables, which are typically the first and last variables in chronological order, are returned for use in the global factor graph. Figure 3.4 depicts the subgraphs solution matrix after elimination and highlights the section that is removed using the internal variables first elimination ordering.

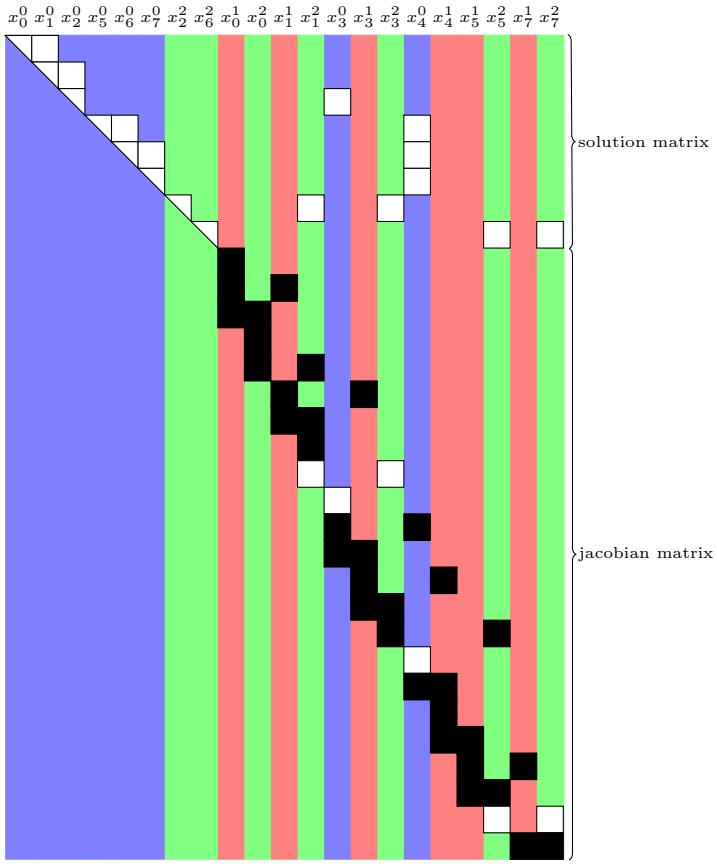
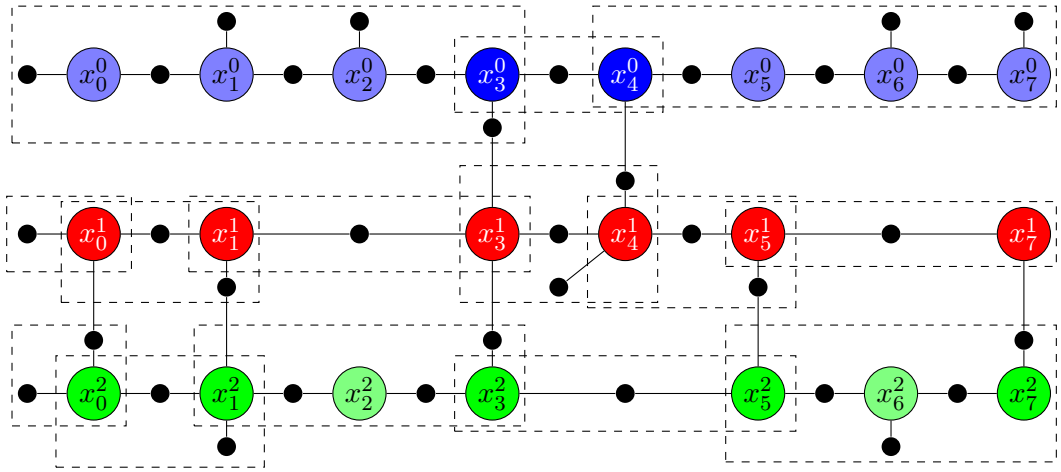
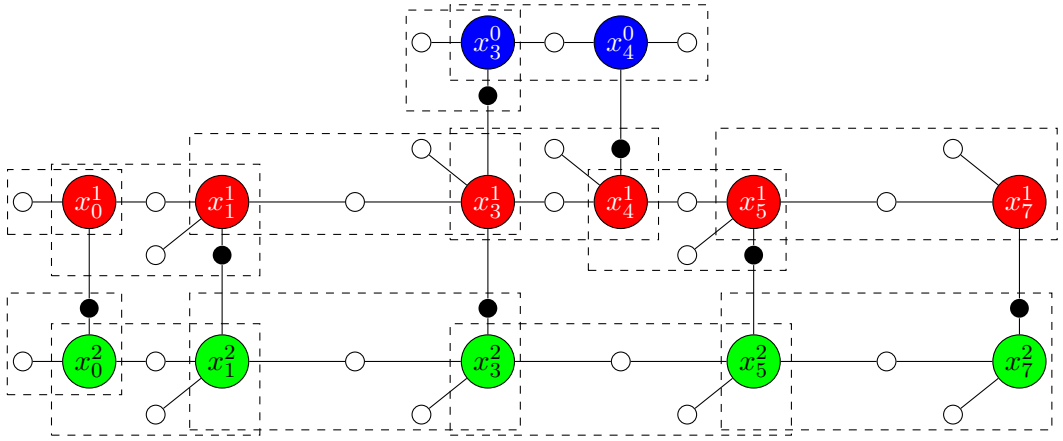


Figure 3.2 – The block sparsity pattern of the combined Jacobian and Solution matrices after elimination of internal variables. White blocks are the results of elimination as either the conditional or returned factor whilst black nodes are factors that have not been touched thus far by the elimination process. Columns are coloured to correspond to the colour of the variable node in Figure 3.1 that this matrix represents.



(a) The factor graph after removal of landmarks with division into subgraphs.



(b) The global graph constructed from combining the intervehicle factors and the results of subgraph elimination with elimination of external states pending.

Figure 3.3 – The complete graph divided into subgraphs before (a) and after (b) elimination of the internal states from each subgraph. The black circles are nonlinear factors, representing sensor error metrics. The small white circles are the linearised and summarised factors resulting from elimination of internal variables. Factors connected to two states of different colours represent the intervehicle observations. The dashed lines enclose the packets of information, showing which factors and linearisation points would be transmitted. States after the last external state on a given vehicle are not part of the shared graph that is constructed - although if desired the information could be transmitted and used to improved the estimates of all vehicles.

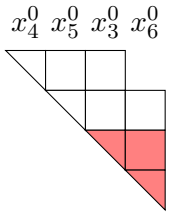


Figure 3.4 – The solution matrix for the elimination subgraph. This is the linear algebra equivalent of the Bayes tree. The highlighted elements are the rows that are passed to the global graph.

The global graph is constructed using the returned nodes from each Bayes Tree and the intervehicle measurement factors. Subgraphs from each vehicle link in a chain through common start and end states whilst intervehicle factors create the links between each vehicle. This reduced graph is shown in Figure 3.3b and is similar to the partially eliminated graph in Figure 3.1c, however factors relating to the external states are also included in the elimination process. All factors in the graph are shared between all vehicles and the solution calculated. The Bayes trees resulting from elimination of each subgraph, which are the same as the fragments in Figure 3.1d, can be attached to the tree created by elimination of the global graph as shown in Figure 3.5 and thus each vehicle can update its local state estimates. As the conditional probabilities are not transmitted each vehicle cannot calculate, and indeed has no information about, internal states from remote vehicles. Each vehicle is running a local estimator that includes the summarised factors from remote vehicles and nonlinear factors for intervehicle and local observations. This estimator supplies the linearisation points for the calculation of the summary packets and they are supplied in a just-in-time manner to enable usage of all information received to that point.

3.1.2 Relinearisation

To this point the description has been of exact solutions to a single iteration of the global nonlinear least squares optimisation problem. Nonlinear optimisation requires iteration of the solve, linearisation update and recalculate factor steps until convergence. Recalculation of the factors used in subgraph elimination, and thus updating of the linearisation of remote vehicle states cannot take place without transmitting the nonlinear factor data alongside the eliminated data which undermines the data savings of distributed elimination.

Alternatively these subgraphs can be recalculated on their origin vehicle and transmitted replacing the previous data - but this results in increased bandwidth. Vehicles can recalculate their local factors and all intervehicle factors as they have the raw measurement data. This permits relinearisation of all local states for the internal solver whilst holding fixed the linearisation points of remote vehicle states. Furthermore if using iSAM2 [39] or other graph solver that permits partial relinearisation not all local variables will require updating at each time.

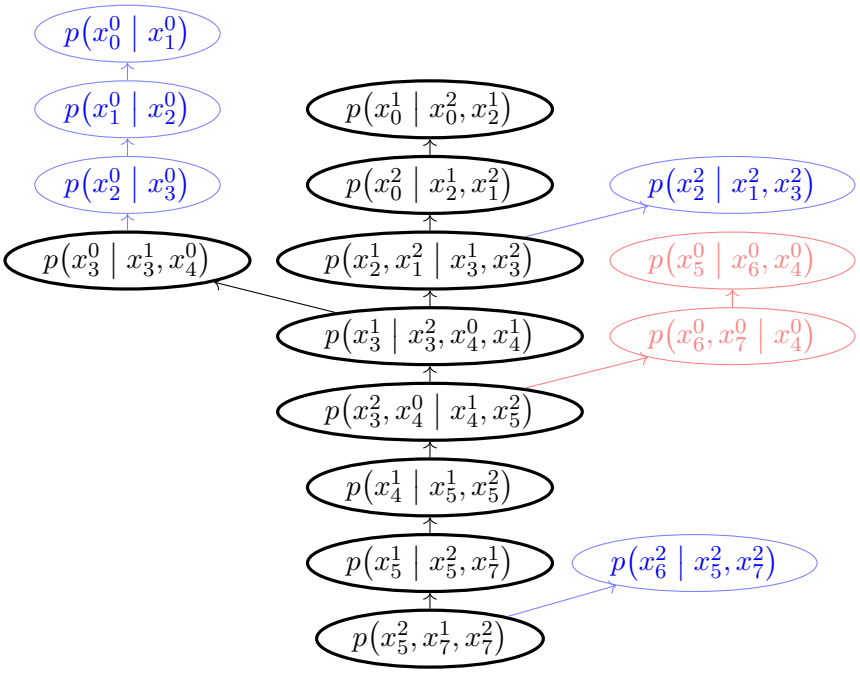


Figure 3.5 – The Bayes tree after elimination of the graph in Figure 3.3b with the newly calculated conditionals in black. The fragments of the tree that are retained from subgraph elimination are shown in blue out with the red fragment representing the states that exist after the last external state and so would not yet have been transmitted. In comparison to the random elimination ordering the cliques in this tree are smaller and thus enable faster calculation.

3.1.3 Indeterminant Subgraphs

Calculation of the subgraphs on each vehicle containing internal states involves solving a smaller section of the overall problem. Whilst we assume that the global problem is of sufficient rank we do not make the same assumption about the subgraphs. The global graph is generally either fixed at the origin (per vehicle) at the start time with offsets between vehicles being calculated or sufficient observations are made relative to the global frame to provide constraint in all dimensions. After this initialisation period further states may not be constrained to the origin and the global frame may not be observable for all dimensions (for example GPS denied environments). Rank sufficiency is maintained globally through motion model links back to the initial states.

The lack of constraint in later states does not pose a significant issue for centralised problems which eliminate with the initialisation measurements present. In contrast our decentralised method partially eliminates each subgraph and so must handle cases where the subgraph is singular. Given the structure of the vehicle localisation problem with a fully constrained vehicle motion model the maximum deficiency of the Jacobian for any segment is the size of the vehicle state n, and will occur when no observations were made outside of motion model updates. Additional measurements reduce this deficiency but require sufficient variation in the state elements that are being observed and depend on the structure of the motion model.

Due to potential singularities or rank deficiency the elimination cannot use a Cholesky based method and must instead use a rank-revealing decomposition such as QR factorisation with pivoting. Using a rank revealing method can assist in removal of rows that are non-zero due to numeric computation precision issues by investigating the leading diagonal entries on each row. In this way further numerical errors and transmission of these rows can be reduced.

We demonstrate the complexity in predicting rank deficiency and handling reduction of these subgraphs using QR factorisation on a simple 2 dimensional constant velocity model. We show Jacobians for two sequential state segments with combinations of position and velocity measurements on either state along with a single forward prediction between the two states. Without loss of generality we demonstrate this with unit noise parameters[1] and $\delta t = 1$ to enable cleaner display of the matrix contents. The Jacobians premultiplied by

[1]Whilst the noise matrix for direct observations is a single element matrix with a value of 1, the process noise matrix is a 2×2 matrix valued function of the time change multiplied by the noise parameter.

$$\begin{bmatrix} -2\sqrt{3} & -\sqrt{3} & 2\sqrt{3} & -\sqrt{3} \\ 0 & -1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = Q \begin{bmatrix} \sqrt{13} & \frac{6}{\sqrt{13}} & -\frac{12}{\sqrt{13}} & \frac{6}{\sqrt{13}} \\ & \frac{4}{\sqrt{13}} & -\frac{3}{2\sqrt{13}} & -\frac{5}{2\sqrt{13}} \\ & & \frac{\sqrt{7}}{2} & -\frac{3}{2\sqrt{7}} \\ & & & \frac{\sqrt{3}}{\sqrt{7}} \end{bmatrix}$$

(a) Update and two position measurements.

$$\begin{bmatrix} -2\sqrt{3} & -\sqrt{3} & 2\sqrt{3} & -\sqrt{3} \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = Q \begin{bmatrix} 2\sqrt{3} & \sqrt{3} & -2\sqrt{3} & \sqrt{3} \\ & \sqrt{2} & & -\frac{1}{\sqrt{2}} \\ & & & \frac{\sqrt{3}}{\sqrt{2}} \end{bmatrix}$$

(b) Update and two velocity measurements.

Figure 3.6 – Jacobians before and after elimination. Each variant shows the Jacobian as calculated from each factor with the noise model included on the left and the R matrix resulting from QR decomposition of this matrix.

the inverse square root of the noise matrix and the final decomposed matrix are shown in Figure 3.6 for the illustrative combinations.

Cases such as a single extra measurement are immediately apparent as singular due to an insufficient row count in the Jacobian. Additionally repeating measurements at each time step does not assist as these measurements are not linearly independent of each other - although they will reduce uncertainty on the measured variable. The obvious case where the system is not singular is where each dimension is directly observed at all times. Further thought intuits that a single measurement of all state elements at any time will also provide a full rank solution even if the measurements of each element are at different times.

The two interesting cases left in our simple system are when measurement of the same state element is repeated at different times. The two state elements we use are position and velocity with the Jacobians and their reduced upper triangular forms being shown in Figures 3.6a and 3.6b respectively. Measurement of position at two different times supplies sufficient information to estimate velocity (distance over time) where observation of velocity at a single time does not. Measurement of velocity at two different times can provide information about distance covered (which is already in the motion model) but does not enable estimation of global position, hence the rank deficiency.

Models with greater complexity can inhibit prediction of rank deficiency. This complication

can be through either nonlinear elements such as motion in a plane or volume with associated rotations. Additionally this can arise in the cooperative localisation scenario where a packet is dropped and connection of current states on a remote vehicle to its own initialisation period is broken. Further inclusion of new packets requires either receiving the missed packet or gaining sufficient information to enable elimination of the new remote chain of states.

3.1.4 Iterative Solving

The final issue is fast inclusion of new data, or iterative inclusion of new local factors and updated estimates. Online optimisation requires that these be fused rapidly to supply near real time updates to the vehicle for control and planning. Research in graph solvers provide a number of options that either optimise for rapid complete solutions [42], hierarchical distribution of the problem [31, 32] or iterative updating of the inverse square root form [37, 39].

Whilst all of these methods are likely suitable as local optimisers for our method we utilise iSAM2 [39] as a number of concepts such as partial relinearisation and exploitation of the tree structure in our method are borrowed from it although applied with different aims and heuristics.

3.1.5 The Algorithm

Algorithm 3.2 covers creation of the packets containing the nodes extracted from the sub-graphs whilst Algorithm 3.1 contains the entire system. Each vehicle uses iSAM2 locally to estimate its own position, although other nonlinear smoothing algorithms can work. Separately on each platform a factor graph is calculated using all factors since the last intervehicle measurement with the linearisation point for the states taken from the local estimator or reused if contained in a previous packet.

This factor graph is eliminated, with the external states eliminated last. The rows relating to these states are extracted from the solution matrix and transmitted to remote vehicles as fixed linearisation factors. When vehicles receive these factors they are included in the local graph estimator and the variables they target are not permitted to relinearise

Algorithm 3.1 Localisation on vehicle i

create local estimator $\mathcal{F}_l$
$j \leftarrow 0$
loop
  create empty packet p_j^i
  create subgraph $\mathcal{F}_j$ in p_j^i
  repeat
    get new local factors f_i
    $\mathcal{F}_l \leftarrow \mathcal{F}_l \cup \{f_i\}$
    $\mathcal{F}_j \leftarrow \mathcal{F}_j \cup \{f_i\}$
  until receive shared factor f_j
  calculate best estimate for local states Θ_i
  $p_j'^i \leftarrow$ compress packet p_j^i using Θ_i
  exchange packets with remote vehicle k
  add $\mathcal{F}_C$ from new packets $p_j'^k$ to $\mathcal{F}_l$
  $\mathcal{F}_l \leftarrow \mathcal{F}_l \cup \{f_j\}$
  $j \leftarrow j + 1$
end loop

Algorithm 3.2 Compress Packet

Require: $\Theta_i, \mathcal{F}_j$ from p_j^i
linearise $\mathcal{F}_j$ at Θ_i from p_j^i
for all internal states x_k^i **do**
  eliminate x_k^i from $\mathcal{F}_j$
end for
copy external state subset Θ_C from Θ_i
$p_j'^i \leftarrow (\mathcal{F}_C, \Theta_C)$
return $p_j'^i$

as the condensed factors they are connected to cannot be recalculated around an updated linearisation point. The transmission of the raw/nonlinear intervehicle measurements enable all local variable linearisations to be updated as the raw/nonlinear data for all factors connected to them is known. Remote state linearisations however must remain constant.

The rows of the solution matrix that are extracted are of a fixed size and depend upon the dimension of the vehicle state. Given a state size n and a fully determined subgraph the last $2n$ rows of the upper triangular matrix are the rows relating to the two external states. Elements below the diagonal are zero and so do not require transmission. Additionally the error vector relating to these rows, the linearisation and time of the latest state need to be transmitted. The linearisation and time for the previous external state that is linked in this packet is unneeded as it was transmitted previously and is unchanged. The result is a maximum number of floating point numbers per packet and is shown in (3.12). If the subgraph is rank deficient the lowest rows of the upper triangular portion are zero and can be omitted, reducing the data transferred. In comparison the transmission of all observations requires the data and time of each observation, along with sensor type and origin vehicle. Transmission of all observations has no inherent upper bound, but also has no inherent lower bound - if no observations occur no data needs to be transmitted beyond protocol demands which could be a single float and a bit to denote time to which no extra data has been received.

$$\text{datasize} = 2n^2 + 4n + 1 \tag{3.12}$$

We compare our compressed packets for a number of state sizes n against raw transmission of single dimension observations. We indicate the number of floats required per subgraph in Figure 3.7 at varying numbers of intra-vehicle observations per intervehicle observation.

3.2 Results

In showing performance of this method we compare against an iterative centralised graph solver on which this method is based (iSAM2). We treat this method as a lower bound on the expected error. An upper bound is then placed on the expected error by using each vehicle's solution independently, without information from the inter-vehicle observations. The usage of the individual solution is to give context to any difference in solution error between

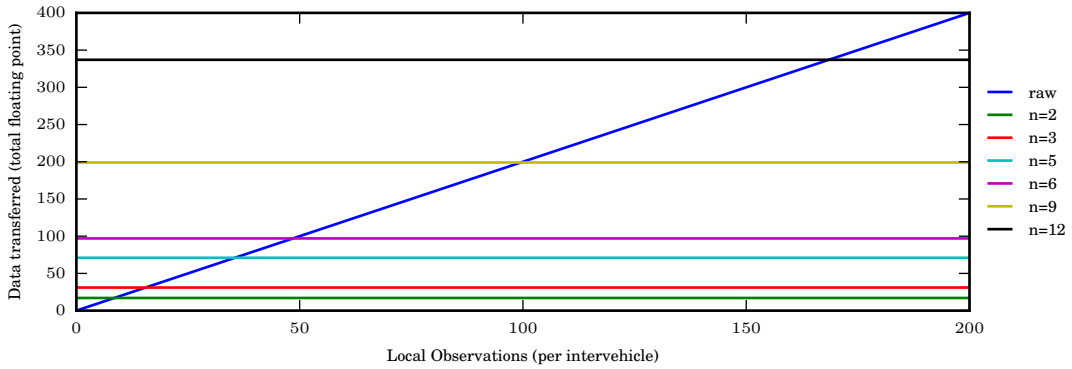


Figure 3.7 – Comparison of number of floats transmitted per subgraph for compressed packets of different sizes (determined by n the dimension of the vehicle state) and transmission of each local observation at varying rates. At low local sensing rates relative to intervehicle sensing rates compression takes up more bandwidth, eventually becoming equal and saving bandwidth at much higher local observation rates. The strange unit of local observations per intervehicle observation is used as this is the base ratio that is applicable across different vehicle designs and scenarios. Where aerial vehicles may have high sensing rates for control slow moving underwater or ground vehicles may have lower sensing rates due to less movement. The important ratio is, for all of these designs, the accumulation of local data relative to intervehicle data.

our method and the centralised solution. Additionally the bandwidth is demonstrated and compared against raw transmission of the measured values.

As we have already shown the method is derived from a single iteration of a nonlinear solver. Thus where the problem is linear or the linearisation point is close to the optimal value the result should closely match the centralised solution. The difference between the centralised solver and our decentralised/distributed method should be where the problem is nonlinear and the choice of linearisation point is not close to the true value. This can be most evident after long durations with denial of absolute/global state measurements like GPS or in fusion of loop closures where large jumps in state estimates can result. If the linearisation point is far from the true value the calculated Jacobian may be significantly different from the true Jacobian and prevent convergence or find local minima different from the minimum we are searching for.

Other methods were considered for comparison, but were deemed to not match the problem or result in issues with direct comparison. Methods derived from a Kalman or Information Filter approach typically solve in Euclidean space, which suffers from singularities in nonlinear problems [48]. This has been addressed in most graph solvers where the value solved for is a small delta to the linearisation point all of which may be expressed using manifolds and their tangents or Lie Groups and Lie Algebra [2, 31]. The Euclidean and delta approaches result in varying models and error metrics when the state isn't well approximated by Euclidean space and usage of appropriate models for each approach results in larger differences than the method. Cooperative localisation graph solvers exist however often constrain the communication topology to be source and destination rather than bidirectional sharing [23]. Whilst these are useful for target tracking it prevents application to group GPS denial or relaying information through an intermediate.

We demonstrate our method on a simple set of problems that are relevant to field robotics - groups of robots moving in the plane. The two scenarios we demonstrate are tracking, where some vehicles have estimates of absolute position and vehicles observe the range/range bearing to remote systems and share all information, and a GPS denied environment where after an initial observation no vehicles obtain absolute position observations.

We use Monte Carlo simulation to show the effects of varying linearisation quality upon the solutions, varying the parameters relating to orientation, such as orientation observation frequency, accuracy and variance in the vehicle model, as it is linked through nonlinear

Figure 3.8 – Robots and environment for the UTIAS dataset. Image courtesy of Autonomous Space Robotics Lab at the University of Toronto.

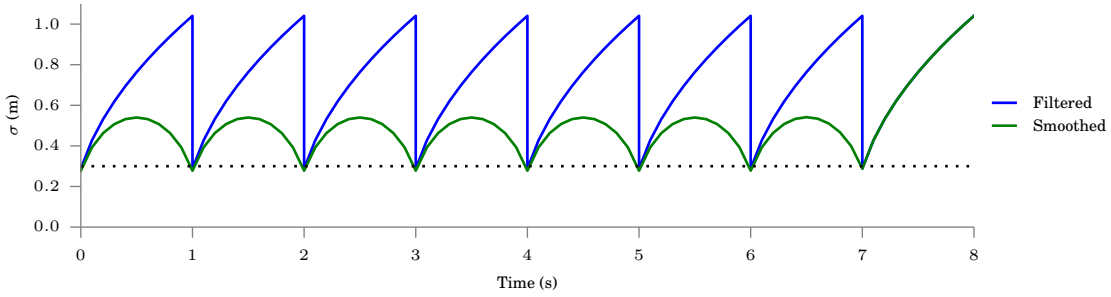
dependencies with other state variables. Each combination of these parameters is simulated 50 times on the same path and each optimiser is run on each simulation before the root mean square error of position and heading is calculated.

Additionally we use subsampled data from the UTIAS multi-robot dataset [47] to demonstrate real world versions of each scenario. An image of the environment for the dataset is in Figure 3.8. Subsampling was used to reduce the high frequency ground truth from the Vicon system to a closer approximation of GPS and compass measurement frequencies.

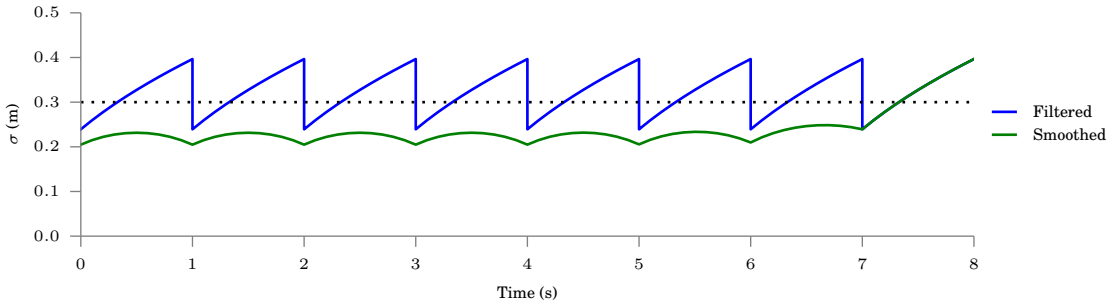
3.2.1 Standard Deviation of Toy System

To better understand the error that we expect with different linearisation points we use a toy system with a single variable in the state space and propagate forward over time with periodic measurements. This enables calculation of the standard deviation which for normally distributed observations and propagation model approximates the root mean squared error about the mean estimate.

Time evolution of the standard deviation is shown in Figure 3.9 for an online filter estimator and an offline batch solution for two different motion model variances. Observations of the state are made at regular intervals and used for both the filtering and smoothing solutions. Steady state variance of the Kalman filter post-observation can be calculated using the sensor period, sensor variance and motion model variance and is independent of the initial variance. This steady state variance is used for an initial observation for the filter and



(a) Large process/motion model noise.



(b) Small process/motion model noise.

Figure 3.9 – Filtered and smoothed estimator standard deviation for two different motion variances over time. The dashed line represents the sensor's standard deviation. Of key note in these graphs is the gain in certainty by usage of 'future' observations in the smoothed estimator.

the batch solvers enabling the effects of initialisation and stabilisation to be omitted and permit observation of long term and 'current state' differences between the solvers. The most immediate difference between the filtered and smoothed solutions is the ability of the smoothed solution to use the impending measurement to further constrain the estimate, resulting in the hump shape instead of a near saw-tooth shape.

Figure 3.9a shows a relatively loose vehicle model, and the filter and smoothed solutions show approximately similar standard deviations upon each observation, but the ability of the batch solution to use 'future' information and constrain the standard deviation growth between observations of the state is apparent. Figure 3.9b shows the same set of observations with a smaller motion model variance (10% of the previous) representing smaller movement uncertainty between observations. The most apparent difference between the two cases is the significant reduction in uncertainty of the smoothed estimator relative to the filtered one. Whereas with a less constraining vehicle model the uncertainty is derived primarily from the previous and next measurements of the state, or just the last one for the filtered case with a tighter constraint from the vehicle model the measurements before the previous and after the next also influence the uncertainty. As the batch solver updates historic state estimates using new information it enables a significant reduction in overall uncertainty for these states. Towards the last time solved however the smoothed solutions gracefully degrade to match the filtered solution with the absence of more 'future' measurements.

In this way the batch solver can be thought of as having two variance modes - the filtering mode in the final intervals, and a reduced uncertainty mode for historical states. In either linear or fixed linearisation systems, or full relinearisation batch solvers these two variance modes are not important. Linear systems have Jacobians that are independent of the state, whilst linearisation is not updated with fixed linearisation systems and in either case the Jacobians will not be recalculated meaning the calculated uncertainty is unchanged. In full relinearisation systems all the Jacobians will be recalculated either at each update, or as updates diverge too far from the calculated linearisation point. For either fixed linearisation or relinearising solvers the difference between batch and filter estimators is not significant, as they fall near exclusively into one or the other for the purposes of calculating linearisation points (a relinearising filter and a fixed linearisation batch solver are both unusual, where fixed linearisation filters and relinearising batch solvers are common).

The method we propose here falls in a middle ground where states can be iterated until

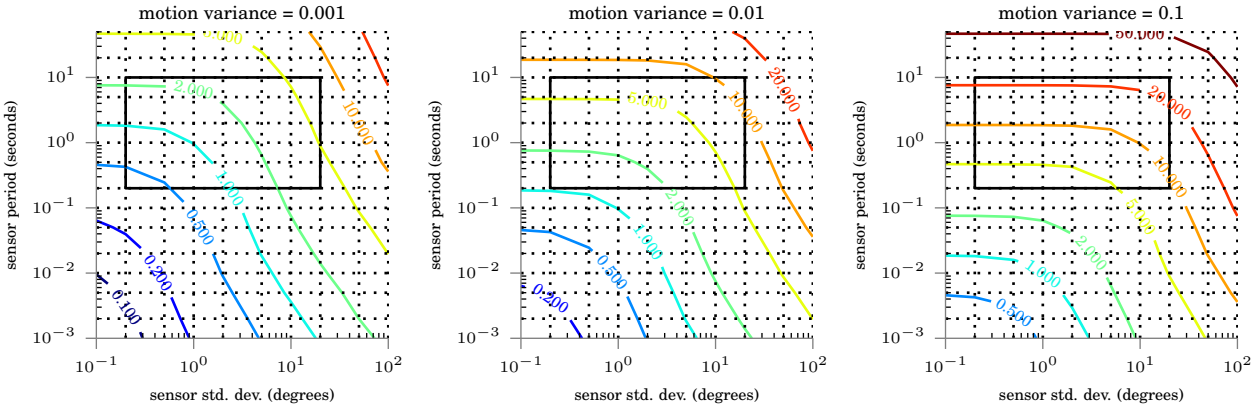


Figure 3.10 – Comparison of single variable smoothing showing contours of constant RMS standard deviation for a range of sensor uncertainties and frequencies. The black rectangle corresponds to the values we are exploring with Monte Carlo simulation in the following sections.

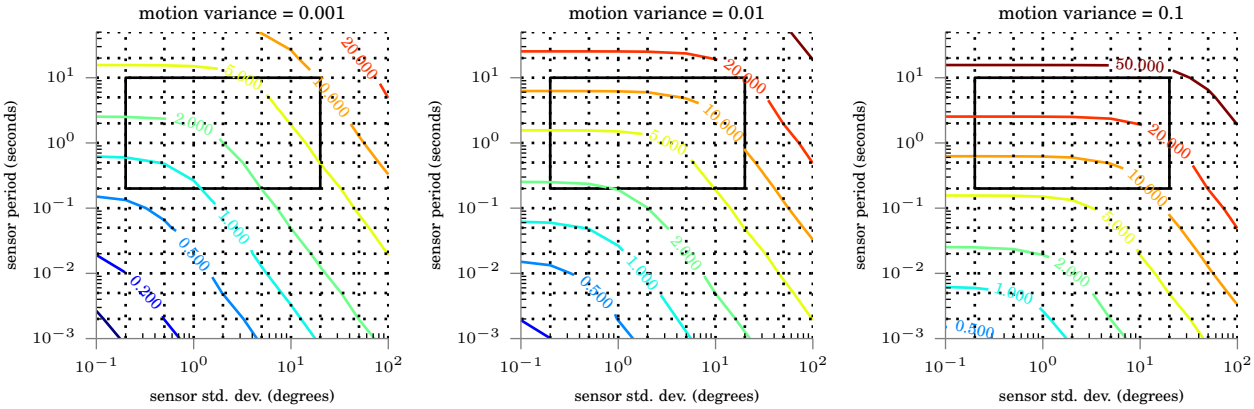


Figure 3.11 – Comparison of single variable filter/online optimisation showing the contours of constant RMS standard deviation for a range of sensor uncertainties and frequencies. The black rectangle corresponds to the values we are exploring with Monte Carlo simulation in the following sections.

convergence pending an intervehicle measurement occurring that requires the linearisation point for historic states to stay constant for shared information. If the intervehicle sensor interval is significantly larger than the local state sensor intervals then the linearisation points for older states will have ideally converged closer to the true state resulting in a more accurate Jacobian calculation. If the intervehicle sensor interval is smaller than the local state sensor intervals we would expect less convergence to the true state and thus poorer linearisation points and less accurate Jacobians calculated from them, resulting in larger system error overall.

Given the single state, single vehicle system we can calculate an 'average' (root mean square)[2] standard deviation over time for combinations of vehicle model variance, sensor variance and sensing intervals. This enables calculation of contour lines showing which sensors with different parameters result in similar overall uncertainty. Results of this are shown for both the filtered and batch solvers in Figure 3.10 and Figure 3.11. The bold rectangle shows the region that will be tested in simulation later and covers reasonable sensor parameter ranges with the area around the box being shown to better illustrate effects of each of the parameters.

Overall the graphs show standard deviation increasing as the sensor variance and sensing interval grow. The underlying determining vertical variable can be shown to be the ratio of period to motion model variance (more accurately the growth in variance per sensor period). Shifting each of the three graphs in Figures 3.10 and 3.11 vertically results in the same values.

Due to the logarithmic scale the standard deviation barely appears to change with low noise values combined with large intervals and this is shown by the near horizontal contour lines. For these sections the standard deviation is dominated by the motion model variance and growth over time in comparison to the relatively small increase in minimums that the sensor variance provides. That there is not a similar set of near vertical lines as sensing intervals shrink show that reductions in periods are not subject to the same diminishing gains that are present for reducing the sensor variance.

In a real system the motion model is a function of the path, or behaviour of the vehicle and so using a constant physical system with varying model parameters should not present root mean square error from truth as clean as the standard deviation calculations here. We would anticipate higher errors where the motion model is too constrained, especially where sensing periods are larger and the model resists change. Where the motion model is overly loose, permitting larger fluctuations we would expect the errors to increase from model mismatch, although not significantly.

[2]Essentially the square root of average variance.

3.2.2 Vehicle Tracking

In the tracking scenario there are two groups of vehicles - those with global position sensor information and those without. For our scenarios all vehicles have orientation and odometry information and measure other vehicles relative location using range sensors. This maps well to cases like underwater robotics where ships on the surface obtain GPS measurements and then observe through acoustic methods the relative locations of submersibles that have accurate onboard odometry and orientation information.

3.2.2.1 Simulated Data

We simulate a test scenario with four vehicles moving in the plane with two of these vehicles having absolute position sensors in addition to the common velocity and orientation sensors. We vary the noise and frequency parameters for the orientation measurements and the orientation component of the motion model as well. We simulate 126 different combinations (3 motion model values, 6 sensor frequencies and 7 sensor variance values) 50 times each over the same path.

We analyse the heading error compared to ground truth for a centralised solver and our method in Figure 3.12. The heading error is taken as the RMS of the online heading error from all samples and all times after an initial stabilisation period of 100 seconds. This can be compared back to the estimate of the single variable solution in Figure 3.11 and Figure 3.10. An example of the heading error, and 3σ bound for low and higher noise scenarios are shown in Figure 3.14 along with related paths in Figure 3.13.

The general shape of the solution matches the expected shape for both our decentralised method and the centralised method - although with slightly steeper lines for higher noise, lower period sensors indicating that increased sensor frequency tends to not help, and is potentially dominated by noise in sensors. Additionally the error in the decentralised method compared to the centralised is very close in the lower noise regions only increasing as the sensor noise and period grow above expected regions for real world sensors (interval up to 5 seconds, sensor standard deviation of 5 gives error less than 0.5 degrees). Further the error to the centralised solution is less when compared to error to ground truth as the local sensing rate increases.

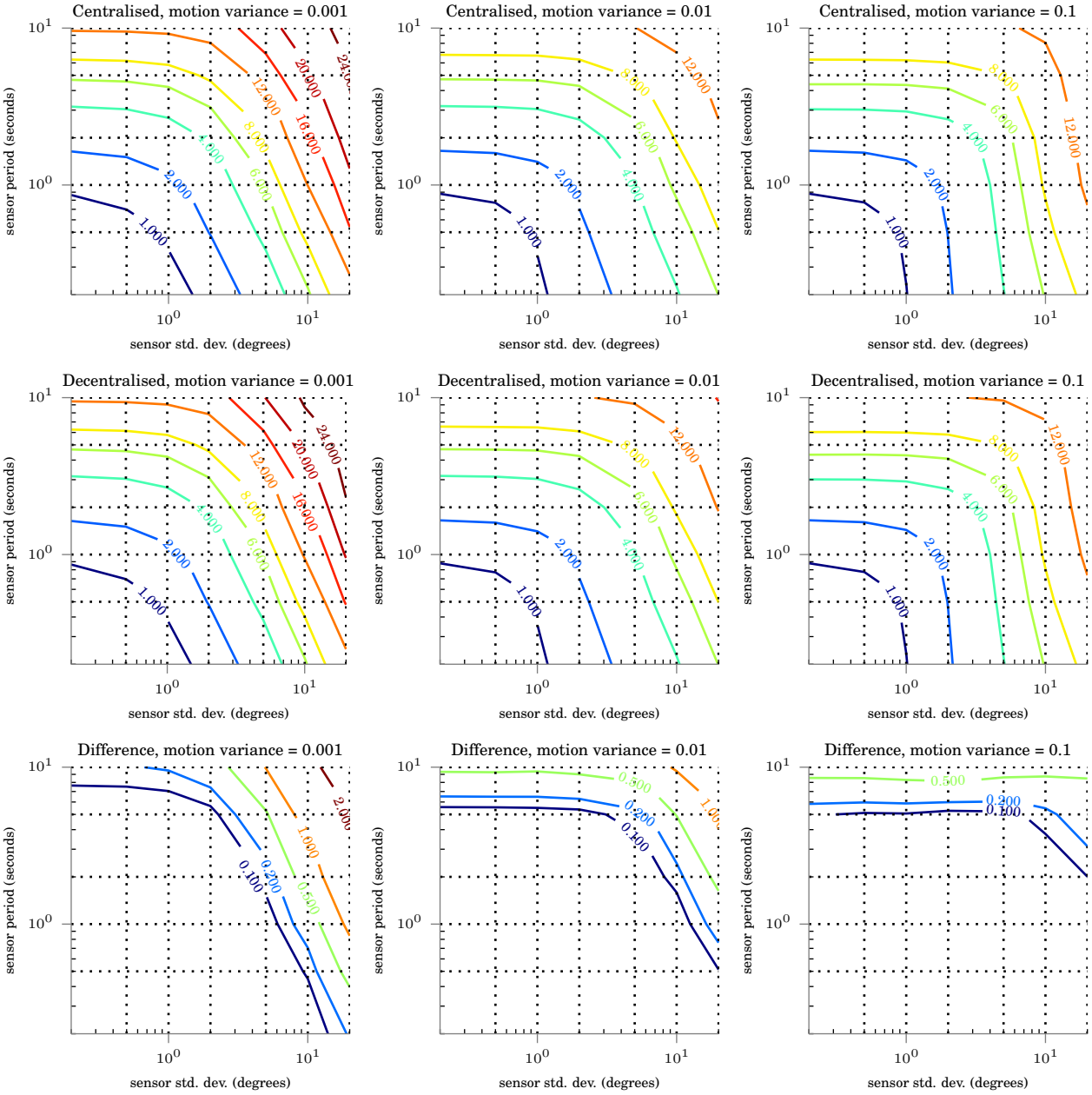
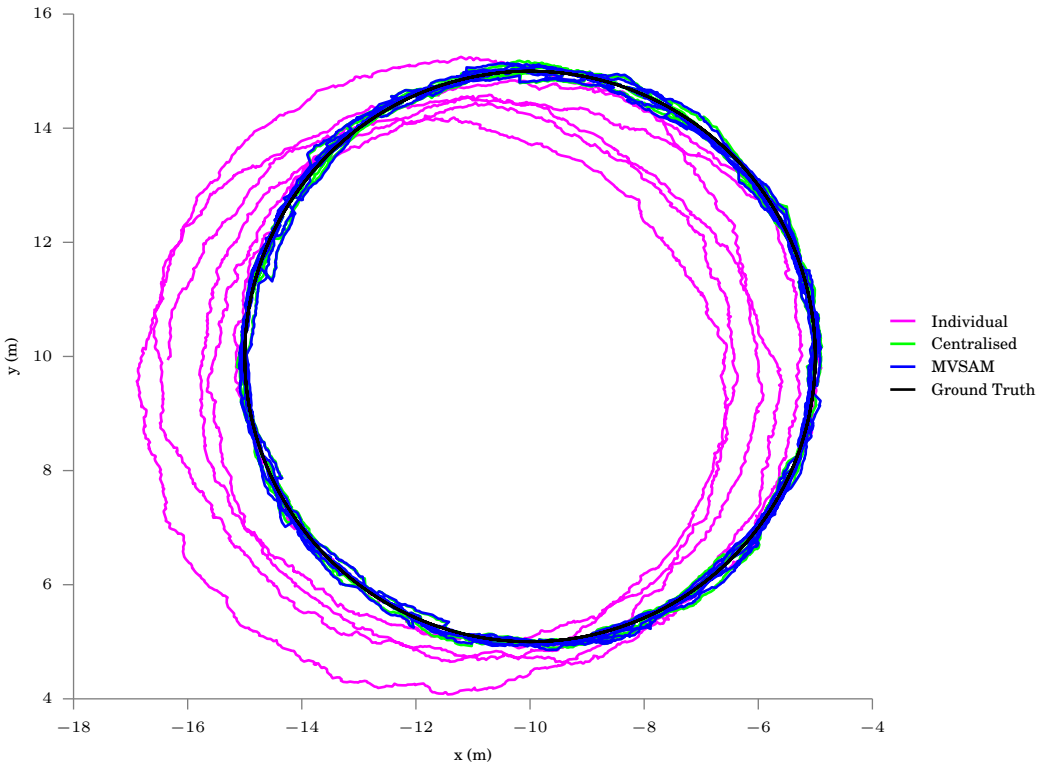
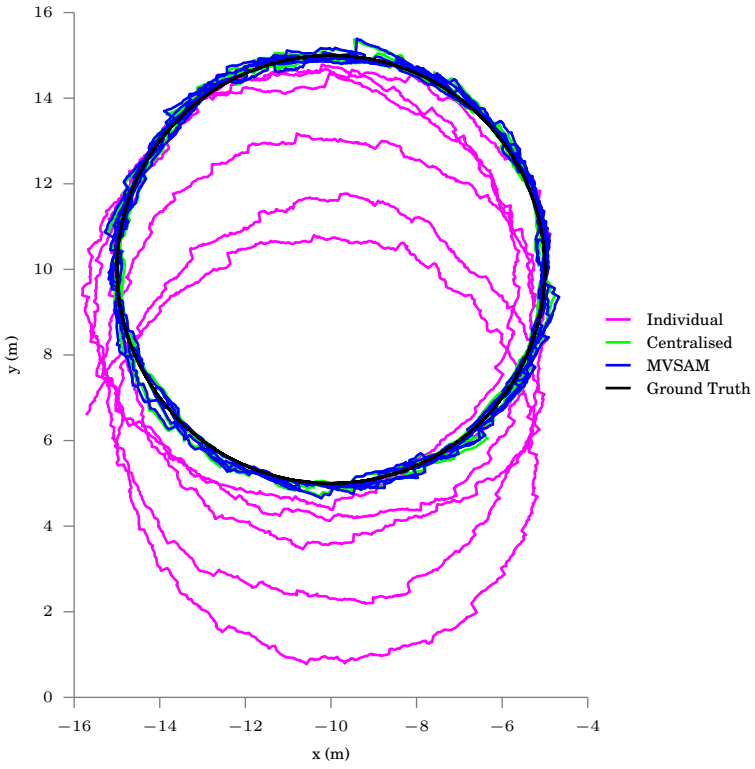


Figure 3.12 – Comparing the RMS error of vehicle heading (in degrees) for a variety of orientation sensor periods, variances and motion model variances. Key observations are that the magnitude of the difference between solutions is less than that to the ground truth, and that in the frequent, low noise observation region where most systems operate (interval up to 5 seconds, standard deviation of sensor noise up to 5 degrees) the difference to the centralised method is less than 0.5 degrees, and less than 0.1 degrees for more accurate sensors.

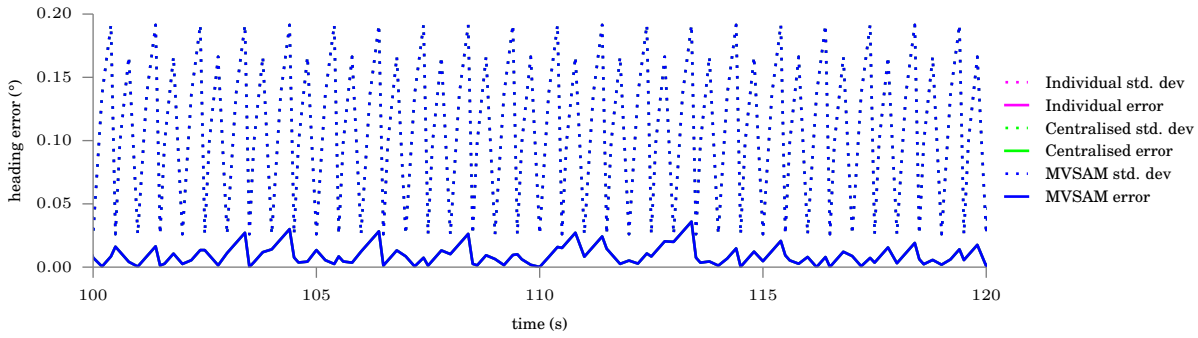


(a) Online path estimation for low noise.

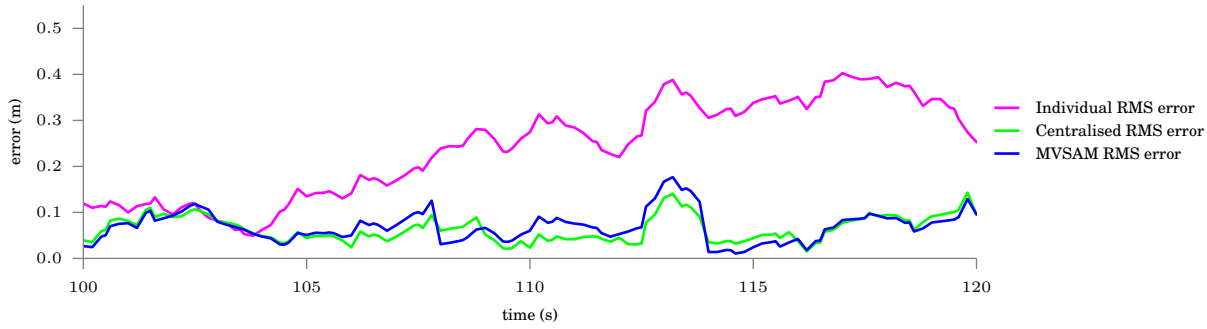


(b) Online path estimation for high noise.

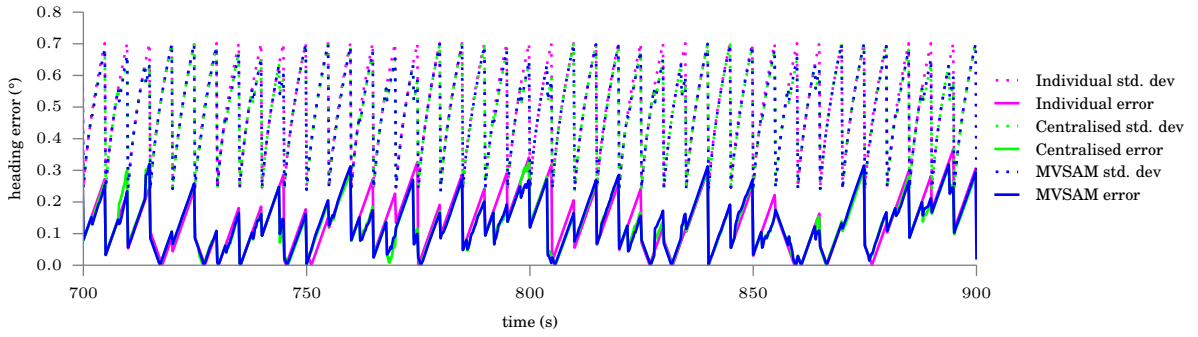
Figure 3.13 – Samples of simulated paths for one of the tracked vehicles in a low (0.5s period, 0.5° standard deviation) and high noise (5s period, 5° standard deviation) environments. The position and heading RMS data for these samples is shown in Figure 3.14.



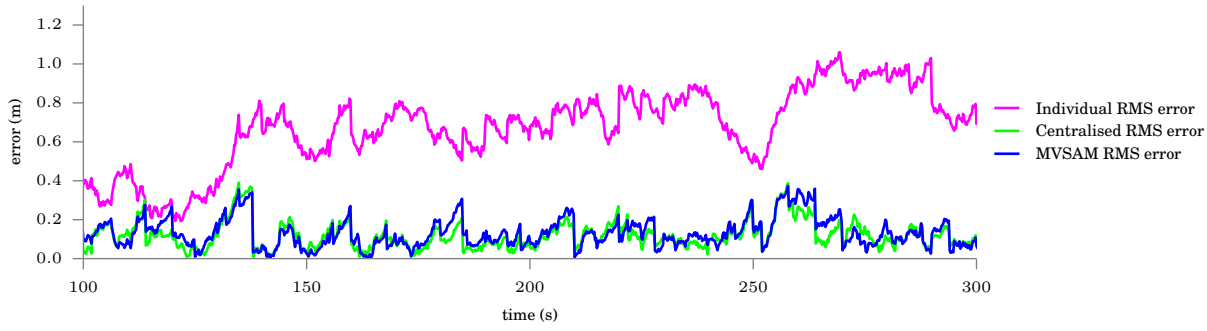
(a) Segment of low noise sensor absolute heading estimation error and 3σ bound. The values here overlap and show no difference between each solution.



(b) Segment of low noise sensor vehicles position RMS error.



(c) Segment of high noise sensor absolute heading estimation error and 3σ bound.



(d) Small segment of high noise sensor vehicles position RMS error.

Figure 3.14 – Samples of simulated data for one of the tracked vehicles in a low and high noise environment. These errors correspond to the sample paths in Figure 3.13. What is barely appreciable from these graphs is the difference between the centralised and summarisation methods - with the expected differences being of the order of 0.01m and 0.2°.

Figure 3.15 shows the RMS position error for the same scenarios taken from the end of each scenario. Noise in these figures is much larger as it is taken from a smaller time section and the relationship is much more complex with other sensors in the system. At low orientation sensor noise/periods the position noise is dominated by the velocity and intervehicle sensor information creating a noise floor that increased orientation accuracy cannot reduce. This shows a similar pattern to the heading, with the difference in error between the centralised and our decentralised method being small in low noise regions, growing to be more significant as the sensor period and variance increase. Both of these figures agree with the concept that the decentralised partially fixed linearisation method and a centralised relinearising method differ most when the linearisation point diverges from the true value.

3.2.2.2 Captured Data

Showing real world performance for this scenario we use the UTIAS MRCLAM dataset [47]. This dataset is a multiple robot dataset captured with five Roombas in a series of nine runs. Each Roomba has a single calibrated camera for performing range and bearing observations of other robots and landmarks. Additionally the motor commands are recorded and given as forward and angular velocities. Ground truth is supplied from a Vicon system that gives both position and orientation with high accuracy for the robots and landmarks. We extract illustrative results from our experiments here but have more complete coverage in Appendix A.

For our heterogeneous configuration we chose the first and third vehicles to be trackers and have access to global position information in addition to the odometry and orientation information that all vehicles have access to. All vehicles can observed the others and we ignore the landmark observations in the datasets. We subsampled the ground truth position and orientation, odometry and intervehicle measurements to better approximate low frequency sensors in field usage. We ran the same set of optimisers across the data as in the simulated environment but refrain from grouping results as the paths and durations differ significantly between each of the 9 samples. The other significant difference is the variability in intervehicle measurements. We add an artificial restriction to create a minimum interval of 6 seconds, however due to orientations of vehicles and limitations of the visual system used to calculate range and bearing between targets this interval can, and is often exceeded.

Figure 3.16 shows an example from dataset 9 that illustrates these delays. This dataset

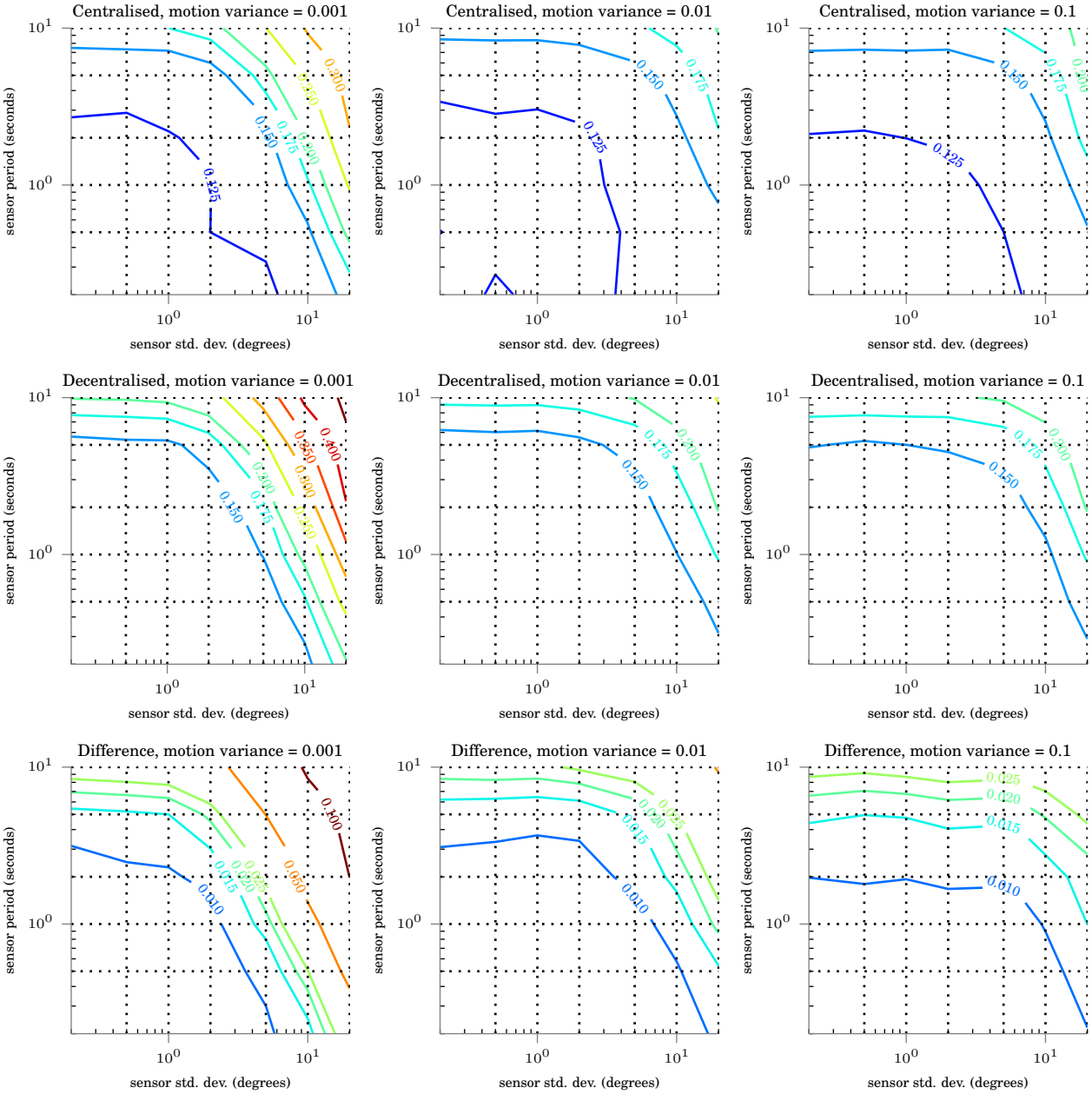
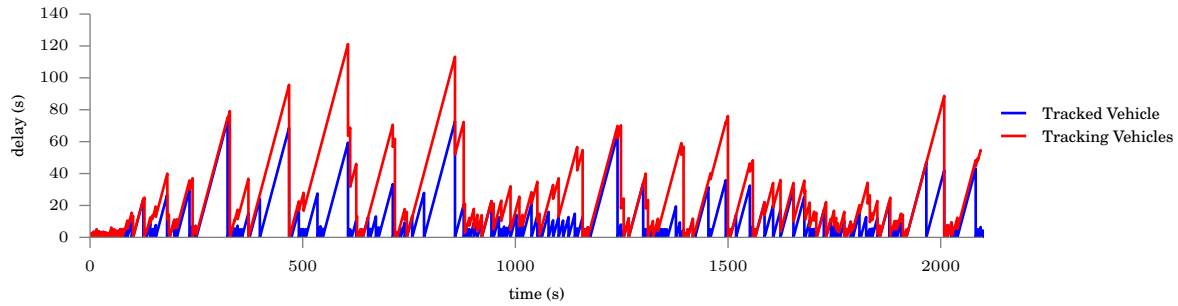
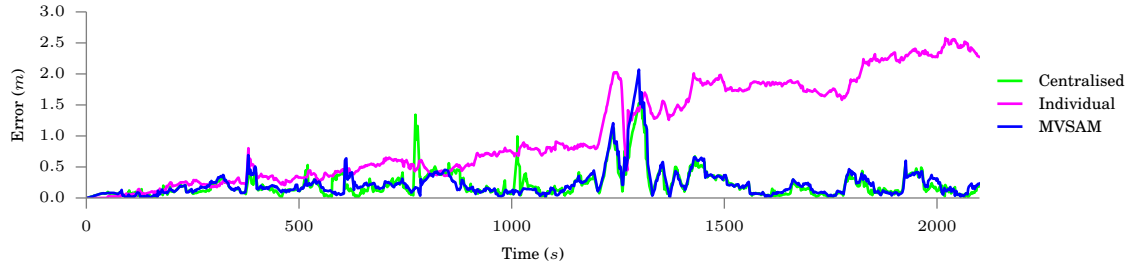


Figure 3.15 – Comparing the RMS Position Error at the end of the simulation for a variety of orientation sensor periods, variances and motion model variances. The error is calculated at each time step for the current state for the last 10 seconds of the simulation (this is done in the event that the position is growing without bound, such as in a dead reckoning case). The RMS error is then calculated across all states in all samples for a given combination of parameters. Again, like the heading error the position error is larger using the partially fixed linearisation. However the difference between the two solutions is still relatively small compared to the error relative to ground truth.



(a) Delay to incorporation of tracking data from either vehicle 1 or 3 on vehicle 2, including indirect observations through vehicles 4 and 5. This includes indirect observations through vehicles 4 and 5.



(b) RMS position error for vehicle 2 from each optimiser. Little difference between the decentralised and centralised optimisers is visible.

Figure 3.16 – Tracked vehicle from MRCLAM dataset 9. Growth in error is apparent when no tracking information is arriving at the vehicle, and is even more pronounced where no external information is arriving at the vehicle.

was chosen as it has the largest RMS position errors and longer delays between tracking updates and better illustrates performance under these adverse conditions. Figure 3.16a shows the time since last incorporation of data from a tracking vehicle (vehicles 1 or 3) or from another tracked vehicle (vehicles 4 or 5) on vehicle 2. Figure 3.16b shows the position error for the same vehicle. The cooperative methods outperform the isolated method as expected, and do not differ significantly from each other, with only two visible portions at around 600s and 800s in this example. Error grows as the intervals increase, reducing again upon subsequent observations - with multiple observations resulting in faster reductions.

Transmitted data is calculated for transmission of all measurements and their timestamps - which is required for calculating the centralised method as well as for transmitting only packets compressed using our method. One item we noted was that there is a large penalty for 'compressing' if only a small number of measurements were taken during this time period. To this end we also include a column with the data transmitted using the minimum

Dataset	Raw	All Compressed	Minimal	Saving
Dataset 1	8213 kiB	4389 kiB	3665 kiB	55.38%
Dataset 2	10087 kiB	5872 kiB	4735 kiB	53.06%
Dataset 3	8353 kiB	4112 kiB	3479 kiB	58.35%
Dataset 4	7688 kiB	4308 kiB	3566 kiB	53.61%
Dataset 5	12850 kiB	7507 kiB	6095 kiB	52.56%
Dataset 6	4032 kiB	2479 kiB	2000 kiB	50.38%
Dataset 7	4710 kiB	2804 kiB	2293 kiB	51.30%
Dataset 8	22642 kiB	12397 kiB	10237 kiB	54.78%
Dataset 9	10609 kiB	4707 kiB	3916 kiB	63.08%
Totals	89188 kiB	48580 kiB	39992 kiB	55.16%

Table 3.1 – Total data transmitted in bytes for each UTIAS dataset in a tracking scenario. Raw denotes timestamps and actual observation values, compressed denotes transmitting the partially eliminated Jacobian, linearisation values and error vector whilst minimal uses the smaller of each for any given packet to be transmitted. Saving is the percent bandwidth saved by using the minimal transmission method.

of either actual measurement values and their timestamps or the compression method and the percent bandwidth saved with the minimal case.

The compressed method for all packets saves approximately 45.5% of bandwidth, although usage of the smaller of raw or compressed on any given packet nets a total saving of 55.2%. Reasons for this can be when intervehicle observations occur very closely so only a small number of observations are contained in the packet. Furthermore when greater periods elapse between intervehicle observations savings from usage of the compressed method become more significant - as was shown in Figure 3.7.

Overall the tracking scenario for cooperative localisation does not show significant difference between error from the centralised or decentralised methods, whilst maintaining less communications and computation cost (due to limited relinearisation).

3.2.3 GPS Denied Environments

An alternate scenario for multiple robots is one where none can obtain absolute position measurements, meaning no extra information about the position states are obtained after initialisation. Excluding scenarios with landmarks or loop closures the uncertainty grows over time, limited by the motion model variance and rate/accuracy of intervehicle measurements and number of vehicles. This can be seen as a group of indoor robots or a fleet

of autonomous submersibles moving in an environment without global constraints through new terrain.

3.2.3.1 Simulated Data

We again use Monte Carlo simulation to show the effects of linearisation error and heading uncertainty. The parameters for this scenario match that of the tracking scenario with the exclusion of global position information from the tracking vehicles. With no global position information entering the system after initialisation the RMS position error is expected to slowly rise in the same manner as the individual solutions of the tracked vehicles from the previous scenario albeit at a reduced rate. Simulation of a single variable system with multiple vehicles gives the limit for variance reduction for a multiple vehicle system as $\frac{1}{n}$ of the single vehicle case where n is the number of vehicles involved. The limit is when there is no uncertainty in intervehicle observations. In this case the covariance reduction for a single observation applies equally to all vehicles. The reduction of variance in a n vehicle system for a single vehicle with no intervehicle observation uncertainty is the same as a single vehicle system with n observations at each interval. This then gives the lower bound of $\frac{1}{n}$. Increased sensor noise, intervals and sparser intervehicle observations increase the variance above this limit, although it will not exceed the single vehicle variance.

We show low and high noise sample solutions in Figure 3.18 and the difference between centralised solutions and our method across the range of orientation parameters in Figure 3.19. The lack of constraint in position after initialisation makes calculation of a metric/contour plot of final RMS more challenging as the variance is not approximately constant but increasing. We find with the circular paths, and even with the more complex paths in the captured data that the errors can by chance bring the vehicle back closer to the true position before again drifting. Due to this the summarised RMS position error plots are not as illuminating in this example.

The poor linearisation at the largest sensor variance and period and smallest vehicle variance we simulate resulted in approximately 20% of the samples failing to solve due to numerical stability problems for our decentralised method. The centralised (and individual) method still yielded a solution for all samples. Failure of the decentralised method is related to a combination of high error and high confidence due to expected minimal motion variance

and the inability to relinearise a poor initial linearisation for remote values when fusing intervehicle measurements. Additionally the relative information for low certainty orientation sensors and high certainty motion model could lead to incorrect estimates of rank due to significant difference in magnitude of information.

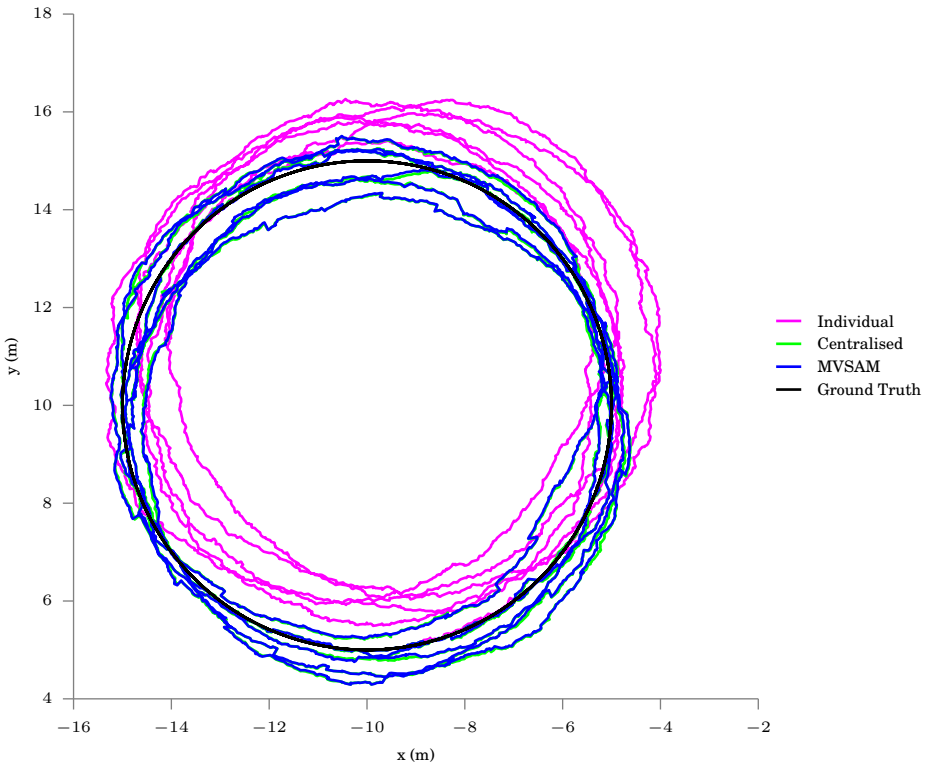
3.2.3.2 Captured Data

Again we utilise the UTIAS datasets for which complete graphs can be seen in the appendices. Similar to the tracking scenario the decentralised and centralised solutions provide very similar solutions with any difference being substantially smaller than the difference to the isolated solutions. There are more occasions of small difference, and the centralised estimator tends to oscillate between two similar solutions slightly more than the decentralised solution which can be attributed to the ability to relinearise all variables, making exploration of nearby locally minimal solutions more likely (see datasets 8 and 9 in the appendices). The inability of the decentralised method to relinearise tends to reduce fluctuations as a large portion of states cannot be adjusted and effectively anchor the elements that can be updated.

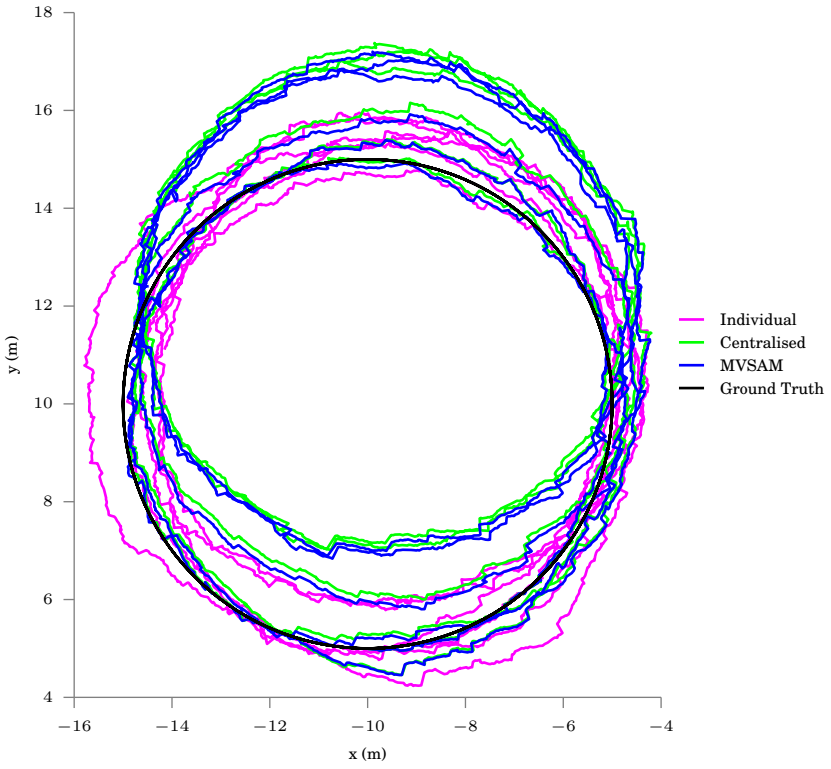
Finally we calculate the data required for transmission of measurements and timestamps compared to transmission of our combined observations. We also include an optimistic version which inspects each packet and uses the smaller of the two - enabling smaller packets when there are a very small number of local measurements between adjacent intervehicle measurements. The compression method uses less bandwidth than transmitting raw measurements, at approximately 43.5% reduction but using the lesser of raw or compressed for any given packet enables total savings of about 53.8%.

3.3 Summary

The relationship between the difference between centralised and decentralised solutions does not exhibit a linear relationship to the ratio of sensor period and sensor standard deviation. Instead regions of minimal difference between centralised and decentralised solution errors can be defined as periods less than 5 seconds and standard deviation of less than 5 degrees. Furthermore sensor noise can be increased as long as the sensor period is reduced. These

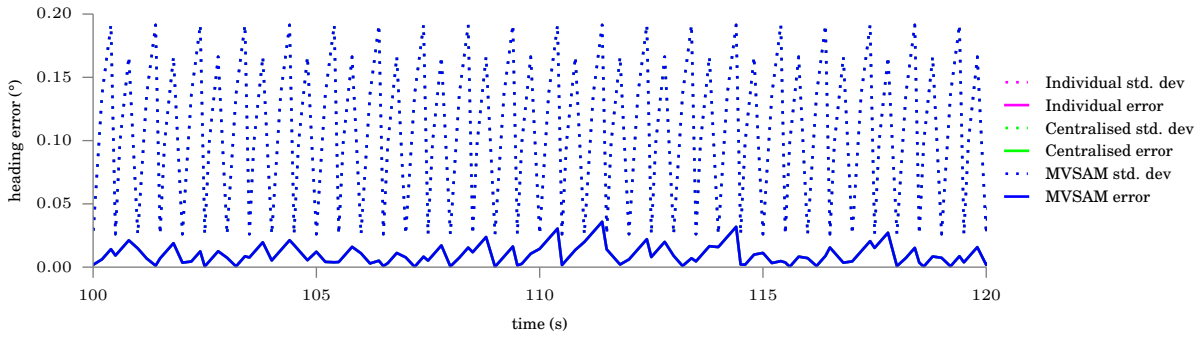


(a) Online path estimation for low noise.

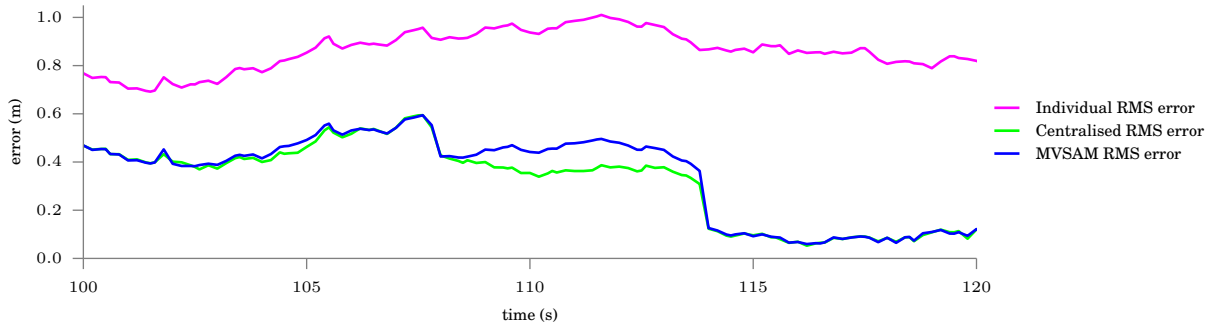


(b) Online path estimation for high noise.

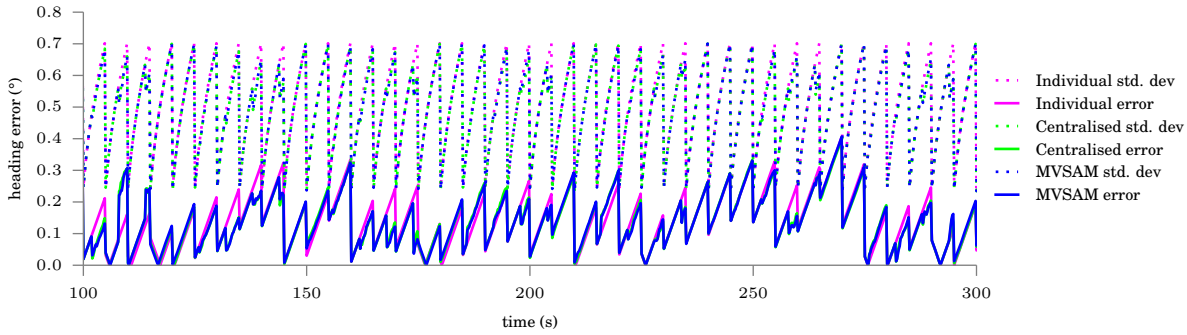
Figure 3.17 – Samples of simulated data for one of the vehicles in a low (0.5s period, 0.5° standard deviation) and high noise (5s period, 5° standard deviation) environments. Plots of position and heading RMS over time are in Figure 3.18.



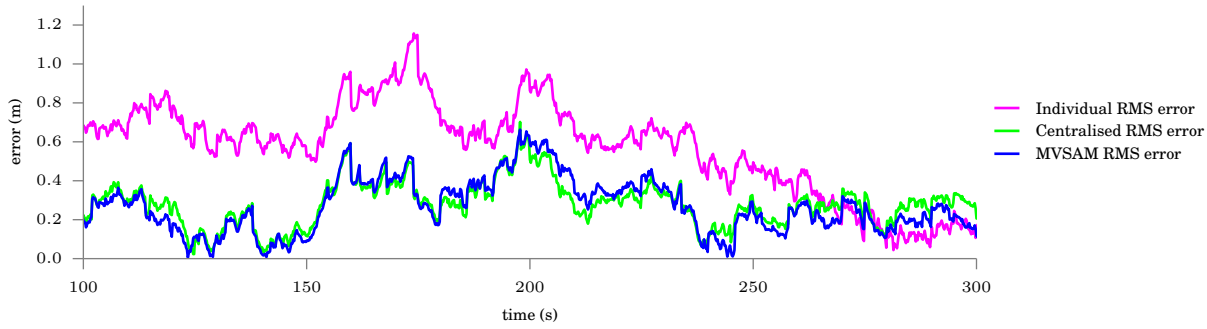
(a) Segment of low noise sensor absolute heading estimation error and 3σ bound. Lines for each solution overlap and obscure each other in this graph.



(b) Segment of low noise sensor vehicles position RMS error.



(c) Segment of high noise sensor absolute heading estimation error and 3σ bound.



(d) Small segment of high noise sensor vehicles position RMS error.

Figure 3.18 – Samples of simulated data for one of the vehicles in a low and high noise environments. These RMS error plots correspond to the paths in Figure 3.17. What is barely appreciable from these graphs is the difference between the centralised and summarisation methods - with the expected differences being of the order of 0.01m and 0.2°.

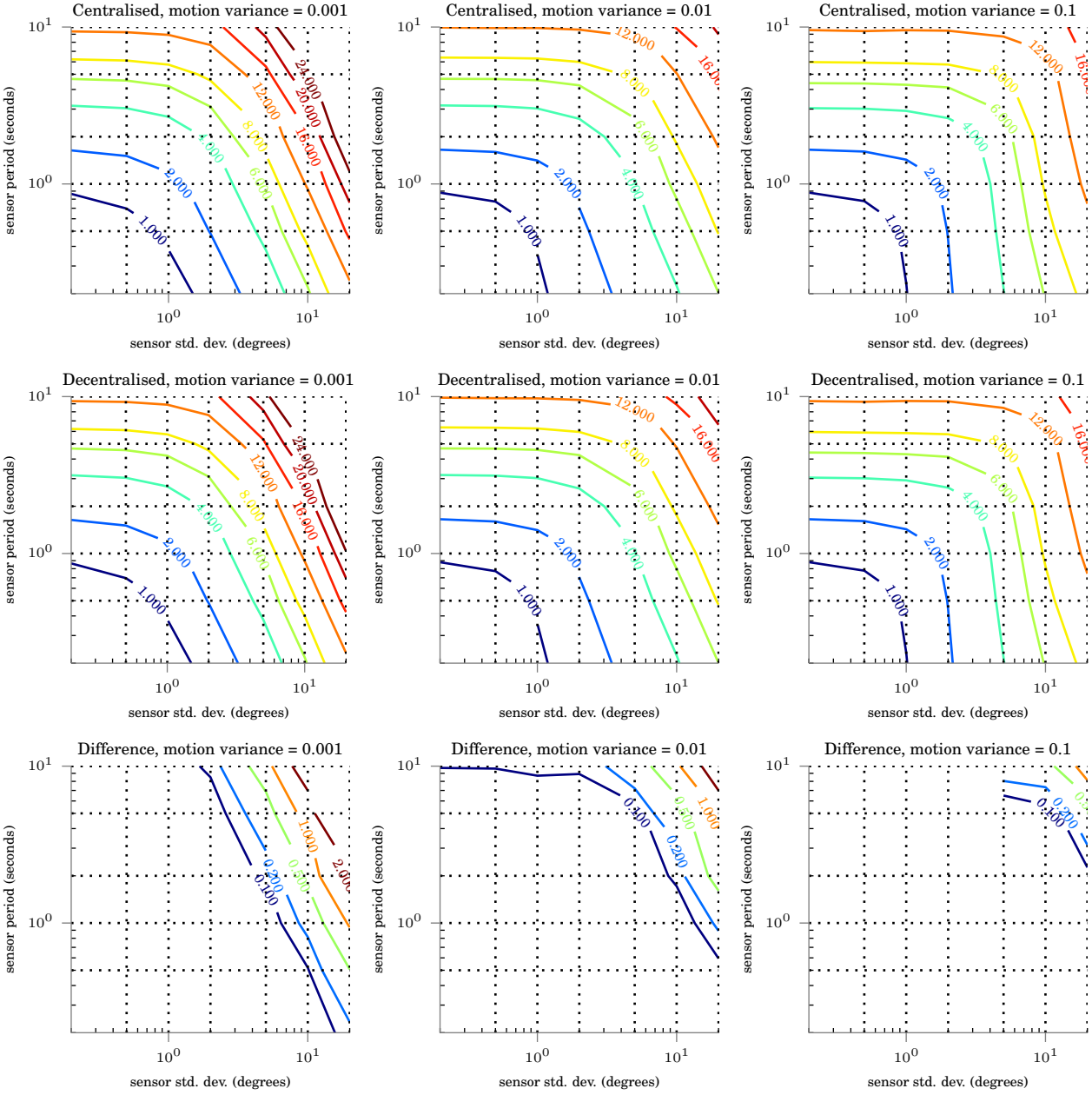
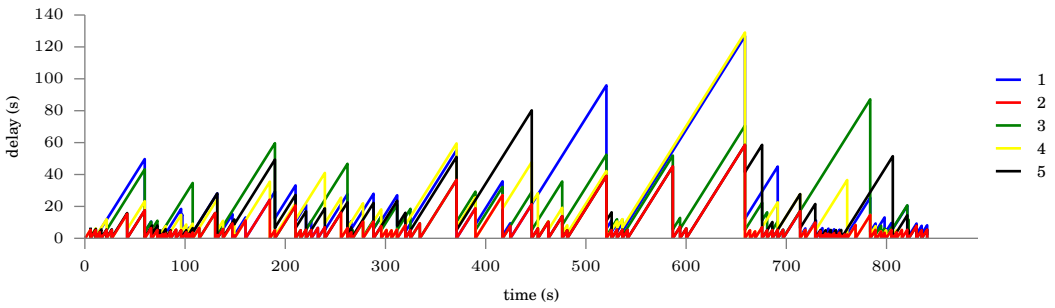


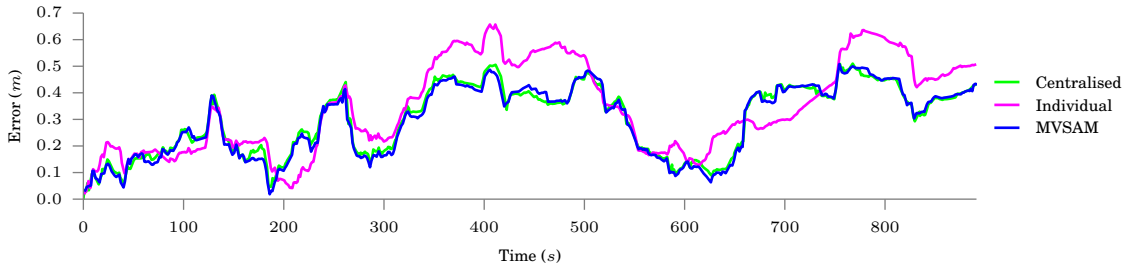
Figure 3.19 – RMS heading error at the end of GPS denied simulation. The difference between the centralised relinearising solution and our method is relatively small when compared to ground truth error.

Dataset	Raw	All Compressed	Minimal	Saving
Dataset 1	7924 kiB	4389 kiB	3643 kiB	54.02%
Dataset 2	9730 kiB	5872 kiB	4700 kiB	51.69%
Dataset 3	8071 kiB	4112 kiB	3462 kiB	57.10%
Dataset 4	7416 kiB	4308 kiB	3541 kiB	52.25%
Dataset 5	12398 kiB	7507 kiB	6045 kiB	51.24%
Dataset 6	3884 kiB	2479 kiB	1982 kiB	48.97%
Dataset 7	4536 kiB	2804 kiB	2278 kiB	49.78%
Dataset 8	21831 kiB	12397 kiB	10168 kiB	53.42%
Dataset 9	10197 kiB	4707 kiB	3891 kiB	61.84%
Totals	85991 kiB	48580 kiB	39714 kiB	53.82%

Table 3.2 – Total data transmitted in bytes for each UTIAS dataset without GPS information. Raw denotes timestamps and actual observation values, compressed denotes transmitting the partially eliminated Jacobian, linearisation values and error vector whilst minimal uses the smaller of each for any given packet to be transmitted. Saving is the percent bandwidth saved by using the minimal transmission method.



(a) Delay to incorporation of data from other vehicles on vehicle 2.



(b) RMS position error for vehicle 2 from each optimiser. Differences between centralised and decentralised are more apparent than in the tracking case.

Figure 3.20 – Vehicle 2 from MRCLAM dataset 7, showing delay to fusion. Jumps in optimiser estimates are apparent when fusing after long gaps in communication as evident at 260s, 440s, 520s and 650s. Of more interest is the reduced error which the centralised estimator has at times 50s and 630s. The ability of the centralised solver to relinearise all factors permits better/more accurate exploration of nearby local minima than the fixed linearisation does.

approximate boundaries represent less than 0.1 metre difference in RMS position error. We believe this rough boundary is higher noise than many systems would have. As noise in states involved in nonlinear factors increases, the difference grows, although does not show significant difference until it has exceeded noise expected in real world scenarios.

Demonstration on captured data shows little difference between the centralised and decentralised method, whilst reducing the bandwidth requirements approximately 32% at minimum although on average nearer to 37.4% using our compression method for large packets when compared to complete measurement transmission.

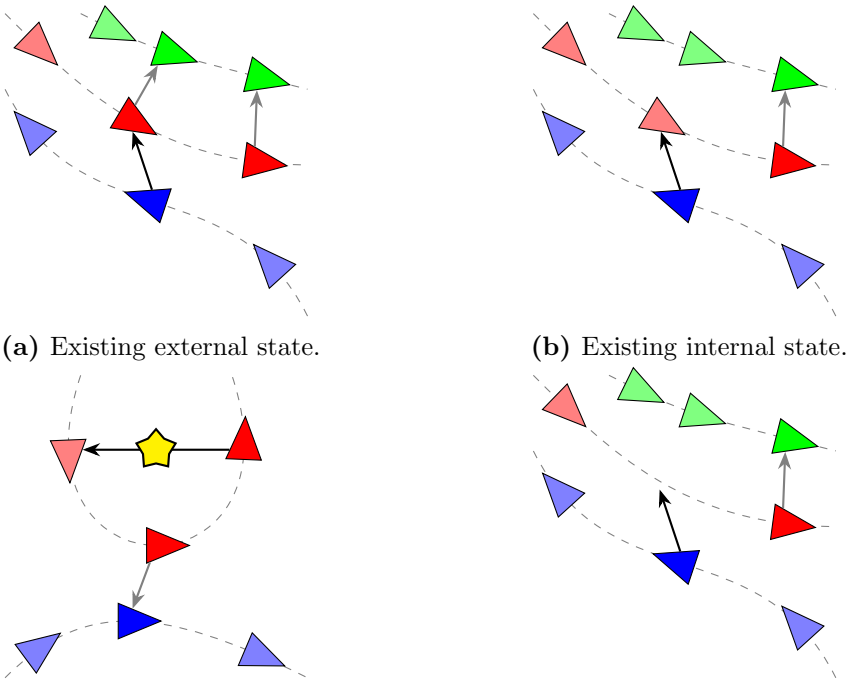
Chapter 4

Out of Sequence Measurements

Whilst methods for combining data can save bandwidth it can introduce issues when there are delays in communication. Existing methods involving compressed, fused or condensed measurements assist in reducing bandwidth but cannot handle delayed observations without a fixed delay horizon beyond which observations are ignored [8]. Alternate methods exist that target combined sensor and communication mediums so the observation does not occur if the communication fails [23]. We make three contributions in this chapter. The first two relate to methods for handling out of sequence observations and the third is investigation and derivation of models that permit usage of the second method where we insert new historical states.

This chapter contributes two methods that handle related variants of the out of sequence problem for intervehicle measurements. The first method handles the case where an internal state at the time of the out of sequence observation already exists and the second method handles the case where no state exists. Neither method requires retransmission or replacement of already transmitted data. Both methods result in conditional probabilities that can be shared and fused similarly to packets in the last chapter and receiving systems cannot differentiate packets from either of the new methods. Where measurements arrive in sequence the subgraph elimination from the previous chapter is used.

This chapter also investigates vehicle model behaviour and the effect of insertion of extra states in the odometry chain of a vehicle and contributes non-trivial vehicle models that, for usage in a nonlinear estimator, do not modify the results for a single iteration.



(a) Existing external state.

(b) Existing internal state.

(c) Existing internal with a loop closure.

(d) Missing state.

Figure 4.1 – Four examples of the three cases that can arise with out of sequence observations. The trivial case with an existing external state is depicted in (a). We show two examples for when an internal state exists; one for an intervehicle observation (b) and the other using a featureless loop closure (c). The cast of a missing state is shown in (d). Each case shows the movement of vehicles along a path with faded states indicating an internal state when the observation was received and the more saturated states indicating an external state. In each case the out of sequence observation is indicated with a black arrow with existing intervehicle observations indicated with a gray arrow.

We identify three distinct cases, depicted in Figure 4.1, to handle when measurements arrive out of sequence:

1. An external/shared state already exists as in Figure 4.1a. This requires no special handling as the desired information is already shared.

2. An internal/unshared state exists as in Figure 4.1b. The most apparent application of this is depicted in Figure 4.1c where an initial observation is made and later associated with another observation to create a landmarkless loop closure. Our first contribution handles this case through exploiting properties of elimination reordering to create a factor dependent on the internal state and the two external states chronologically either side of it. This does not require retransmission of information in previous packets that contain information about these two neighbouring external states.

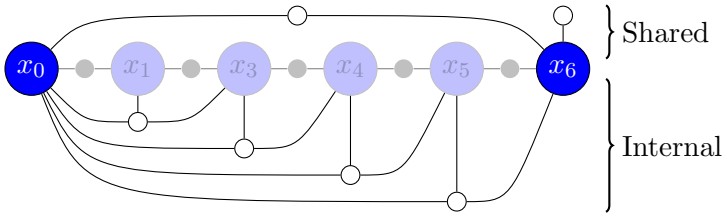
3. No state exists at the target time. This requires insertion of a new state. This is the most likely case where intervehicle observations occur with delay. Our second and third contribution in this chapter refers to this case. The first of these contributions is derivation of a set of constraints that the vehicle model must obey to insert an intermediate state without affecting existing factorised solutions. Once this new state is inserted as the first eliminated state in a subgraph this case can be treated as per the previous case. The second of these contributions is derivation of two vehicle models that can be used in real world problems and obey the derived constraints.

4.1 Elimination Reordering

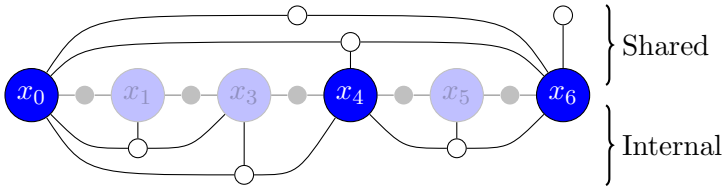
The simplest case for handling an out of sequence intervehicle measurement factor is when a variable representing the target state already exists. This may occur frequently with high internal measurement rates or rarely with lower internal measurement rates. Depending on intervehicle sensing mechanisms an accurate clock could be used to synchronise observations across all platforms. Even if no local observation of remote vehicles is made a local state can be added in case a remote vehicle observes the local vehicle. Alternatively methods that used featureless loop closures can add local states upon every image or laser scan even if not initially informative.

Given that an internal variable already exists the elimination ordering can be adjusted to include this variable with the external variables in the subgraph. Naively doing this requires performing the entire elimination of the subgraph and retransmission of the root of the Bayes Tree referring to the external variables. Instead we want to do this to minimise recomputation and prevent the need to retransmit the bottom of the Bayes Tree relating to already existing external variables. We depict a subgraph over six variables and the process of 'promoting' an existing variable in Figures 4.2, 4.3 and 4.4 in factor graph, matrix and Bayes tree form.

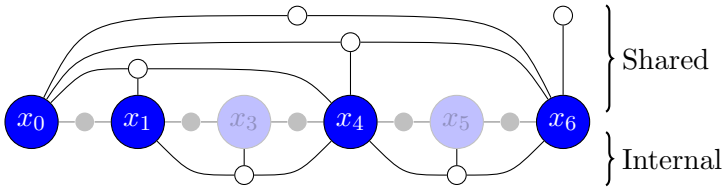
Instead of starting with the initial Jacobian we begin with the eliminated form upon creation of the initial packet using the method developed in the previous chapter. Mathematically this is the upper triangular solution matrix calculated by variable elimination and is shown in Figure 4.3a.



(a) After compression, before state promotion.

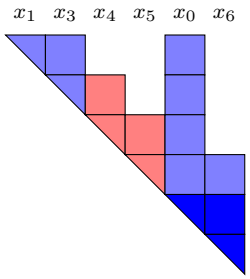


(b) After promotion of a single state.

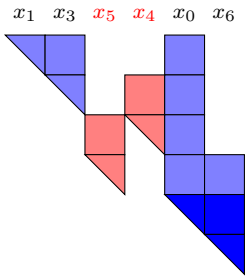


(c) After promotion of a second state.

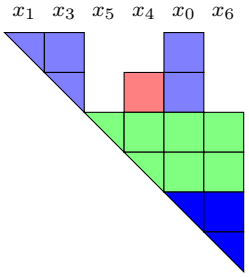
Figure 4.2 – Factor graphs showing the original graph faded out with the calculated factors produced by elimination in white. External states are shown in bright blue and internal states are faded. Factors above the faded factors are the factors that are shared between vehicles whilst those below the line are held internally. Each promoted state rearranges the elimination order and requires sharing of a single extra factor linking three states - the newly promoted state and the two states chronologically either side of it. Here we omit x_2 as no observations took place at that time step.



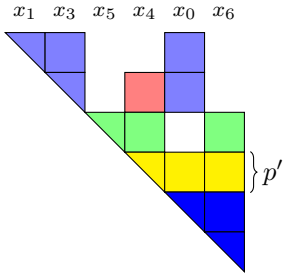
(a) Before reordering. Red blocks indicate the variables being reordered.



(b) The now reordered variables are shown in red along with their reordered blocks.

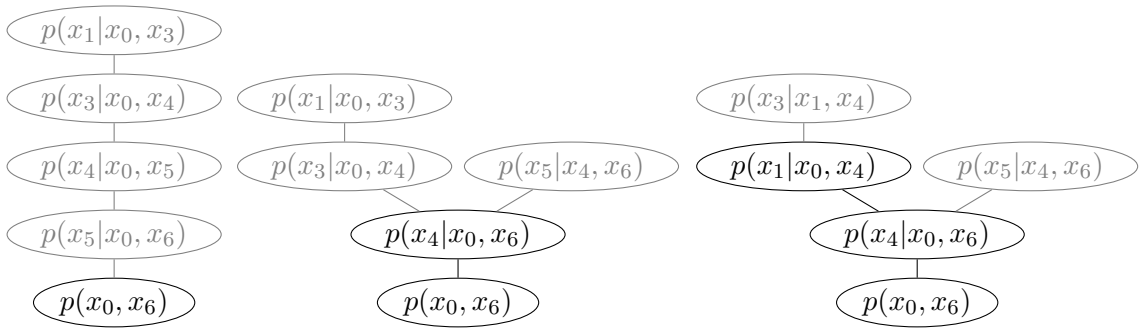


(c) Eliminate/factorisation update. Blocks touched by elimination are in green



(d) Extract conditional probability - which is the new factor we are sharing and is shown in yellow.

Figure 4.3 – The process of reordering the variables in a packet to promote an existing internal variable x_4. This is the matrix view of the changes in Figure 4.2 to produce the first state promotion. We highlight successive changes through variable reordering and decomposition before extraction of the additional conditional row to transmit which is depicted in yellow in the final matrix. Particular note is drawn to the blocks that are zeroed in (d); using the Bayes Tree structure shows that these two blocks are sparse where a linear algebra approach does not without calculation and sufficient numerical accuracy. Additionally the conditional rows for x_3 and x_5 could also be extracted with ease as the conditionals only depend on states which have already been shared.



(a) Original. **(b)** Bayes tree with x_4 promoted. **(c)** Bayes tree with x_1 promoted.

Figure 4.4 – Bayes trees for the initial graph and after promoting internal variables. The shared nodes are shown in black whilst nodes that are not transmitted are grayed out. This structure can be used to identify which nodes will be affected by promotion of a new external varible.

In this case x_4 is promoted to become an external variable. Columns are reordered to match the new ordering which places x_4, our variable of interest, to be eliminated after all internal variables and before all existing external variables (x_0 and x_6) in the subgraph as in Figure 4.3b. This breaks the triangular structure of the solution matrix and requires updating of the factorisation. This is achieved using an iterative QR decomposition algorithm, such as Givens Rotations or Householder Reflections. We track changed blocks in the example subgraph and show the modified blocks in green in Figure 4.3c. Certain nodes in the Bayes Tree are left unmodified and these nodes exist at both the leaves (as might be expected from initial work on the Bayes Tree and relinearisation [38]) but also, and more importantly for our case, the root of the tree that has already been transmitted.

The affected rows correspond to variables that have other variables swap between 'eliminated before' and 'eliminated after'. Thus in our example x_1, x_3, x_0 and x_6 are unaffected as they remain in the first two and last two places in the ordering and so no variables swap from their before and after sets. Given this, the previously shared rows relating to existing external states x_0 and x_6 are unchanged and do not require retransmission. This extends to the promotion of further variables within the subgraph as long as each successive promotion places the new external variable before existing external variables and after the internal variables. The Bayes Tree structure assists in evaluation of which variables require recalculation for the reordering and through caching previous elimination results at the leaves of the tree can reduce required recomputation. The use of this is apparent with the

second promotion we depict in Figure 4.2c.

The promotion of x_1 modifies the sets of variables either side of x_5, but does not affect the factor/conditional relating states x_5, x_4, x_6 which the linear algebra approach does not predict. The Bayes tree illustrates this through the update process where the node containing the new external variable as a frontal variable and all nodes down to the shared portion of the Bayes Tree are removed. The conditional probabilities are reinterpreted as linearised factors and elimination is performed over these variables before returning to the Bayes Tree. The leaves and shared root can then be reattached and are unaffected.

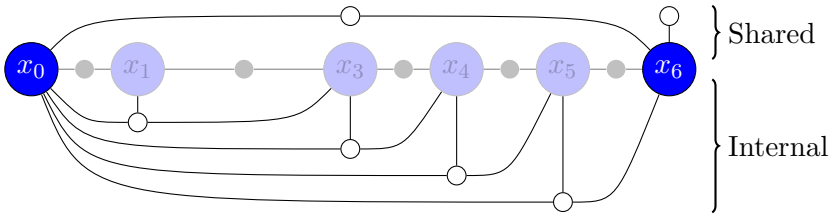
Finally the packet that relates the new external variables to the existing external variables can be extracted as depicted in Figure 4.3d and shared with remote vehicles.

We have reduced the elimination to an iterative update of the Jacobian, minimising the region that needs to be recomputed. We have also shown that the already transmitted portion of the subgraph is unaffected by changing elimination order, as long as the external variables remain in the same order. The final part of the algorithm is to extract the conditional row that is required to use the new external variable in the global graph. This is the row shown in yellow in Figure 4.3d. This figure also shows the actual block structure of the solution matrix with two elements being zero that are not apparent in a pure linear algebra derivation. The sparsity of these elements is apparent when looking at the Bayes Tree in Figure 4.4b.

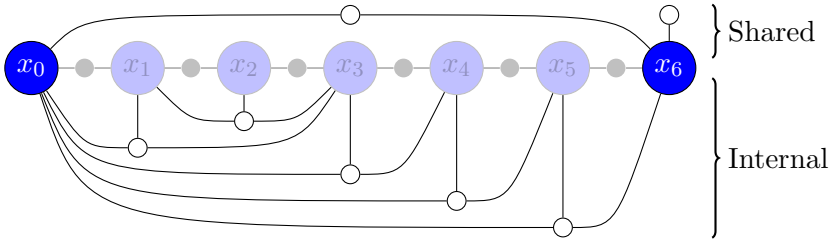
We briefly mentioned the case where additional out of sequence observations arrive that refer to a single subgraph. Our algorithm readily extends to this case and the size of the communicated information is constant. The promotion of x_1 in our examples involves transmission of the conditional probability of x_1 upon its chronologically adjacent external states x_0 and x_4. This occurs as x_1 and x_6 are conditionally independent given x_4. In this way a newly promoted variable will always be conditioned upon exactly two other variables giving the constant size update we desire.

4.2 State Insertion

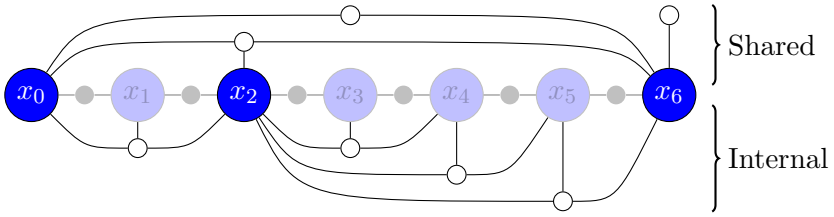
The case where a state does not already exist presents issues with connecting the new variable into the existing graph without invalidating the previously transmitted components



(a) After compression, before state insertion.



(b) After insertion of x_2.



(c) After promotion of x_2.

Figure 4.5 – Factor graphs showing the process of inserting a new state in a subgraph. The original graph is shown faded out with the calculated factors produced by elimination in white. We insert the new factor and calculate the composite factor, or conditional of the new state x_2 upon the adjacent states x_1 and x_3. This behaves as though x_2 was eliminated before all other variables in the subgraph and we use the elimination reordering of the previous section to promote this variable.

from the subgraph and requiring retransmission. The two new temporal factors and linearisation point for the new state must be equivalent after elimination to the existing temporal factor which they replace between the two adjacent states in the subgraph. This allows us to calculate conditions that the temporal error metric and choice of linearisation point must fulfil to prevent invalidation of the previous segment. The aim for state insertion is to create the state as if it were eliminated first in the original elimination order then use the promotion method of the previous section to bring it from first eliminated of the internal states to first eliminated of the external states.

Given an error metric which fulfills the conditions we derive in Section 4.2.1 we can recalculate the subgraph and can guarantee that addition of the state does not affect the joint

probability at the root of the Bayes Tree. Calculation of this update to create the Jacobian can be done iteratively by creating a two step system and eliminating the intermediate state first using QR elimination. Calculation of the marginal information matrix and comparison with the information matrix from the original factor can be used to verify that this method is indeed correct and does not invalidate existing calculations. The conditional probability derived from this can be added to the solution matrix and state promotion can occur as for the existing states. Alternatively the entire subgraph can be recalculated with the new elimination order.

Complete recalculation is simpler to implement and utilises less memory but requires greater computation effort. Alternatively the small two step system can be created and the intermediate variable eliminated. The probability of the intermediate state conditioned on its adjacent states can then be attached to the Bayes Tree for the segment. The elimination order can then be adjusted to promote the inserted state to the externals and the resulting conditional probability is transmitted to other vehicles.

4.2.1 Derivation of Conditions for Applicability

To calculate these conditions we look at the factor graph for a simple two state optimisation problem with time step $(\alpha + \beta)$, where $\alpha \geq 0$ and $\beta \geq 0$. The linearised version of the problem is calculated symbolically and includes the Jacobian matrices, error metric at the linearisation points and noise models. We also calculate the same for a three state system with time steps α and β and an as yet unknown value for the linearisation point of the intermediate state. To be able to find the conditions, we need to eliminate the intermediate state from the three state system. This is best achieved analytically through calculation of the Information matrix and marginalisation rather than calculation of the square root decomposition which involves factorisation steps that are difficult to calculate analytically. Calculation of the Information matrix and the Information vector is done by premultiplying the least squares problem with the transpose of the Jacobian (and if not already incorporated in the Jacobian matrix the inverse of the measurement Covariance matrix Λ):

$$J\Delta\Theta = -g\left(\Theta^0\right) \tag{4.1}$$

$$J^T\Lambda^{-1}J\Delta\Theta = -J^T\Lambda^{-1}g\left(\Theta^0\right) \tag{4.2}$$

$$\mathcal{I}\Delta\Theta = \mathbf{i} \tag{4.3}$$

The matrix and vector for the one step problem with start and end states θ_0 and θ_2 respectively:

$$\mathcal{I} = \begin{bmatrix} \frac{\partial f_{\alpha+\beta}}{\partial \theta_0}^T \\ \frac{\partial f_{\alpha+\beta}}{\partial \theta_2}^T \end{bmatrix} \begin{bmatrix} \Lambda_{\alpha+\beta}^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial f_{\alpha+\beta}}{\partial \theta_0} & \frac{\partial f_{\alpha+\beta}}{\partial \theta_2} \end{bmatrix} \tag{4.4}$$

$$= \begin{bmatrix} \frac{\partial f_{\alpha+\beta}}{\partial \theta_0}^T \Lambda_{\alpha+\beta}^{-1} \frac{\partial f_{\alpha+\beta}}{\partial \theta_0} & \frac{\partial f_{\alpha+\beta}}{\partial \theta_0}^T \Lambda_{\alpha+\beta}^{-1} \frac{\partial f_{\alpha+\beta}}{\partial \theta_2} \\ \frac{\partial f_{\alpha+\beta}}{\partial \theta_2}^T \Lambda_{\alpha+\beta}^{-1} \frac{\partial f_{\alpha+\beta}}{\partial \theta_0} & \frac{\partial f_{\alpha+\beta}}{\partial \theta_2}^T \Lambda_{\alpha+\beta}^{-1} \frac{\partial f_{\alpha+\beta}}{\partial \theta_2} \end{bmatrix} \tag{4.5}$$

$$\mathbf{i} = - \begin{bmatrix} \frac{\partial f_{\alpha+\beta}}{\partial \theta_0}^T \\ \frac{\partial f_{\alpha+\beta}}{\partial \theta_2}^T \end{bmatrix} \begin{bmatrix} \Lambda_{\alpha+\beta}^{-1} \end{bmatrix} \begin{bmatrix} f_{\alpha+\beta} \end{bmatrix} \tag{4.6}$$

$$= \begin{bmatrix} -\frac{\partial f_{\alpha+\beta}}{\partial \theta_0}^T \Lambda_{\alpha+\beta}^{-1} f_{\alpha+\beta} \\ -\frac{\partial f_{\alpha+\beta}}{\partial \theta_2}^T \Lambda_{\alpha+\beta}^{-1} f_{\alpha+\beta} \end{bmatrix} \tag{4.7}$$

and the two step problem, placing the intermediate state θ_1 first in the column order, is then calculated as:

$$\mathcal{I} = \begin{bmatrix} \frac{\partial f_\alpha}{\partial \theta_1}^T & \frac{\partial f_\beta}{\partial \theta_1}^T \\ \frac{\partial f_\alpha}{\partial \theta_0}^T & \\ & \frac{\partial f_\beta}{\partial \theta_2}^T \end{bmatrix} \begin{bmatrix} \Lambda_\alpha^{-1} & \\ & \Lambda_\beta^{-1} \end{bmatrix} \begin{bmatrix} \frac{\partial f_\alpha}{\partial \theta_1} & \frac{\partial f_\alpha}{\partial \theta_0} & \\ \frac{\partial f_\beta}{\partial \theta_1} & & \frac{\partial f_\beta}{\partial \theta_2} \end{bmatrix} \tag{4.8}$$

$$= \begin{bmatrix} \frac{\partial f_\alpha}{\partial \theta_1}^T \Lambda_\alpha^{-1} \frac{\partial f_\alpha}{\partial \theta_1} + \frac{\partial f_\beta}{\partial \theta_1}^T \Lambda_\beta^{-1} \frac{\partial f_\beta}{\partial \theta_1} & \frac{\partial f_\alpha}{\partial \theta_1}^T \Lambda_\alpha^{-1} \frac{\partial f_\alpha}{\partial \theta_0} & \frac{\partial f_\beta}{\partial \theta_1}^T \Lambda_\beta^{-1} \frac{\partial f_\beta}{\partial \theta_2} \\ \frac{\partial f_\alpha}{\partial \theta_0}^T \Lambda_\alpha^{-1} \frac{\partial f_\alpha}{\partial \theta_1} & \frac{\partial f_\alpha}{\partial \theta_0}^T \Lambda_\alpha^{-1} \frac{\partial f_\alpha}{\partial \theta_0} & \\ \frac{\partial f_\beta}{\partial \theta_2}^T \Lambda_\beta^{-1} \frac{\partial f_\beta}{\partial \theta_1} & & \frac{\partial f_\beta}{\partial \theta_2}^T \Lambda_\beta^{-1} \frac{\partial f_\beta}{\partial \theta_2} \end{bmatrix} \tag{4.9}$$

$$\mathbf{i} = - \begin{bmatrix} \frac{\partial f_\alpha}{\partial \theta_1}^T & \frac{\partial f_\beta}{\partial \theta_1}^T \\ \frac{\partial f_\alpha}{\partial \theta_0}^T & \\ & \frac{\partial f_\beta}{\partial \theta_2}^T \end{bmatrix} \begin{bmatrix} \Lambda_\alpha^{-1} & \\ & \Lambda_\beta^{-1} \end{bmatrix} \begin{bmatrix} f_\alpha \\ f_\beta \end{bmatrix} \tag{4.10}$$

$$= \begin{bmatrix} -\frac{\partial f_\alpha}{\partial \theta_1}^T \Lambda_\alpha^{-1} f_\alpha - \frac{\partial f_\beta}{\partial \theta_1}^T \Lambda_\beta^{-1} f_\beta \\ -\frac{\partial f_\alpha}{\partial \theta_0}^T \Lambda_\alpha^{-1} f_\alpha \\ -\frac{\partial f_\beta}{\partial \theta_2}^T \Lambda_\beta^{-1} f_\beta \end{bmatrix} \tag{4.11}$$

After which the intermediate state needs to be eliminated to enable comparison with the one step problem. Instead of using the elimination algorithms previously defined we change to treat the Information matrix and vector as a multivariate Gaussian and calculate the marginal distribution over the two end states. This effectively eliminates the intermediate state but does not require decompositions of the block symbols we have been using and thus facilitates simpler calculation and comparison of the one and two step cases.

The marginal information matrix and vector are:

$$\mathcal{I} = \begin{bmatrix} \frac{\partial f_\alpha}{\partial \theta_0}^T \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \left(\Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \Lambda_\alpha \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \right)^{-1} \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \frac{\partial f_\alpha}{\partial \theta_0} & -\frac{\partial f_\alpha}{\partial \theta_0}^T \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \left(\Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \Lambda_\alpha \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \right)^{-1} \frac{\partial f_\beta}{\partial \theta_2} \\ -\frac{\partial f_\beta}{\partial \theta_2}^T \left(\Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \Lambda_\alpha \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \right)^{-1} \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \frac{\partial f_\alpha}{\partial \theta_0} & \frac{\partial f_\beta}{\partial \theta_2}^T \left(\Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \Lambda_\alpha \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \right)^{-1} \frac{\partial f_\beta}{\partial \theta_2} \end{bmatrix} \tag{4.12}$$

$$\mathbf{i} = \begin{bmatrix} \frac{\partial f_\alpha}{\partial \theta_0}^T \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \left(\Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \Lambda_\alpha \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \right)^{-1} \left(f_\beta - \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} f_\alpha \right) \\ -\frac{\partial f_\beta}{\partial \theta_2}^T \left(\Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} \Lambda_\alpha \frac{\partial f_\alpha}{\partial \theta_1}^{-T} \frac{\partial f_\beta}{\partial \theta_1}^T \right)^{-1} \left(f_\beta - \frac{\partial f_\beta}{\partial \theta_1} \frac{\partial f_\alpha}{\partial \theta_1}^{-1} f_\alpha \right) \end{bmatrix} \tag{4.13}$$

We can now equate the marginal information matrix and vector of the two step problem with the Information matrix and vector of the single step. From this set of 6 equations a simpler set of four conditions can be extracted:

$$f_{\alpha+\beta} = f_\beta - \frac{\partial f_\beta}{\partial \theta_1}\frac{\partial f_\alpha}{\partial \theta_1}^{-1} f_\alpha \tag{4.14}$$

$$\frac{\partial f_{\alpha+\beta}}{\partial \theta_0} = -\frac{\partial f_\beta}{\partial \theta_1}\frac{\partial f_\alpha}{\partial \theta_1}^{-1}\frac{\partial f_\alpha}{\partial \theta_0} \tag{4.15}$$

$$\frac{\partial f_{\alpha+\beta}}{\partial \theta_2} = \frac{\partial f_\beta}{\partial \theta_2} \tag{4.16}$$

$$\Lambda_{\alpha+\beta} = \Lambda_\beta + \frac{\partial f_\beta}{\partial \theta_1}\frac{\partial f_\alpha}{\partial \theta_1}^{-1}\Lambda_\alpha\frac{\partial f_\alpha}{\partial \theta_1}^{-T}\frac{\partial f_\beta}{\partial \theta_1}^{T} \tag{4.17}$$

These equations place conditions on the error metric value, Jacobians and covariance matrices evaluated at the linearisation points. The total error is given in (4.14) as a combination of the two steps by projecting the error from the first step via a transformation through θ_1. The Jacobians are related in (4.15) and (4.16). Both of these conditions can be derived from the first equation by calculating the derivative. Finally (4.17) combines the covariance terms by projecting the noise in the space of the first step to the space of the second. This is the condition that many vehicle models will not obey so careful consideration of appropriate vehicles models is required.

Trivial solutions can be found for any invertible metric where θ_1 is the value that results in a zero valued error metric, that is if $f_\alpha = 0$ then $f_\beta = f_{\alpha+\beta}$. The Jacobian conditions are directly the partial derivatives of the first condition on the metric and thus satisfied. Finally if the covariance matrix is set as zero for the first time step α then $\Lambda_{\alpha+\beta} = \Lambda_\beta$. This results in a constant offset between the two values θ_0 and θ_1 however and is not what is desired here.

In the search for more interesting solutions we restrict our search to the space where the vehicle model metric is of the form in (4.18) so we can make further insights. In this case u is at most a function of the initial state, the change in time and a measured constant - the latter two of which are constant for a given factor.

$$f(\theta_i, \theta_j, \delta t, m_{ij}) = \log\left(\ominus(\theta_i \oplus u(\theta_i, \delta t, m_{ij})) \oplus \theta_j\right) \tag{4.18}$$

Here our metric is the adjoint representation of the Lie Algebra representing the difference between the projected value from θ_i and the estimated value θ_j, and u is a function of any or all of the starting state θ_i, the time between states δt and m_{ij} an odometry observation. Given this the derivatives of the error metric relative to the end state are the identity and so in our two step equations $\frac{\partial f_\alpha}{\partial \theta_1}$ and $\frac{\partial f_\beta}{\partial \theta_2}$ are the identity. This is advantageous as $\frac{\partial f_\alpha}{\partial \theta_1}$ is the only matrix to be inverted in these conditions and is now a trivial operation.

The simplest set of models that obey this condition are linear. We have also found a set of models that work and contain translation, orientation and linear velocity components in two and three dimensions. These models project forward with linear motion but can calculate and distribute orientation changes evenly across multiple time steps. The pose is represented using group $SE(N)$ and the linear velocities are represented with $\mathbb{R}^N$ and are relative to the body frame.

Inclusion of angular velocities was investigated although deemed not possible due to interaction of time dependencies in the noise model, independent of orientation representation methods. The time dependence creates issues as the position or translation components are first order time dependent through the linear velocities and second order dependent through the angular velocities via the orientation components. This results in mixed order time dependence and complicating derivation of a compatible noise model. Inclusion of linear acceleration was initially considered as it would result in a matched order of dependence on time but was found to be too complex to calculate.

4.2.2 Vehicle Models for State Insertion

The vehicle models we investigated for usage with state insertion were based around planar (x, y, θ) or 6DOF $(x, y, z, \phi, \theta, \psi)$ states with inclusion of body velocity terms. We found models that conformed to our conditions of the previous subsection that had linear velocities, but not angular velocities. This results in a state representation using orientation, translation and linear velocity terms giving 5 dimensions in the plane and 9 dimensions in 6DOF space. We use the product of the SE(N) group with $\mathbb{R}^N$ to represent our states, with $N = 2$ for the planar version or $N = 3$ for 6DOF. Elements of the matrix Lie Group

$SE(2) \times \mathbb{R}^2$ are representable as

$$\theta = \begin{bmatrix} R & \mathbf{t} & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & 1 & \mathbf{0}_{1\times 2} & 0 \\ \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 1} & I_{2\times 2} & \mathbf{v} \\ \mathbf{0}_{1\times 2} & 0 & \mathbf{0}_{1\times 2} & 1 \end{bmatrix} \tag{4.19}$$

$$= \begin{bmatrix} R & \mathbf{t} & & \\ & 1 & & \\ & & I_2 & \mathbf{v} \\ & & & 1 \end{bmatrix} \tag{4.20}$$

and elements of the matrix Lie group $SE(3) \times \mathbb{R}^3$ are representable as

$$\theta = \begin{bmatrix} R & \mathbf{t} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 & \mathbf{0}_{1\times 3} & 0 \\ \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 1} & I_{3\times 3} & \mathbf{v} \\ \mathbf{0}_{1\times 3} & 0 & \mathbf{0}_{1\times 3} & 1 \end{bmatrix} \tag{4.21}$$

$$= \begin{bmatrix} R & \mathbf{t} & & \\ & 1 & & \\ & & I_3 & \mathbf{v} \\ & & & 1 \end{bmatrix} \tag{4.22}$$

We denote the algebra of these groups as $(t_x, t_y, \omega_\psi, v_u, v_v)$ and $(\omega_\phi, \omega_\theta, \omega_\psi, t_x, t_y, t_z, v_u, v_v, v_w)$.

We use the error metric form in (4.18) and here define

$$u(\theta_i, \delta t) = \exp\left(\begin{bmatrix} \mathbf{0}_{N\times N} & \delta t\mathbf{v} & \mathbf{0}_{N\times N+1} \\ \mathbf{0}_{1\times N} & 0 & \mathbf{0}_{1\times N+1} \\ \mathbf{0}_{N+1\times N} & \mathbf{0}_{N+1\times 1} & \mathbf{0}_{N+1\times N+1} \end{bmatrix}\right) \tag{4.23}$$

$$= \begin{bmatrix} I_{N\times N} & \delta t\mathbf{v} & \mathbf{0}_{N\times N+1} \\ \mathbf{0}_{1\times N} & 1 & \mathbf{0}_{1\times N+1} \\ \mathbf{0}_{N+1\times N} & \mathbf{0}_{N+1\times 1} & I_{N+1\times N+1} \end{bmatrix} \tag{4.24}$$

with no odometry observation m_{ij} and exp being the exponential map from the Lie Algebra to our Lie Group $SE(N) \times \mathbb{R}^N$. This predicted moves in the body velocity direction for the

specified time interval and includes no angular motion.

The models we found for odometry error involve calculating the difference between a forward prediction from θ_i at time t to θ_{i+1} at time $t + \delta t$ and the current estimate for state θ_{i+1}. The prediction function returns an update to the translation only as we do not use angular velocities or any acceleration terms.

Calculation of Jacobians and the Covariance matrix are complex due to the usage of non-Euclidean spaces and Lie Algebra (and matrix exponentials and logarithms). To this end we merely give the resulting Jacobians and covariance matrices with a high level explanation of the constituent portions. Due to the number of variables involved we will use some symbols that are generally familiar, such as R for rotation matrices, and less familiar such as the Adjoint matrix $[Ad_x]$ from Lie Group theory.

The Adjoint matrix for our spaces can be calculated in terms of rotation, translation and velocity coordinates (RTV) of a group element. The Adjoint matrix for an element θ of the planar group is

$$[Ad_\theta] = \begin{bmatrix} \cos\psi & -\sin\psi & y & & \\ \sin\psi & \cos\psi & -x & & \\ 0 & 0 & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \tag{4.25}$$

whilst for an element θ of the 6DOF group is

$$[Ad_\theta] = \begin{bmatrix} R & & \\ TR & R & \\ & & I \end{bmatrix} \tag{4.26}$$

where T is a skew symmetric matrix made from the translation elements

$$T = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{4.27}$$

This term arises frequency in calculating the derivatives of functions of Lie groups.

We begin with the 6DOF model as the planar model is a special case where height, roll, pitch and vertical velocity are 0. The Jacobians of the vehicle model error metric with respect to the initial or source state θ_0 and the end or destination state θ_1 are

$$\frac{\partial f}{\partial \theta_0} = -\left[Ad_{\theta_1^{-1}\oplus\theta_0}\right]\begin{bmatrix} I & & \\ & I & \delta t I \\ & & I \end{bmatrix} \tag{4.28}$$

$$= -\begin{bmatrix} R & & \\ TR & R & \delta t R \\ & & I \end{bmatrix} \tag{4.29}$$

$$\frac{\partial f}{\partial \theta_1} = I \tag{4.30}$$

where empty regions are zero elements.

The covariance is again more complex, and has two diagonal covariance matrices Λ_R and Λ_V representing the motion model noise in the orientation and linear velocity dimensions respectively. The central component represents the time dependence of each dimension on the noise matrices and the general form of the time dependence is familiar from linear constant velocity covariance models [9, p. 270].

$$\Lambda = \left[Ad_{\theta_1^{-1}\oplus\theta_0\oplus u}\right]\begin{bmatrix} \delta t\Lambda_R & \frac{\delta t^2}{2}\Lambda_R V & \\ \left(\frac{\delta t^2}{2}\Lambda_R V\right)^T & \frac{\delta t^3}{3}\left(\Lambda_V - V\Lambda_R V\right) & \frac{\delta t^2}{2}\Lambda_V \\ & \frac{\delta t^2}{2}\Lambda_V & \delta t\Lambda_V \end{bmatrix}\left[Ad_{\theta_1^{-1}\oplus\theta_0\oplus u}\right]^T \tag{4.31}$$

V here is the skew-symmetric matrix of elements in $\mathbf{v}$ as for (4.27).

The Jacobians and covariance models for the planar case, with scalar covariance values $\lambda_u, \lambda_v, \lambda_\psi$ are

$$\frac{\partial f}{\partial \theta_0} = -\left[Ad_{\theta_1^{-1}\oplus\theta_0}\right]\begin{bmatrix} I & & \delta t I \\ & 1 & \\ & & I \end{bmatrix} \tag{4.32}$$

$$= \begin{bmatrix} \cos\psi & -\sin\psi & y & \delta t\cos\psi & -\delta t\sin\psi \\ \sin\psi & \cos\psi & -x & \delta t\sin\psi & \delta t\cos\psi \\ 0 & 0 & 1 & & \\ & & & 1 & \\ & & & & 1 \end{bmatrix} \tag{4.33}$$

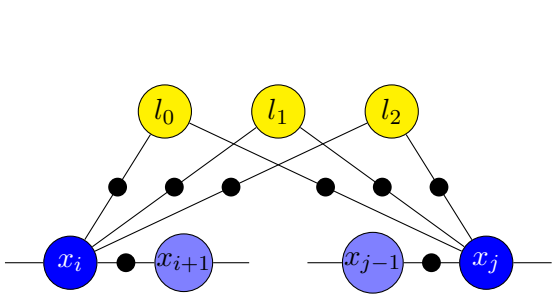
$$\frac{\partial f}{\partial \theta_1} = I \tag{4.34}$$

$$\Lambda = \left[Ad_{\theta_1^{-1}\oplus\theta_0\oplus u}\right] \begin{bmatrix} \frac{\delta t^3}{3}(\lambda_\psi v^2+\lambda_u) & -\frac{\delta t^3}{3}\lambda_\psi uv & -\frac{\delta t^2}{2}\lambda_\psi v & \frac{\delta t^2}{2}\lambda_u & \\ -\frac{\delta t^3}{3}\lambda_\psi uv & \frac{\delta t^3}{3}(\lambda_\psi u^2+\lambda_u) & \frac{\delta t^2}{2}\lambda_\psi u & & \frac{\delta t^2}{2}\lambda_v \\ -\frac{\delta t^2}{2}\lambda_\psi v & \frac{\delta t^2}{2}\lambda_\psi u & t\lambda_\psi & & \\ \frac{\delta t^2}{2}\lambda_u & & & t\lambda_u & \\ & \frac{\delta t^2}{2}\lambda_v & & & t\lambda_v \end{bmatrix} \left[Ad_{\theta_1^{-1}\oplus\theta_0\oplus u}\right]^T \tag{4.35}$$

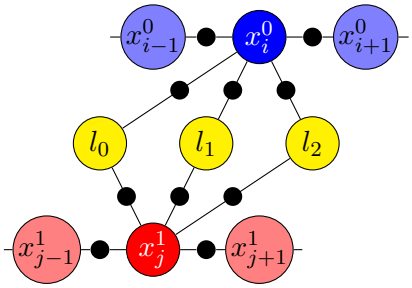
4.3 Results

As in the previous chapter we compare against a centralised relinearising solution and individual solutions for each platform. We use the method from the previous chapter as a comparison to illustrate the effect of the state insertion method to include out of sequence observations. This provides us with four base estimators. We then use two variants of each - one with and one without loop closures from landmark observations to provide eight estimator solutions. The method from the last chapter and the new one including state insertion use the state promotion method to handle loop closures as it is general for all vehicle models.

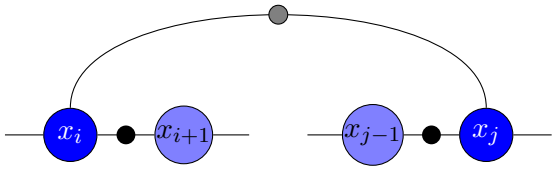
Our results here are calculated using our estimators on the UTIAS datasets [47], this time based only on a homogeneous configuration with the addition of subsampled landmark observations. Rather than including landmark states in the estimator as additional variables we use measurements of landmark states to generate observations of the relative pose and associated covariance between vehicle poses and include these as factors. Figure 4.6 shows two situations in which we can generate this new factor from multiple landmark observations. Figure 4.6a shows a local loop closure which we utilise here whilst Figure 4.6b shows a loop closure with a remote vehicle which we do not utilise here due to extra data transfer required for data association. We calculate the landmarkless loop closure factor for both of



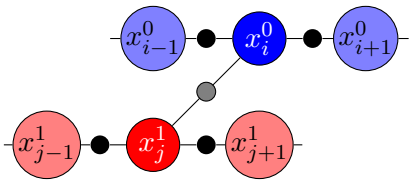
(a) Local loop closure with landmarks.



(b) Intervehicle loop closure with landmarks.

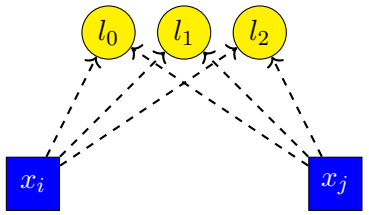


(c) Local loop closure after landmarkless loop closure construction.

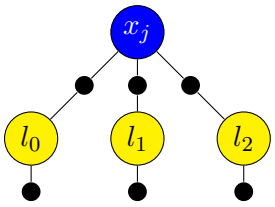


(d) Intervehicle loop closure after landmarkless loop closure construction.

Figure 4.6 – Factor graphs before (a) (b) creation of landmarkless loop closure factors and after (c) (d). The creation of these loop closures enables us to avoid including extra states in the estimator. Double counting and inconsistency is avoided through usage of each landmark observation for a single landmarkless factor.



(a) Vehicle states and landmark observations.



(b) Constrained factor graph.

Figure 4.7 – The physical structure and constrained factor graph for the optimisation of loop closure candidates. The factor graph only shows one state as the starting state is constrained and thus a constant. The binary factors that would have connected the landmarks to it are then unary factors connected only to the landmarks. The marginal probability of the end state x_j from this graph provides the conditional probability covariance relative to the starting state x_i.

these giving the local case in Figure 4.6c and the intervehicle case in Figure 4.6d.

This map free approach has origins in the original graph smoothing solutions with laser scan matching [49] and more recently with stereo and monocular frame matching for visual odometry and loop closures [21, 34, 41, 50]. These methods eschew addition of the features, or lack features to add, and instead compute a pose offset and covariance from the measurements. These values are then used as an observation in the localisation method.

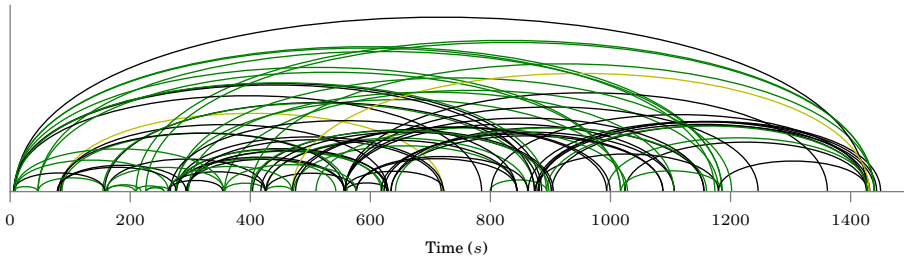
Here we calculate the loop closures through nonlinear optimisation over the range and bearing observations in the dataset to the commonly observed landmarks between two poses as depicted in Figure 4.7. We avoid dependence on the current linearisations of the states by constraining the earlier state x_i in the loop closure candidate to be the origin and solving for the later state x_j relative to this. Initial linearisations for the reduced problem are calculated geometrically from the observations to avoid any dependence on current estimates. We iterate until convergence, discarding any potential loop closures that don't converge within 50 iterations. The landmark states are marginalised out and the marginal probability of x_j is extracted and used as the covariance of the loop closure factor. We also utilise innovation gating through a one sided χ^2 filter to ignore loop closures that are unlikely based on state estimates at the time of capture using a threshold of 99% (which equates to a Normalised Innovation Squared of approximately 11.34 for a three dimensional observation). All estimators then receive the same set of measurements, including loop closures, however due to differences in estimates the loop closures may be rejected by each estimator.

To better simulate the real world case we also introduce a delay to message passing between platforms. In each case the communication delay is a fixed amount plus a range dependent component to simulate computation time and the slower propagation delay in either large range scenarios or slower communication mediums such as acoustic communications underwater.

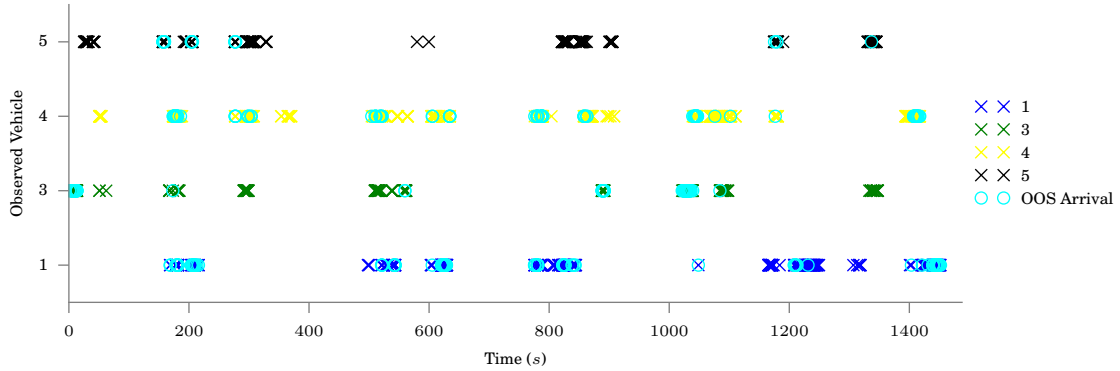
4.3.1 Captured Data

Given the desire to reduce bandwidth the selection of loop closure candidates from the total possible pool deserves some attention. As a coarse measure to decrease the covariance we required three common landmark observations although only two are required for a full rank observation. As this is also not intended as a visual odometry solution a minimum time between start and end states of a loop closure was required of 30 seconds. Additionally repeat observations, or similar observations that link similar start and end states were limited to reduce bandwidth and favour more informative loop closures. To this end the difference of time at the start states added to the difference in time at the end states was enforced to be above 10 seconds.

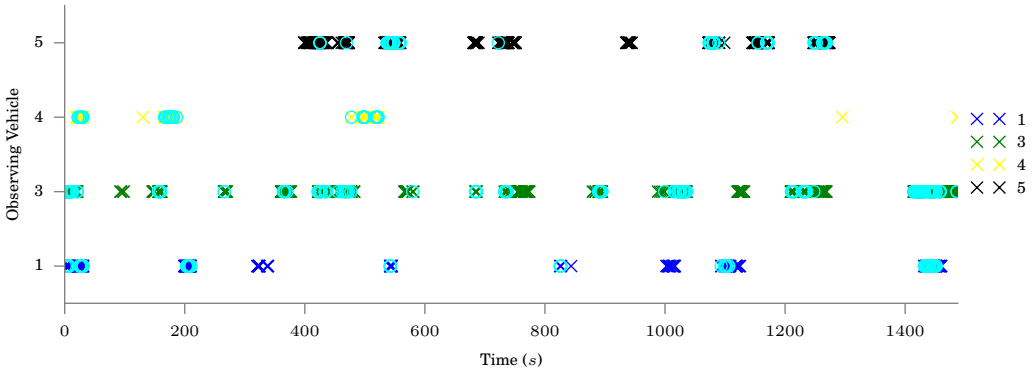
Detailed results for the first dataset in Figure 4.9 show that inclusion of loop closures has a significant effect - even without loop closures between vehicles. As can be predicted the presence of loop closures provides a constraint on the odometry drift that results in a near constant uncertainty level over time [28] whilst the solutions that do not use landmarks exhibit growth in uncertainty of odometry only solutions as can be calculated from the Kalman filter equations. The use of intervehicle measurements still results in significant reduction in uncertainty and error, even with the inclusion of loop closures. Of greater interest however is the relative performance of decentralised estimators with and without state insertion. Looking at the error and covariance the methods that can handle delayed observations gain slightly - however it is not a large gain in this situation. Table 4.2 shows that approximately 25% of intervehicle measurements arrived out of sequence for dataset 1. The promotion method was able to handle 3% of the observations but insertion of additional states was required for the other out of sequence observations. Overall this is an increase of approximately 30% of usable measurements when using state insertion compared to our base method. Looking at the time most of these observations took place in Figures 4.8b and 4.8c the grouping of these with other in sequence observations shows the relatively small gain that



(a) Loop closure connections for vehicle 2. A χ^2 filter on each estimator resulted in some valid loop closures being rejected and so there is variation in what each estimator actually used. Black links indicate all estimators used the link whilst green indicates the centralised estimator did not. Yellow indicates only MVSAM+Insertion+Landmarks used the link.

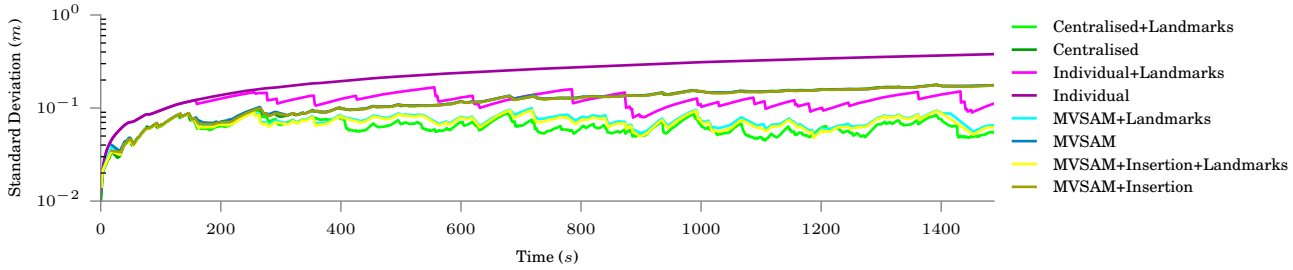


(b) Times at which each vehicle was observed by vehicle 2. Cyan circles indicate that the observation arrived out of sequence in the MVSAM solution with landmarks and OOS insertion.

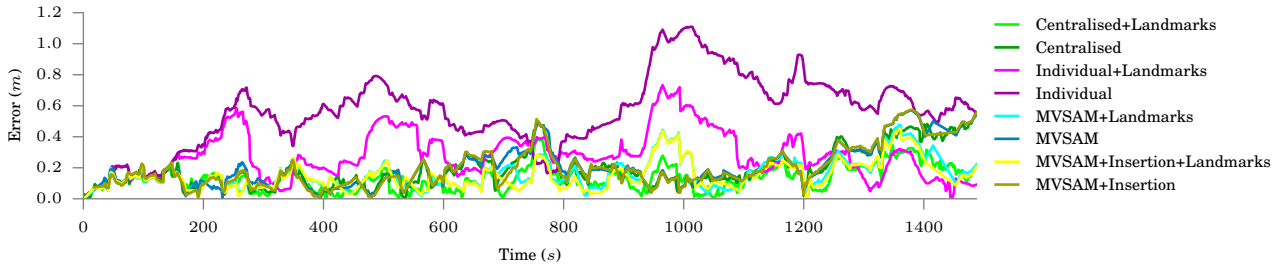


(c) Times at which vehicle 2 was observed by other vehicles. Cyan circles indicate that the observation arrived out of sequence in the MVSAM solution with landmarks and OOS insertion.

Figure 4.8 – Vehicle 2 from dataset 1 showing loop closure candidates and intervehicle observations. Figure 4.9 has the position standard deviation and error for this data.



(a) Radius of circle approximation to standard deviation around position of vehicle 2 at each point in time.



(b) RMS position error for vehicle 2 from each optimiser.

Figure 4.9 – Vehicle 2 from dataset 1 showing estimator uncertainty and estimator error. The connections with other vehicles are shown in Figure 4.8. The usefulness of even local loop closures is apparent with the performance of the individual landmark solutions exceeding the centralised and decentralised solutions without landmarks. However the usage of landmarks and intervehicle observations permit further improvements in estimator error and certainty over the methods of the previous chapter.

Vehicle		Landmarks			Insertion			Landmarks+Insertion		
		IS	OOS-I	OOS-P	IS	OOS-I	OOS-P	IS	OOS-I	OOS-P
1	EXT	1827	0	72	1810	216	36	1827	235	70
	IV	1788	0	38	1809	216	36	1788	235	38
	LC	92	0	34	0	0	0	88	0	32
2	EXT	1956	0	102	1919	376	43	1956	398	104
	IV	1889	0	50	1918	376	43	1889	398	50
	LC	123	0	52	0	0	0	125	0	54
3	EXT	2218	0	110	2152	294	38	2218	300	110
	IV	2141	0	42	2151	294	38	2141	300	42
	LC	148	0	68	0	0	0	150	0	68
4	EXT	2019	0	50	2018	311	45	2019	311	50
	IV	2015	0	47	2017	311	45	2015	311	47
	LC	9	0	3	0	0	0	11	0	3
5	EXT	1920	0	92	1832	164	24	1920	169	92
	IV	1824	0	26	1831	164	24	1824	169	26
	LC	231	0	66	0	0	0	231	0	66

Table 4.1 – Counts of external states (EXT) on each robot that were generated in sequence (IS) or out of sequence via insertion (OOS-I) or promotion (OOS-P) methods for intervehicle (IV) and local loop closure (LC) observations in dataset 1 as well as the total count of external states. This gives a sense of data transfer requirements, assuming no duplication of transmissions where Table 4.2 better indicates how many observations were rejected due to lack of out of sequence handling. Two states are required for each intervehicle or loop closure observation and one of these states is always generated 'in sequence' at the time of observation whilst the other state may correspond to an existing external state, arrive in sequence or require an out of sequence handler. Additionally the number of external states generated does not match the sum of the count of states involved in local loop closures and intervehicle observations due to overlap between the two sets.

Vehicle		Landmarks			Insertion			Landmarks+Insertion		
		IS	OOS-I	OOS-P	IS	OOS-I	OOS-P	IS	OOS-I	OOS-P
1	IV	596	0	46	599	261	40	596	258	46
	LC	32	0	31	0	0	0	31	0	29
2	IV	816	0	44	822	273	43	816	278	44
	LC	36	0	51	0	0	0	36	0	53
3	IV	1250	0	60	1289	440	55	1250	474	60
	LC	40	0	67	0	0	0	41	0	67
4	IV	548	0	12	551	148	12	548	151	12
	LC	3	0	3	0	0	0	4	0	3
5	IV	1309	0	43	1328	244	38	1309	258	43
	LC	85	0	65	0	0	0	85	0	65
Total	IV	95.7%	0.0%	4.3%	74.7%	22.2%	3.1%	73.6%	23.1%	3.3%
	LC	47.5%	0.0%	52.5%				47.6%	0.0%	52.4%

Table 4.2 – Counts of method used to handle intervehicle observations made on each vehicle in dataset 1. Local loop closures here are from the same sensor as intervehicle observations so alignment of approximately 50% of loop closure observations with existing external states is not surprising. However approximately 22% of observations arrived out of sequence and required insertion and another 3% aligned with an existing internal state.

Dataset		Landmarks			Insertion			Landmarks+Insertion		
		IS	OOS-I	OOS-P	IS	OOS-I	OOS-P	IS	OOS-I	OOS-P
1	IV	95.7%	0.0%	4.3%	74.7%	22.2%	3.1%	73.6%	23.1%	3.3%
	LC	47.5%	0.0%	52.5%				47.6%	0.0%	52.4%
2	IV	95.1%	0.0%	4.9%	70.2%	26.6%	3.2%	68.8%	27.7%	3.6%
	LC	40.6%	0.0%	59.4%				40.2%	0.0%	59.8%
3	IV	95.8%	0.0%	4.2%	75.8%	21.3%	2.9%	74.7%	22.0%	3.3%
	LC	58.0%	0.0%	42.0%				57.4%	0.0%	42.6%
4	IV	96.0%	0.0%	4.0%	74.8%	22.6%	2.6%	74.1%	22.9%	3.1%
	LC	75.9%	0.0%	24.1%				75.9%	0.0%	24.1%
5	IV	94.3%	0.0%	5.7%	67.8%	28.6%	3.6%	66.4%	29.5%	4.0%
	LC	52.1%	0.0%	47.9%				51.8%	0.0%	48.2%
6	IV	94.7%	0.0%	5.3%	67.8%	28.7%	3.4%	67.1%	29.2%	3.7%
	LC	56.2%	0.0%	43.8%				56.4%	0.0%	43.6%
7	IV	96.0%	0.0%	4.0%	74.0%	23.3%	2.7%	72.3%	24.6%	3.0%
	LC	50.4%	0.0%	49.6%				50.0%	0.0%	50.0%

Table 4.3 – Showing the percentage each method was used to handle sharing of state data for shared observations in each dataset. Overall we see the rate of out of sequence varies, however the relative ratio of promotion vs insertion for handling intervehicle observations remains constant.

was to be found is due to these measurements being closely followed or preceded by many other measurements of the same target and thus being relatively insignificant. Table 4.1 shows the count of external states that are shared between vehicles and the method of generating the packet for each decentralised estimator.

Interestingly the local loop closure based methods vary more in their estimates and uncertainty than the intervehicle only methods. Possible cause for this relates to the partially fixed linearisation of remote states being unable to correct for odometry drift when updated with loop closures distorting the estimator. Whereas in the case without loop closures the entire system solution may drift from the mean together the loop closures pull on the graph at varying rates and accuracies on each vehicle. The centralised estimator is better able to accommodate these updates as it can update the linearisation for all vehicles.

Comparison of out of sequence observation counts for all datasets we tested are summarised in Table 4.3. Here we see that the percentage that are out of sequence varies between approximately 25% and 30% in our datasets and so remains fairly stable for our constant configuration. We would expect variation in the rate of intervehicle observations or delay in message passing to provide the most significant effects on this. The loss of these intervehicle measurements does reduce certainty and increase error although the effect is relatively small.

Finally the communication bandwidth required for each estimator is given in Table 4.4.

Dataset	Raw	MVSAM		+Insertion	
1	4114 kiB	2894 kiB	70.3%	2916 kiB	70.9%
2	5130 kiB	4027 kiB	78.5%	4114 kiB	80.2%
3	4146 kiB	2651 kiB	64.0%	2683 kiB	64.7%
4	3908 kiB	2805 kiB	71.8%	2819 kiB	72.1%
5	6514 kiB	4906 kiB	75.3%	4989 kiB	76.6%
6	2043 kiB	1493 kiB	73.1%	1498 kiB	73.3%
7	2421 kiB	1795 kiB	74.2%	1810 kiB	74.8%
Total	28276 kiB	20571 kiB	72.8%	20829 kiB	73.7%

(a) Raw and summarised usage with landmark data usage for loop closures.

Dataset	Raw	MVSAM		+Insertion	
1	4068 kiB	2531 kiB	62.2%	2560 kiB	62.9%
2	5025 kiB	3304 kiB	65.8%	3317 kiB	66.0%
3	4079 kiB	2317 kiB	56.8%	2342 kiB	57.4%
4	3837 kiB	2538 kiB	66.2%	2559 kiB	66.7%
5	6365 kiB	4166 kiB	65.5%	4208 kiB	66.1%
6	2012 kiB	1322 kiB	65.7%	1325 kiB	65.9%
7	2387 kiB	1582 kiB	66.3%	1589 kiB	66.6%
Total	27773 kiB	17760 kiB	63.9%	17900 kiB	64.5%

(b) Raw and summarised data usage with no usage of landmark data for loop closures.

Table 4.4 – Total data and percentage used relative to transmission of the raw measurements for the UTIAS datasets. The summarisation methods result in 27% reduction for loop closure scenarios and 35% reduction without loop closures. The difference in data savings is due primarily to the extra bandwidth required for out of sequence promotion for the historic states in loop closure factors. Alternate heuristics for selecting raw or summarised transmission of data could assist in reducing data further.

Overall we still retain reduction in bandwidth of around 27% when including landmarks and 35% without. The difference in bandwidth reduction with out without landmarks is due in part to the bandwidth required for the loop closure which includes the noise model as well as calculated offset but also requires inclusion of the out of sequence summary data for the earlier state. When transmitting raw data no extra information needs to be transmitted as remote systems can calculate the earlier state information. When using summarised data the out of sequence data needs to be transmitted and this equates to around 7% of the 8% difference when using loop closures – the other 1% is for the loop closures factors themselves. However there is scope to reduce the number of these loop closures shared with smarter mechanics or simplification of the observations. Smarter mechanics could adjust when raw observations are transmitted instead of summarised data. Currently we use the summary algorithm if it results in less data than raw transmission. This threshold could be adjusted to be 10% less data than raw transmission or other heuristics based on the likelihood of an out of sequence observation occurring in the current subgraph. The loop closure observations could also be simplified to be only the calculated mean offset with a constant covariance instead of deriving the covariance from the simplified SLAM problem. Inclusion of inserted states also increases the bandwidth requirement but amounts to around 1% reduction in savings on the base method.

4.4 Summary

Overall we see the inclusion of local landmark loop closures and their sharing across vehicles provides the largest benefits to error and reduced uncertainty. This does come at a larger cost to communications although this can be reduced through either more careful selection of which loop closures to share or a constant covariance assumption for loop closure noise. The ability to transmit for out of sequence observations does reduce uncertainty and error but is small relative to the reduction of error due to loop closures and places restrictions upon the vehicle model that do not suit all applications. It does however work for the models and application we propose here.

Chapter 5

Conclusion

5.1 Summary

This thesis develops a new method for multiple robot localisation. The method handles asynchronous updates between robots and minimises bandwidth. Importantly the method is decentralised giving resilience against agent failure and distributes much of the computation. Additionally we develop algorithms to handle significant delays including out of sequence arrival of information from remote platforms without retransmission or significant recalculation. We verified this with simulated and captured data for sets of small robot teams. The method reduced the bandwidth requirements in our tests on captured data by approximately 50% and further benefits would arise in cases with more erratic communication where vehicles do not interact with others for longer periods. We calculated the savings for various combinations of regular internal and external sensor rates and state space size. Whilst we did get a measure of the handling with erratic communications in the UTIAS dataset tests they are not exhaustive or necessarily representative of other systems.

We extended our decentralised method to handle communication delays that can arise from slow signal propagation and computation times or communication range limitations. This method does not require retransmission of existing data and is purely additive requiring the same simple incorporation methods that our base method uses. This involved two related extensions. The first handles delayed messages that connect to an existing internal state on a remote vehicles estimator. The second handles delayed messages through the insertion of a new state. This can be done without requiring retransmission given a vehicle model

that conforms to a set of conditions we derived from equating one and two odometry step localisation solutions.

Furthermore we applied these methods to enable sharing of local loop closure information. In this case vehicles calculate local loop closures that connected vehicle states over long periods and then share the loop closure information with remote vehicles. Using local loop closures reduces the growth in covariance similar to the effect for single vehicles and SLAM and can lead to significant improvements in estimates. The method also permits loop closures between vehicles, although it is expected that the bandwidth cost of calculating the data association is prohibitive and was not investigated.

5.2 Contributions

Four novel contributions to the state of the art in multiple robot decentralised localisation with low bandwidth communication were presented.

The first contribution was the development of a method for information fusion that allows iterative in-sequence construction and transmission of a summary graph containing only the states involved in intervehicle measurements. It does not require retransmission or special incorporation such as removing previously transmitted data before inclusion. The method developed is exact, both in the sense of no approximations and producing an identical solution for a single iteration of a nonlinear optimisation solver. Differences to a centralised solution emerge due to the ability of a centralised (or multi-centralised) solver to relinearise all states whilst our method is limited to relinearising only local states.

The second and third contributions were the extension of the previous method to facilitate the inclusion of out of sequence intervehicle measurements without retransmission. The two contributions handle two cases of delayed observations that cannot be implicitly handled by the base method. The first of these two contributions handles delayed observations that connect to an internal state. This most obviously arises with landmarkless loop closures which we demonstrate. The method involves reordering elimination in a subgraph between two existing external states to ‘promote’ the targeted internal state to an external state. The subgraph structure developed for the base method when used for state promotion reduces the required computation - which is a factorisation update and does not involve linearisation of factors.

The third contribution and second method for handling delayed and out of sequence messages concerns the case where no state exists at the target time on the target vehicle. This arises where vehicles are passively observable such as through vision systems, LIDAR or RADAR. Vehicles are unaware of their being observed and so cannot create a state to link to the remote vehicle. The method handles insertion of a new intermediate state and fast ways to calculate this that reduce the number of factors that need to be linearised and involved in refactorisation. Unlike our second contribution this is not applicable to all vehicles and we derive the conditions that vehicle models must conform to. The derivation compares the joint marginal probability of two states with and without an intermediate state and generates four constraints that cover the error metric, Jacobians of the error metric with respect to the first or second state variable and the error metric covariance.

Our fourth contribution involved derivation of two vehicle models that conform to the conditions derived in the third contribution. These models cover motion in the plane and general 6DOF movement and enable usage on real systems.

As a whole these contributions permit efficient calculation of a near centralised equivalent cooperative localisation solution. Significantly, when compared to existing cooperative localisation algorithms, it does not disregard late arriving information or out of sequence messages, which in our tests represent approximately 25% of all intervehicle observations. Nor does it require retransmission of data, reducing bandwidth and the need for more complex information tracking. We utilise this to demonstrate usage of local loop closures, reducing the growth in uncertainty and need for association of landmarks across platforms.

5.3 Future Work

A number of avenues for future work exist surrounding selection of informative shared states instead of complete sharing, reduction in computation required for factorisation updates through alternate variable ordering in subgraph elimination and extension to more general landmark solutions.

Improved selection of loop closures to be shared across platforms is the simplest way to reduce bandwidth further with the potential to reduce the number of shared states whilst retaining the constraints on uncertainty that loop closures permit. This can involve selection using Information based metrics such as entropy. Alternatively usage of a constant noise

model for loop closures instead of extracting the uncertainty from the reduction of features would reduce bandwidth for the loop closures themselves and may not affect the final solution significantly.

An alternative improvement is anticipation of state promotion for loop closures when transmitting data for a segment. The current algorithm uses the compressed data if the size of raw data transmission exceeds the compressed packet size. Without out of sequence observations this guarantees that the algorithm will use less bandwidth than complete raw transmission. With out of sequence observations this is not guaranteed (although still likely). Anticipation of out of sequence packets for loop closures and transmission of raw data for the related segments could result in further reduction of bandwidth and could come from knowledge of the robots planned motion or tracking of potential loop closure observations.

Modification of the simple odometry chain order proposed with the initial packet calculation to one based on a binary tree could be used to reduce the number of affected variables should a state require sharing. Changing the Bayes Tree generated from narrow and tall to wider and shorter could reduce the average number of nodes affected upon reordering. Currently it is worst case n in the number of states, but could be reduced.

Incorporation of landmarks in the graph in an iterative manner would enable a general multiple robot SLAM solution and may have benefits where vehicles only see a small number of landmarks at any point in time. Iterative transmission of this data could assist in building larger shared maps with reduced or more even bandwidth usage than comparable batch transmission methods such as DDF-SAM [13, 14]. This also can be compared with the landmarkless solutions we use and enable determination on a system by system basis whether landmarks and data association or loop closures yield reduced bandwidth and improved accuracy of the final solution.

Relinearisation of the shared graphs could result in improved solutions where initial estimates are poor approximations to the true value. Investigation of this could include merging of adjacent packets to reduce bandwidth required on retransmission. Additionally determination of update priority for variables based on delta size and stability of estimates could assist in using bandwidth optimally.

Bibliography

[1] Pratik Agarwal and Edwin Olson. Variable reordering strategies for SLAM. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 3844–3850, Oct 2012.

[2] Motilal Agrawal. A Lie algebraic approach for consistent pose registration for general euclidean motion. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 1891–1897. IEEE, 2006.

[3] Motilal Agrawal and Kurt Konolige. Real-time localization in outdoor environments using stereo vision and inexpensive GPS. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 3, pages 1063–1068. IEEE, 2006.

[4] Lars AA Andersson and Jonas Nygards. C-sam: Multi-robot slam using square root information smoothing. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 2798–2805. IEEE, 2008.

[5] Hajime Asama, Akihiro Matsumoto, and Yoshiki Ishida. Design of an autonomous and distributed robot system: Actress. In *Intelligent Robots and Systems, 1989.(IROS 1989). 1989 IEEE/RSJ International Workshop on*, pages 283–290, 1989.

[6] Alexander Bahr, Matthew R Walter, and John J Leonard. Consistent cooperative localization. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3415–3422. IEEE, 2009.

[7] Tim Bailey, Juan Nieto, Jose Guivant, Michael Stevens, and Eduardo Nebot. Consistency of the EKF-SLAM algorithm. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3562–3568. IEEE, 2006.

[8] Tim Bailey, Mitch Bryson, Hua Mu, John Vial, Lachlan McCalman, and Hugh Durrant-Whyte. Decentralised cooperative localisation for heterogeneous teams of mobile robots. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2859–2865. IEEE, 2011.

[9] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2001.

[10] Rodney Brooks et al. Visual map making for a mobile robot. In *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, volume 2, pages 824–829. IEEE, 1985.

[11] Y Uny Cao, Alex S Fukunaga, and Andrew Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1):7–27, 1997.

[12] Gregory S Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media, 2011.

[13] Alexander Cunningham, Manohar Paluri, and Frank Dellaert. DDF-SAM: Fully distributed slam using constrained factor graphs. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3025–3030. IEEE, 2010.

[14] Alexander Cunningham, Vadim Indelman, and Frank Dellaert. DDF-SAM 2.0: Consistent distributed smoothing and mapping. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 05/2013 2013.

[15] Frank Dellaert and Michael Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1203, 2006.

[16] John E Dennis Jr and Robert B Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996.

[17] Gamini Dissanayake, Paul Newman, Steven Clark, Hugh F Durrant-Whyte, and Michael Csorba. A solution to the simultaneous localization and map building (SLAM) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, 2001.

[18] Tom Duckett, Stephen Marsland, and Jonathan Shapiro. Learning globally consistent maps by relaxation. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 4, pages 3841–3846. IEEE, 2000.

[19] Hugh F Durrant-Whyte. Uncertain geometry in robotics. *Robotics and Automation, IEEE Journal of*, 4(1):23–31, 1988.

[20] Carlos Estrada, José Neira, and Juan D Tardós. Hierarchical SLAM: Real-time accurate mapping of large environments. *Robotics, IEEE Transactions on*, 21(4): 588–596, 2005.

[21] Ryan Eustice, Oscar Pizarro, and Hanumant Singh. Visually augmented navigation in an unstructured environment using a delayed state history. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 25–32. IEEE, 2004.

[22] Ryan M Eustice, Hanumant Singh, and John J Leonard. Exactly sparse delayed-state filters for view-based SLAM. *Robotics, IEEE Transactions on*, 22(6):1100–1114, 2006.

[23] Ryan M Eustice, Louis L Whitcomb, Hanumant Singh, and Matthew Grund. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4257–4264. IEEE, 2007.

[24] John Folkesson, Patric Jensfelt, Henrik Christensen, et al. Graphical SLAM using vision and the measurement subspace. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 325–330. IEEE, 2005.

[25] Udo Frese and Gerd Hirzinger. Simultaneous localization and mapping-a discussion. In *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics*, pages 17–26, 2001.

[26] Toshio Fukuda and Yoshio Kawauchi. Cellular robotic system (cebot) as one of the realization of self-organizing intelligent universal manipulator. In *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pages 662–667. IEEE, 1990.

[27] Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.

[28] Peter W Gibbens, Gamini Dissanayake, and Hugh F Durrant-Whyte. A closed form solution to the single degree of freedom simultaneous localisation and map building (SLAM) problem. In *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, volume 1, pages 191–196. IEEE, 2000.

[29] Matteo Golfarelli, Dario Maio, and Stefano Rizzi. Elastic correction of dead-reckoning errors in map building. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 905–911. IEEE, 1998.

[30] Gene H Golub and Charles F Van Loan. *Matrix computations*. JHU Press, 3 edition, 1996.

[31] Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, Udo Frese, and Christoph Hertzberg. Hierarchical optimization on manifolds for online 2d and 3d mapping. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 273–278. IEEE, 2010.

[32] Giorgio Grisetti, Rainer Kümmerle, and Kai Ni. Robust optimization of factor graphs by using condensed measurements. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 581–588. IEEE, 2012.

[33] Jose E Guivant, Favio R Masson, and Eduardo M Nebot. Simultaneous localization and map building using natural features and absolute information. *Robotics and Autonomous Systems*, 40(2):79–90, 2002.

[34] Andrew Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 3946–3952. IEEE, 2008.

[35] Shoudong Huang and Gamini Dissanayake. Convergence and consistency analysis for extended Kalman filter based SLAM. *Robotics, IEEE Transactions on*, 23(5): 1036–1049, 2007.

[36] Simon J Julier and Jeffrey K Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Robotics and Automation, 2001.*

Proceedings 2001 ICRA. IEEE International Conference on, volume 4, pages 4238–4243. IEEE, 2001.

[37] Michael Kaess, Ananth Ranganathan, and Frank Dellaert. iSAM: Incremental smoothing and mapping. *Robotics, IEEE Transactions on*, 24(6):1365–1378, 2008.

[38] Michael Kaess, Viorela Ila, Richard Roberts, and Frank Dellaert. The Bayes tree: An algorithmic foundation for probabilistic robot mapping. In *Algorithmic Foundations of Robotics IX*, pages 157–173. Springer, 2011.

[39] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *The International Journal of Robotics Research*, 31(2):216–235, 2012.

[40] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[41] Kurt Konolige, Motilal Agrawal, and Joan Sola. Large-scale visual odometry for rough terrain. In *Robotics Research*, pages 201–212. Springer, 2011.

[42] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g^2o: A general framework for graph optimization. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3607–3613. IEEE, 2011.

[43] Ryo Kurazume, Shigemi Nagata, and Shigeo Hirose. Cooperative positioning with multiple robots. In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pages 1250–1257. IEEE, 1994.

[44] Maria Teresa Lazaro, Lina María Paz, Pedro Pinies, Jose Castellanos, Giorgio Grisetti, et al. Multi-robot SLAM using condensed measurements. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1069–1076. IEEE, 2013.

[45] John J Leonard and Hans Jacob S Feder. Decoupled stochastic mapping [for mobile robot & AUV navigation]. *Oceanic Engineering, IEEE Journal of*, 26(4):561–571, 2001.

[46] Keith YK Leung, Timothy D Barfoot, and Hugh HT Liu. Decentralized localization of sparsely-communicating robot networks: A centralized-equivalent approach. *Robotics, IEEE Transactions on*, 26(1):62–77, 2010.

[47] Keith YK Leung, Yoni Halpern, Timothy D Barfoot, and Hugh HT Liu. The UTIAS multi-robot cooperative localization and mapping dataset. *The International Journal of Robotics Research*, 30(8):969–974, 2011.

[48] Andrew W Long, Kevin C Wolfe, Michael Mashner, and Gregory S Chirikjian. The banana distribution is Gaussian: A localization study with exponential coordinates. In *Robotics: Science and Systems*, 2012.

[49] Feng Lu and Evangelos Milios. Globally consistent range scan alignment for environment mapping. *Autonomous robots*, 4(4):333–349, 1997.

[50] Ian Mahon, Stefan B Williams, Oscar Pizarro, and Matthew Johnson-Roberson. Efficient view-based SLAM using visual loop closures. *Robotics, IEEE Transactions on*, 24(5):1002–1014, 2008.

[51] Peter S Maybeck. *Stochastic models, estimation, and control*, volume 141 of *Mathematics in Science and Engineering*. 1979.

[52] Eric J Msechu, Stergios I Roumeliotis, Alejandro Ribeiro, and Georgios B Giannakis. Decentralized quantized Kalman filtering with scalable communication cost. *Signal Processing, IEEE Transactions on*, 56(8):3727–3741, 2008.

[53] Esha D Nerurkar and Stergios I Roumeliotis. Asynchronous multi-centralized cooperative localization. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4352–4359. IEEE, 2010.

[54] Esha D Nerurkar, Ke X Zhou, and Stergios I Roumeliotis. A hybrid estimation framework for cooperative localization under communication constraints. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 502–509. IEEE, 2011.

[55] Eric W Nettleton, Hugh F Durrant-Whyte, Peter W Gibbens, and Ali H Göktogan. Multiple-platform localization and map building. In *Intelligent Systems and Smart Manufacturing*, pages 337–347. International Society for Optics and Photonics, 2000.

[56] Kai Ni, Drew Steedly, and Frank Dellaert. Tectonic sam: Exact, out-of-core, submap-based SLAM. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1678–1685. IEEE, 2007.

[57] Frank C Park, James E Bobrow, and Scott R Ploen. A Lie group formulation of robot dynamics. *The International Journal of Robotics Research*, 14(6):609–618, 1995.

[58] Mark A Paskin. Thin junction tree filters for simultaneous localization and mapping. In Georg Gottlob and Toby Walsh, editors, *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1157–1164, San Francisco, CA, 2003. Morgan Kaufmann Publishers.

[59] Liam Paull, Guoquan Huang, Mae Seto, and John J Leonard. Communication-constrained multi-auv cooperative slam. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 509–516. IEEE, 2015.

[60] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1998.

[61] Ioannis M Rekleitis, Gregory Dudek, and Evangelos E Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *International Joint Conference on Artificial Intelligence*, volume 15, pages 1340–1345. Lawrence Erlbaum Associates Ltd, 1997.

[62] Alejandro Ribeiro, Georgios B Giannakis, and Stergios I Roumeliotis. SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations. *Signal Processing, IEEE Transactions on*, 54(12):4782–4795, 2006.

[63] Stergios I Roumeliotis and George A Bekey. Distributed multirobot localization. *Robotics and Automation, IEEE Transactions on*, 18(5):781–795, 2002.

[64] Arthur C Sanderson. A distributed algorithm for cooperative navigation among multiple mobile robots. *Advanced Robotics*, 12(4):335–349, 1997.

[65] David Schleicher, Luis M Bergasa, Manuel Ocaña, Rafael Barea, and María Elena López. Real-time hierarchical outdoor SLAM based on stereovision and GPS fusion. *Intelligent Transportation Systems, IEEE Transactions on*, 10(3):440–452, 2009.

[66] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[67] Randall C Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68, 1986.

[68] Sebastian Thrun. A probabilistic on-line mapping algorithm for teams of mobile robots. *The International Journal of Robotics Research*, 20(5):335–363, 2001.

[69] Sebastian Thrun and Michael Montemerlo. The graph SLAM algorithm with applications to large-scale mapping of urban structures. *The International Journal of Robotics Research*, 25(5-6):403–429, 2006.

[70] Sebastian Thrun, Yufeng Liu, Daphne Koller, Andrew Y Ng, Zoubin Ghahramani, and Hugh Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *The International Journal of Robotics Research*, 23(7-8): 693–716, 2004.

[71] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment – a modern synthesis. In *Vision algorithms: theory and practice*, pages 298–372. Springer, 2000.

[72] John Vial, Hugh Durrant-Whyte, and Tim Bailey. Conservative sparsification for efficient and consistent approximate estimation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 886–893. IEEE, 2011.

[73] Jeffrey M Walls and Ryan M Eustice. An exact decentralized cooperative navigation algorithm for acoustically networked underwater vehicles with robustness to faulty communication: Theory and experiment. In *Proceedings of the Robotics: Science & Systems Conference*, Berlin, Germany, June 2013.

[74] Jeffrey M Walls, Alexander G Cunningham, and Ryan M Eustice. Cooperative localization by factor composition over a faulty low-bandwidth communication channel. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 401–408. IEEE, 2015.

[75] Matthew R Walter, Ryan M Eustice, and John J Leonard. Exactly sparse extended information filters for feature-based SLAM. *The International Journal of Robotics Research*, 26(4):335–359, 2007.

[76] Sarah E Webster, Jeffrey M Walls, Louis L Whitcomb, and Ryan M Eustice. Decentralized extended information filter for single-beacon cooperative acoustic navigation: Theory and experiments. *Robotics, IEEE Transactions on*, 29(4): 957–974, 2013.

[77] Stefan B Williams, Gamini Dissanayake, and Hugh Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 1, pages 406–411. IEEE, 2002.

[78] Stefan B Williams, Gamini Dissanayake, and Hugh Durrant-Whyte. Towards multi-vehicle simultaneous localisation and mapping. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 3, pages 2743–2748. IEEE, 2002.

Appendix A

Extended Data Graphs

Extra graphs from the main text where their presence there would overcrowd the text without significant gains from their inclusion there. They are included here for completeness and to enable the reader to better verify that the conclusions reached are not specific to the examples shown and were presented more generally.

A.1 Compression and Packetisation

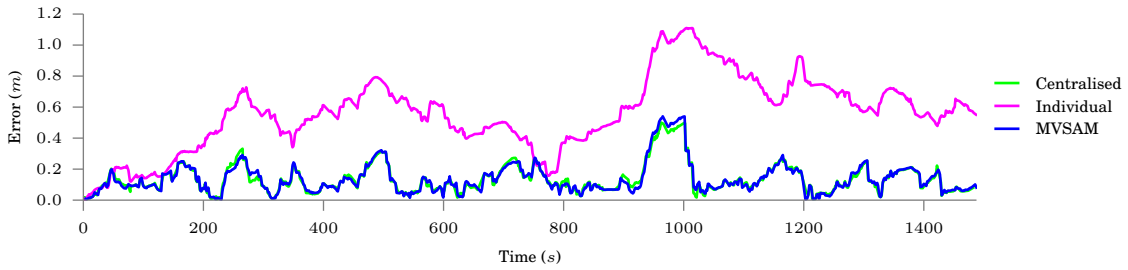
A.1.1 UTIAS Examples

Including all 9 datasets for completeness. Figure A.1 covers the tracking scenario and Figure A.2 covers the GPS denied scenario.

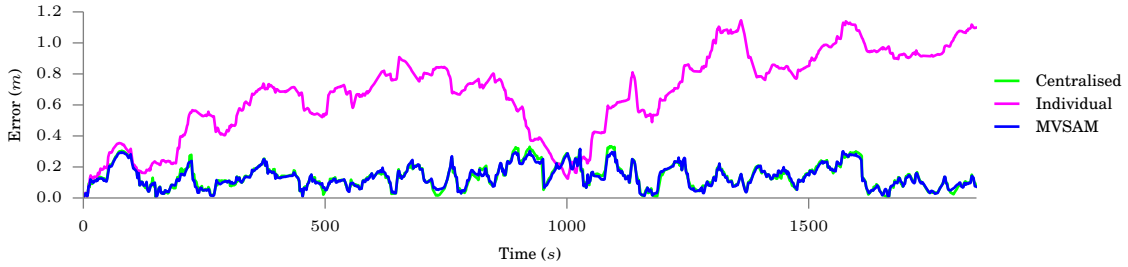
A.2 Out of Sequence Measurements

A.2.1 UTIAS Examples

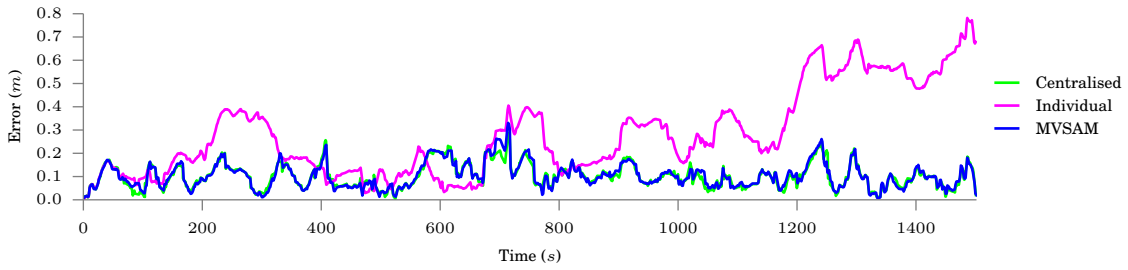
RMS error for a vehicle from the first 7 datasets is in Figure A.3.



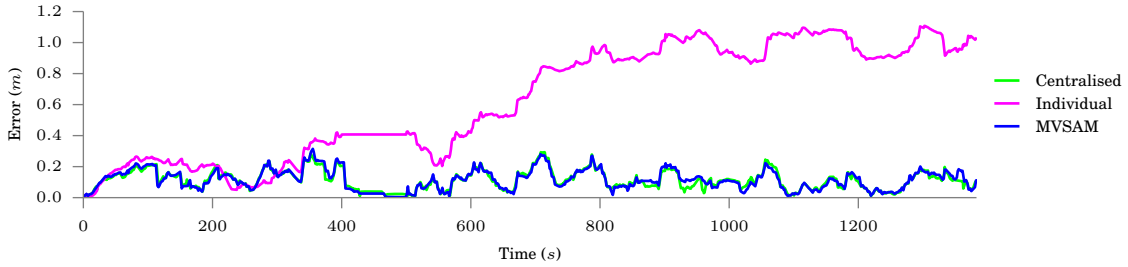
(a) Online estimator position error for UTIAS dataset 1.



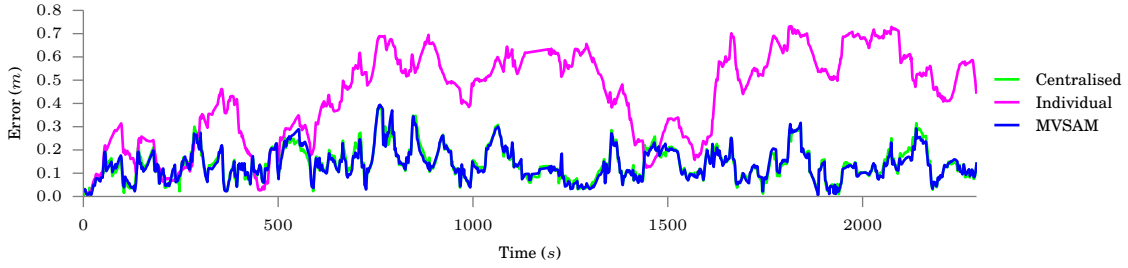
(b) Online estimator position error for UTIAS dataset 2.



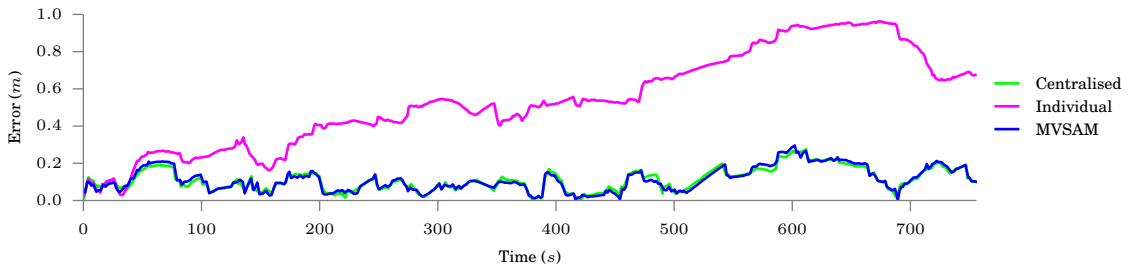
(c) Online estimator position error for UTIAS dataset 3.



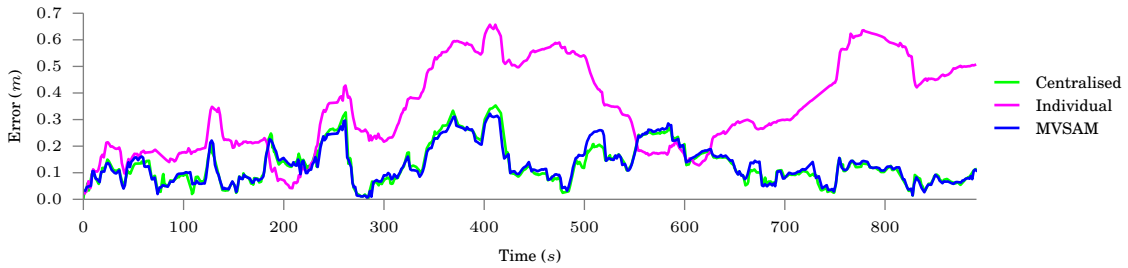
(d) Online estimator position error for UTIAS dataset 4.



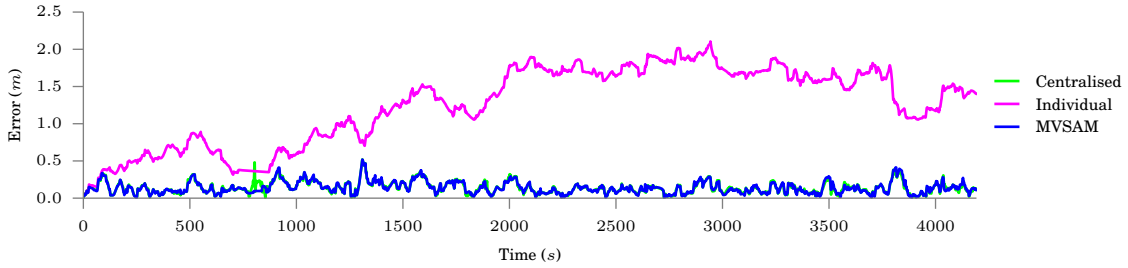
(e) Online estimator position error for UTIAS dataset 5.



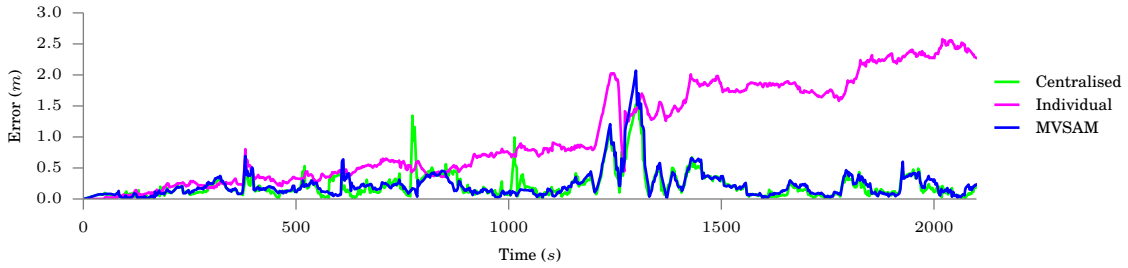
(f) Online estimator position error for UTIAS dataset 6.



(g) Online estimator position error for UTIAS dataset 7.

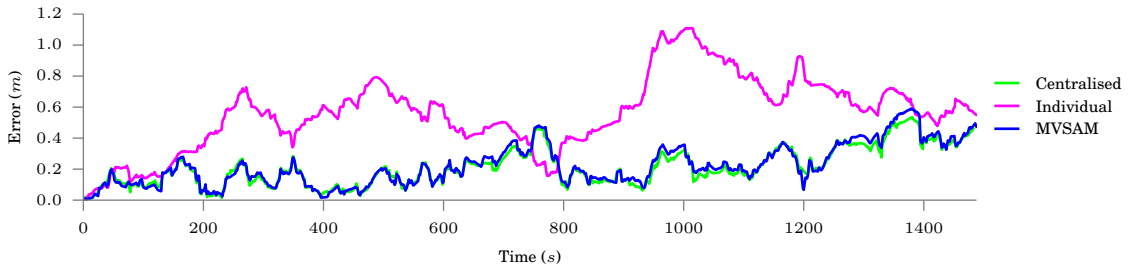


(h) Online estimator position error for UTIAS dataset 8.

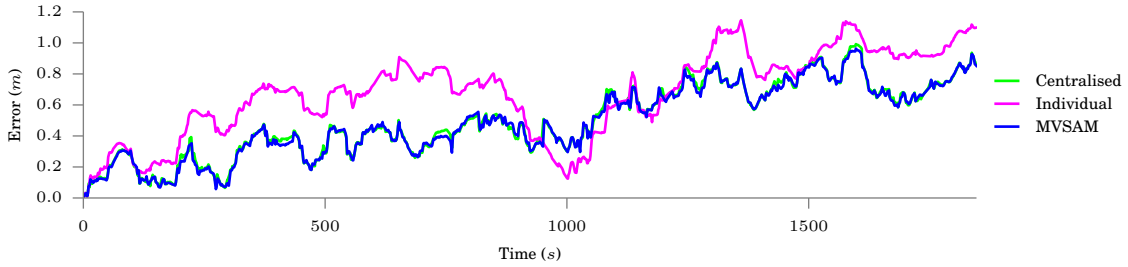


(i) Online estimator position error for UTIAS dataset 9.

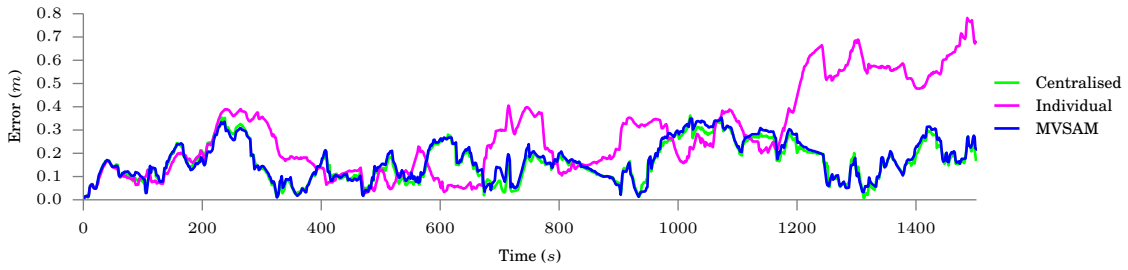
Figure A.1 – Position error on one vehicle from each dataset using the tracking setup.



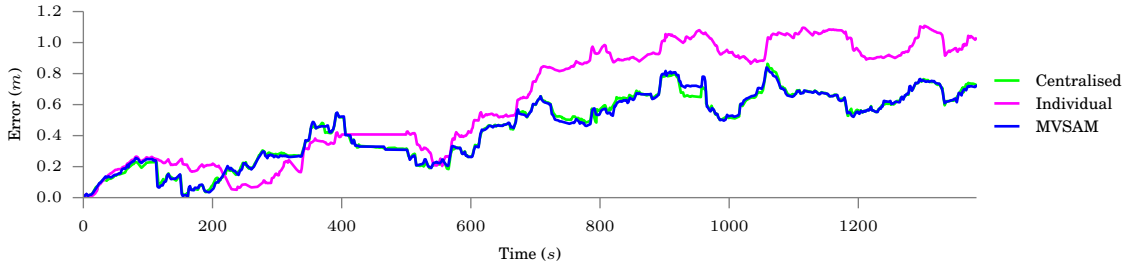
(a) Online estimator position error for UTIAS dataset 1.



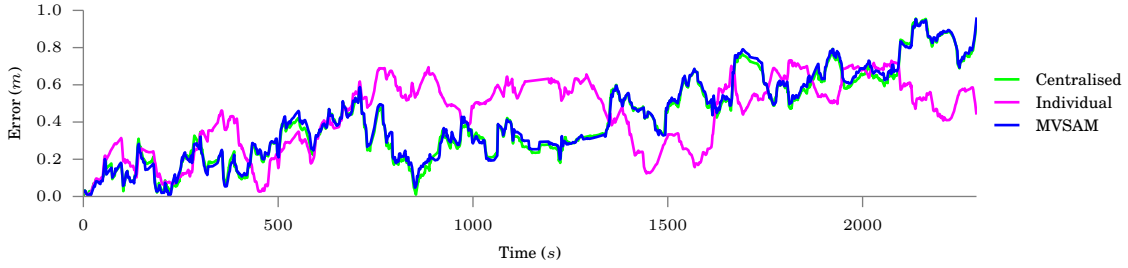
(b) Online estimator position error for UTIAS dataset 2.



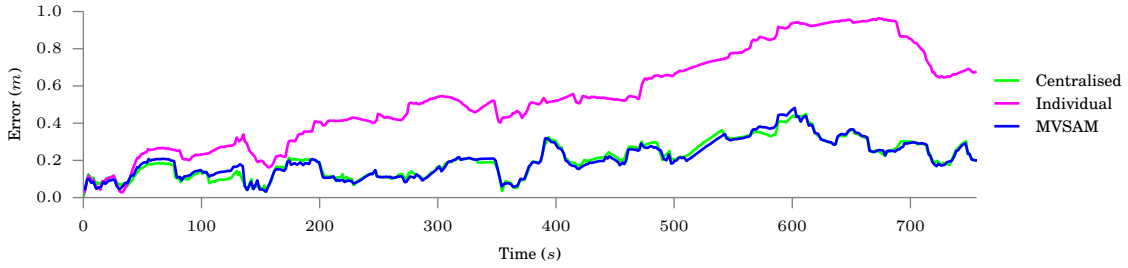
(c) Online estimator position error for UTIAS dataset 3.



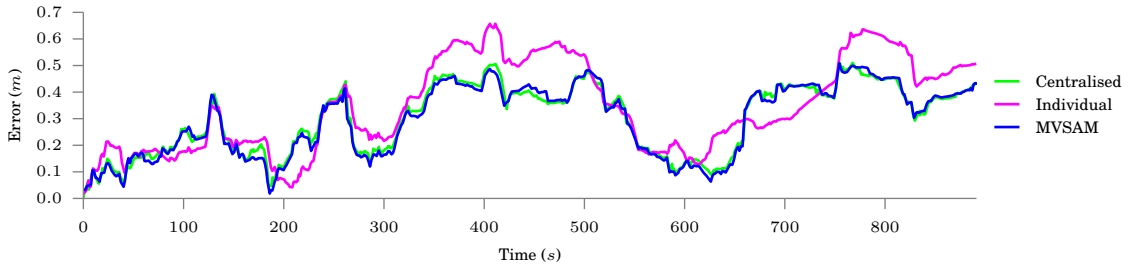
(d) Online estimator position error for UTIAS dataset 4.



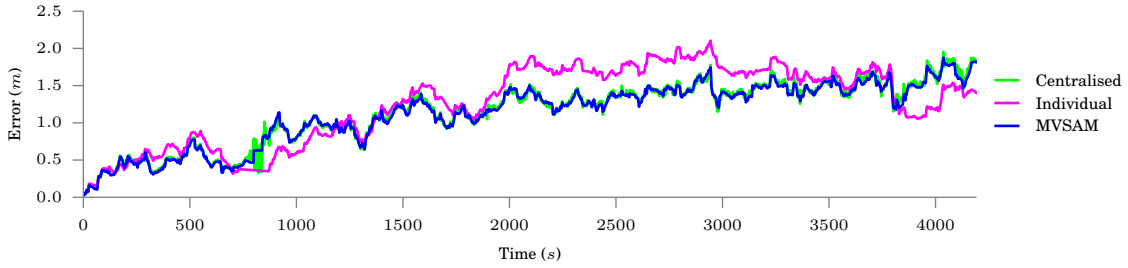
(e) Online estimator position error for UTIAS dataset 5.



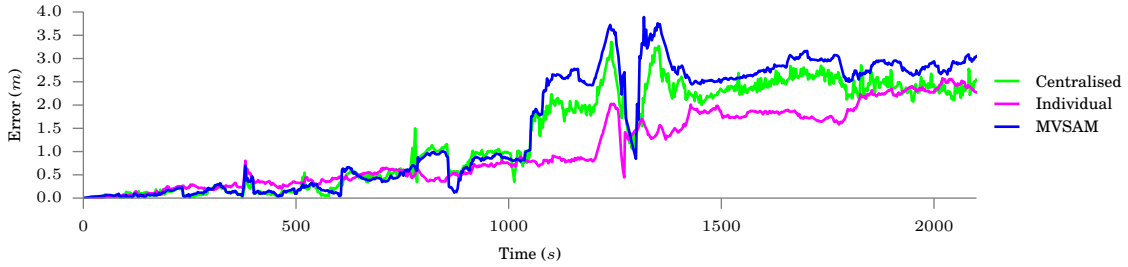
(f) Online estimator position error for UTIAS dataset 6.



(g) Online estimator position error for UTIAS dataset 7.

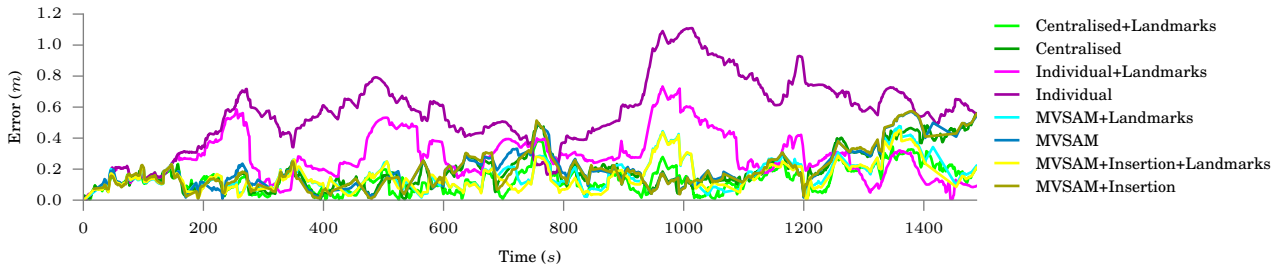


(h) Online estimator position error for UTIAS dataset 8.

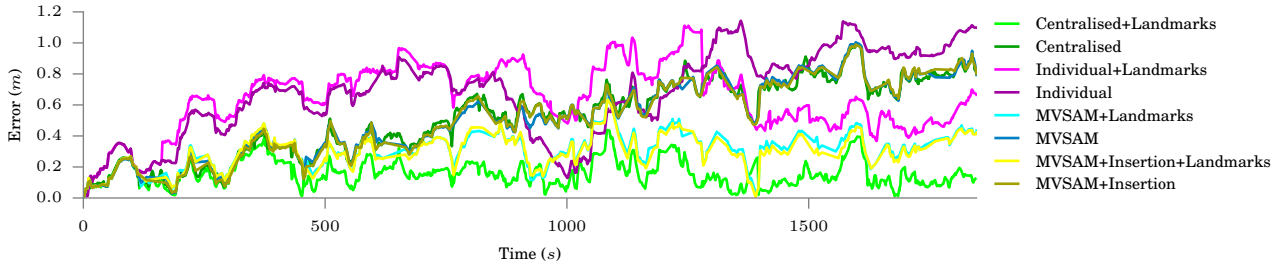


(i) Online estimator position error for UTIAS dataset 9.

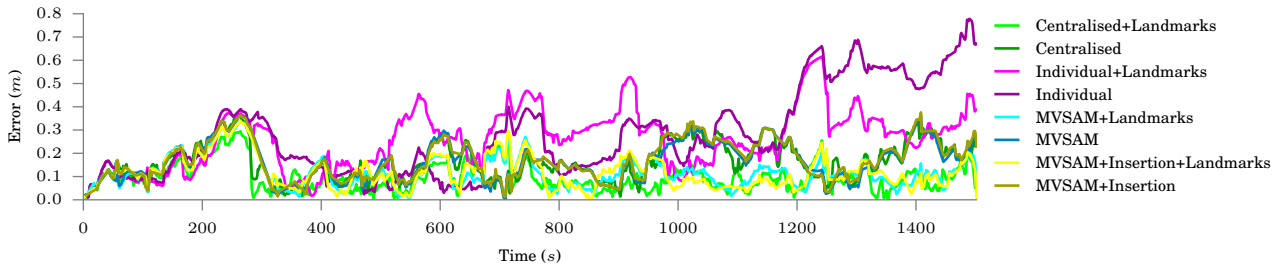
Figure A.2 – Position error on one vehicle from each dataset in a GPS denied environment. The isolated/individual solutions match those in Figure A.1 as the on board sensing/setup is identical.



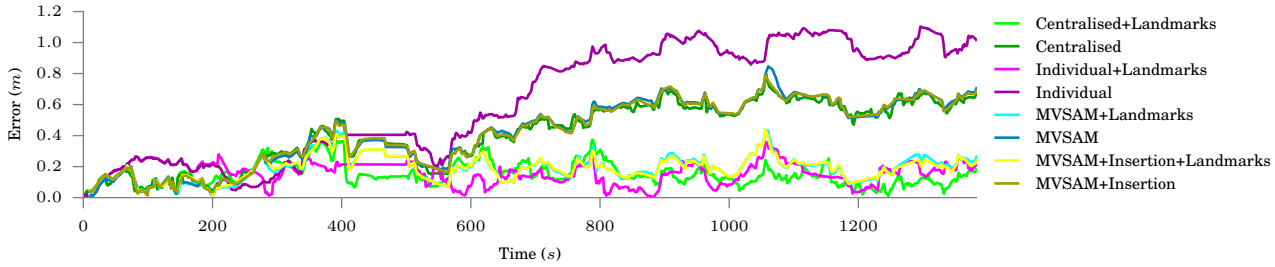
(a) Online estimator position error for UTIAS dataset 1.



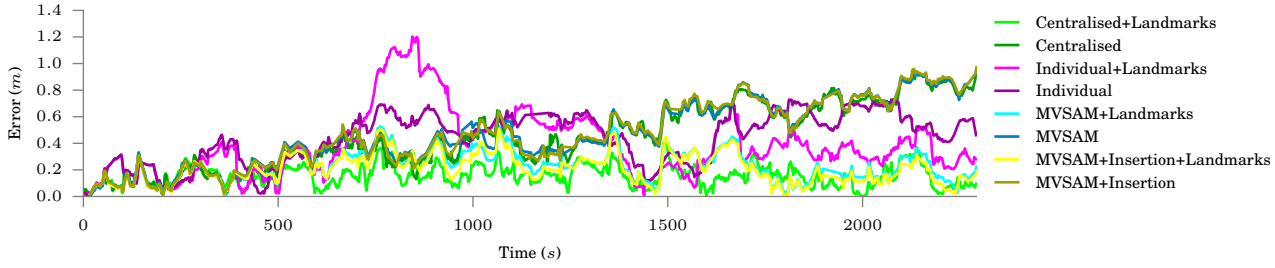
(b) Online estimator position error for UTIAS dataset 2.



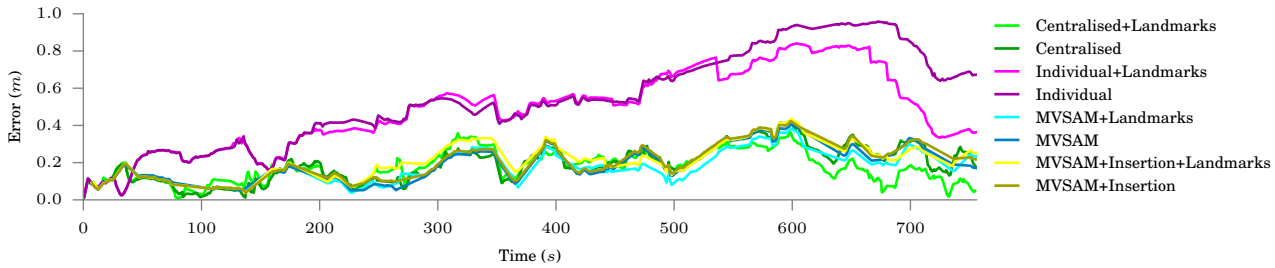
(c) Online estimator position error for UTIAS dataset 3.



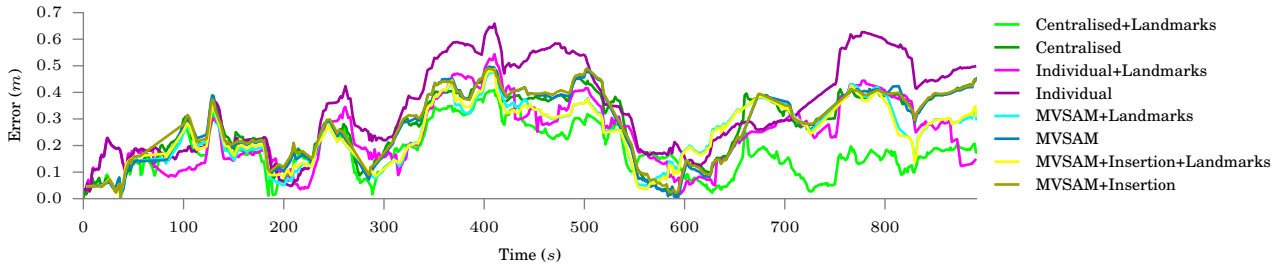
(d) Online estimator position error for UTIAS dataset 4.



(e) Online estimator position error for UTIAS dataset 5.



(f) Online estimator position error for UTIAS dataset 6.



(g) Online estimator position error for UTIAS dataset 7.

Figure A.3 – Position error on one vehicle from each dataset in a GPS denied environment with communication delay and loop closures. Optimisers with landmarks outperform those without. The decentralised optimisers without loop closures perform similarly to the centralised but the centralised solver with loop closures outperforms the decentralised optimisers in a number of cases with a visible difference.