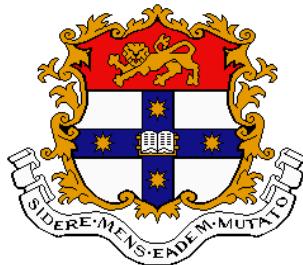


Inertial SLAM with Delayed Initialisation

Todd Lupton

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



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

March 2010

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.

Todd Lupton

March 31, 2010

Abstract

Todd Lupton
The University of Sydney

Doctor of Philosophy
March 2010

Inertial SLAM with Delayed Initialisation

This thesis focuses on the development and implementation of a new method for using inertial measurement unit (IMU) observations along with observations from body frame sensors, such as cameras, to perform simultaneous localisation and mapping (SLAM).

The inertial observation integration equations are re-parameterised into a fixed body frame coordinate system where the components dependent on initial conditions are isolated. Inertial observations can then be integrated before initial velocity or attitude of the platform are known or the effect of gravity is compensated for.

The resultant algorithm allows inertial information to be integrated and used before the initial conditions of the platform are known and the initial condition are recovered automatically in a linear way. Delayed initialisation of the inertial solution means the vision-inertial SLAM implementation quickly self initialises and can be used for applications where specialised initialisation routines are not practical or possible.

The ability for inertial observations to be used to estimate the scale of a map constructed through bearing only SLAM is also investigated. An analysis of the observability of map scale in this situation as well as methods of implementation are explored.

Analysis of the performance of the developed algorithms through statistical simulation tests and real world experimental results are presented as well as comparisons to the performance of traditional inertial SLAM algorithms.

Acknowledgements

This thesis would not have been possible without the help and support of many people.

I would like to thank my supervisor Prof. Salah Sukkarieh for his guidance and support over the last four years. Thanks also go to Esa for his constant help with hardware development and maintenance of the sensor suite used for experiments.

Thanks to Mitch and Michael for help in the early days when I was still trying to get my head around inertial SLAM. And also to Lash for insightful discussions on inertial implementations and help in talking through my ideas in the later stages.

I would also like to thank everyone at the ACFR for helping to contribute to a stimulating and entertaining working environment with a special thanks to Sisir, Tim and Calvin for helping to keep me sane through constant debate.

Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
Contents	iv
List of Figures	vii
List of Tables	xv
1 Introduction	1
1.1 Motivating problems	1
1.2 Objectives of this thesis	2
1.3 Main contributions of this thesis	3
1.4 Thesis structure	3
2 Inertial navigation	6
2.1 Underlying science	6
2.1.1 Inertial navigation development	7
2.2 Inertial measurement units	7
2.2.1 Inertial reference frames	8
2.2.2 Reference frame selection	9
2.3 Inertial observation models	10
2.3.1 Simplified strap-down inertial modelling	11
2.4 Aiding inertial navigation	12
2.4.1 Sources of inertial aiding	13
2.4.2 How is inertial aiding implemented?	13
2.5 Initial alignment	13
2.5.1 Removing initialisation stage requirements	15
2.6 Pre-integration of inertial observations	15

3 Visual SLAM	17
3.1 Visual SLAM	19
3.1.1 Stereo implementations	20
3.1.2 Monocular implementations	21
3.1.3 Structure from motion	23
3.2 Vision aided inertial	25
3.3 Graphical SLAM	26
3.4 Outcomes of the literature review	28
4 Inertial Initialisation and Pre-Integration	30
4.1 Inertial aiding and estimation	30
4.1.1 Importance of linearity	30
4.1.2 Where does the non-linearity come from?	32
4.1.3 Importance of initial conditions	33
4.1.4 How are initial conditions obtained?	34
4.2 Achieving linearity	35
4.2.1 Re-parameterisation to determine initial conditions	37
4.2.2 Removing initial condition requirements	38
4.3 Inertial pre-integration	40
4.3.1 Inertial delta components	41
4.3.2 Inertial delta observation creation	43
4.3.3 Inertial delta Jacobian and covariance creation	45
4.3.4 Inertial delta observation implementation	47
4.4 Importance of the Δp^+ component	49
4.5 Chapter summary	54
5 Analysis of pre-integrated inertial delta observations	56
5.1 Initial condition recovery	56
5.1.1 Initial velocity estimation	56
5.1.2 Gravity vector estimation	57
5.1.3 Monte-Carlo simulation for initial conditions	59
5.2 Remaining non-linearity	64
5.2.1 Non-linearity from bias correction	64
5.2.2 Monte-Carlo simulation of non-linearity from bias correction	65
5.2.3 Non-linearity from inertial drift	67
5.3 Comparison to standard inertial integration	68
5.3.1 Inertial only navigation	68
5.3.2 Aided inertial SLAM	71
5.3.3 Vision aided inertial initialisation	72
5.4 Chapter summary	80

6 Scale estimation	81
6.1 The scale factor problem	81
6.2 Scale estimation	82
6.2.1 The scale equation	82
6.2.2 Scale observability	85
6.2.3 Scale uncertainty	86
6.3 Extending beyond 3 poses	87
6.3.1 Scale estimate correlation	89
6.4 One dimensional scale observability analysis	91
6.5 Extending to the 6DoF case	94
6.6 Scale parameterisation	96
6.6.1 Three dimensional explicit scale estimation	97
6.6.2 Implicit scale parameterisation	99
6.6.3 Implicit scale analysis	102
6.7 Chapter summary	103
7 Practical Techniques	104
7.1 Observation edge energy	104
7.1.1 Pre-integrated inertial delta observation edge energy	105
7.1.2 Observation edge energy distribution	107
7.2 Feature extraction and matching	110
7.2.1 Using pose estimates for data association	111
7.2.2 Fundamental matrix outlier rejection	111
7.2.3 Incorporating gyro observations into fundamental matrix estimation	113
7.2.4 Using observation edge energies to remove outliers	115
7.3 Pose initialisation options	116
7.3.1 Prediction from previous filter estimates	116
7.3.2 Initialisation of a new pose from image observations	117
7.4 Optimising information smoothers for long trajectories	119
7.4.1 Marginalisation	120
7.4.2 Conditioning	121
7.4.3 Sliding window forced independence	122
7.5 Chapter summary	124
8 Experiments	126
8.1 Experimental set-up	126
8.1.1 Sensors	126
8.1.2 SLAM implementation	127
8.1.3 Experimental location	130
8.2 Initial testing	131
8.3 Second test	135

8.4	First run	140
8.5	Second run	149
8.6	Monocular scale estimation	154
8.7	Chapter summary	159
9	Conclusions	161
9.1	Summary of contributions	161
9.1.1	Development of pre-integrated inertial delta observations .	161
9.1.2	Analysis of pre-integrated inertial delta observations . . .	162
9.1.3	Visual map scale estimation using inertial observations .	162
9.1.4	Inclusion of gyro information in fundamental matrix calcu- lations	163
9.1.5	Sliding window forced independence	163
9.2	Future work	164
9.2.1	Yaw estimation using pre-integrated inertial delta observa- tions	164
9.2.2	Loop closure for forced independence smoothing	165
9.2.3	More stable monocular scale estimation	166
	Bibliography	167
A	Simulation setup	174
A.1	Simulation model	174
A.2	Simulated trajectory	175
B	Fundamental matrix calculation with known rotation	177
B.1	Three point fundamental matrix algorithm with known rotation .	178
B.2	Normalised 3 point fundamental matrix algorithm with known ro- tation	181

List of Figures

2.1	Different inertial navigation frames used for integrating inertial observations. The earth inertial frame is fixed to the centre of the earth but does not rotate with it, while the ECEF frame rotates with the earth so a point on the surface of the earth has a fixed location in the ECEF frame. The local level frame is fixed to a point in the area of operation of the vehicle and aligned with gravity vector pointing down. The body frame is fixed to the body of the vehicle.	9
3.1	Representation of the SLAM problem.	18
3.2	Representation of the Graphical SLAM formulation. Notice the similarity to the representation of the SLAM problem in figure 3.1.	27
4.1	Illustration of global frame navigation where the inertial observations are transformed into the global frame for integration.	36
4.2	Illustration of body frame navigation where the inertial observations are transformed into the body frame of the first pose for integration.	37
4.3	Flowcharts comparing the steps in standard inertial integration to those of the re-parameterised form.	39
4.4	Acceleration profiles with respect to time for the four cases considered when analysing the effect of the Δp^+ component when compared to constant velocity and constant acceleration methods.	52
4.5	Position profiles with respect to time for the four cases considered when analysing the effect of the Δp^+ component when compared to constant velocity and constant acceleration methods.	53
5.1	Sample estimate convergence over the first 5 poses for the initial velocity estimate. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.	59

5.2	Sample estimate convergence over the first 5 poses for the gravity vector estimate. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.	60
5.3	Monte Carlo results for the initial velocity estimates after the first 5 poses over 1000 runs. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.	61
5.4	Monte Carlo results for the gravity vector estimates after the first 5 poses over 1000 runs. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.	62
5.5	Difference in estimated and true position after navigating using pre-integrated inertial delta observations without bias compensation for a period of 100 seconds.	66
5.6	Difference in estimated and true attitude after navigating using pre-integrated inertial delta observations without bias compensation for a period of 100 seconds.	66
5.7	Difference in estimated and true position after navigating using pre-integrated inertial delta observations corrected for biases after pre-integration is performed for a period of 100 seconds.	67
5.8	Difference in estimated and true attitude after navigating using pre-integrated inertial delta observations corrected for biases after pre-integration is performed for a period of 100 seconds.	67
5.9	Gravity vector estimate convergence using the pre-integrated inertial delta observation technique over the first 20 poses.	73
5.10	Initial velocity estimate convergence using the pre-integrated inertial delta observation technique over the first 20 poses.	74
5.11	Initial roll estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors. . .	75
5.12	Initial pitch estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors. .	75
5.13	Initial x velocity estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.	76
5.14	Initial y velocity estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.	76
5.15	Initial z velocity estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.	77
5.16	Initial roll estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.	78

5.17 Initial pitch estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.	78
5.18 Initial x velocity estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.	79
5.19 Initial y velocity estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.	79
5.20 Initial z velocity estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.	79
 6.1 Sample problem to investigate map scale observability with bearing only observations and an IMU. In this case the platform moves in one dimension only with accelerometer observations along this dimension used to construct pre-integrated inertial delta observations.	83
6.2 Sample problem with 4 poses to show which pose estimates and inertial observations are used to construct the 2 different estimates of the scale factor, α_1 and α_2 . In this case the platform moves in one dimension only with accelerometer observations along this dimension used to construct pre-integrated inertial delta observations.	88
6.3 True simulated position and estimated position using bearing only SLAM and an arbitrary distance on 1 metre between the first two poses. Inertial observations are not used in the SLAM estimate.	92
6.4 Comparison of the estimated scale factor from equation 6.7 to the true scale factor from equation 6.18 for the simulated trajectory shown in figure 6.3.	93
6.5 Plots of the numerator and denominator components of the scale equation shown in figure 6.4	94
6.6 True simulated position and estimated position using bearing only SLAM and an arbitrary distance on 1 metre between the first two poses for the 6DoF case. Gyro observations are used in the bearing only SLAM estimate but accelerometer observations are not.	97
6.7 True simulated attitude and estimated attitude using bearing only SLAM and an arbitrary distance on 1 metre between the first two poses for the 6DoF case. Gyro observations are used in the bearing only SLAM estimate, leading to the high accuracy of the estimate shown, but accelerometer observations are not.	98
6.8 Estimated scale factor (α) for each direction in the 6DoF simulation with the estimated 2σ uncertainty bounds.	99

6.9	True simulated position and estimated position using bearing only SLAM and inertial observations via the implicit scale estimation method for the 6DoF case.	100
6.10	Estimated scale factor α for the three directions with associated 2σ uncertainty bounds for the 6DoF trajectory with implicit scale estimation. Notice how the true scale of 1 is within the uncertainty bounds for the whole trajectory.	101
7.1	Graphical structure of pre-integrated inertial delta observations showing the graph edges connecting the observation nodes with the estimated vehicle state nodes. The gravity and bias states are common to all poses.	106
7.2	Normalised observation edge energy for the delta attitude observations using the estimated poses. Note how the energies are very small, this is because the rotation information provided by the gyro observations is much more accurate than that provided by the visual observations so the estimated attitude is mostly derived from the gyro observations.	108
7.3	Normalised observation edge energy for the delta attitude observations using the true poses. Note how this distribution is much more consistent with what would be expected for innovations resembling zero mean white noise than what is shown in figure 7.2.	109
7.4	Two consecutive images showing matched features and pruning of outliers using RANSAC fundamental matrix estimation.	112
7.5	Two consecutive images showing matched features and pruning of outliers using RANSAC fundamental matrix estimation with gyro observations.	115
7.6	Joint probability distribution of a trajectory and map. The states in part A and B of the map are estimated together using observations from both part A and B of the map.	120
7.7	Marginal probability distribution of a trajectory and map. The states in part B of the map are estimated using the observations from part B as well as the estimated states from part A with their associated uncertainties.	121
7.8	Conditional probability distribution of a trajectory and map. The states in part B of the map are estimated using the observations from part B assuming the estimated states in part A are known with complete certainty.	122
7.9	Forced independence probability distribution of a trajectory and map. The states in part B of the map are estimated using the observations from part B completely independent of the observations or estimates from part A.	123

8.1	Sensor suite used to obtain the walking datasets. The Point Grey Research Bumblebee2 stereo camera can be seen on the top of the unit with the Honeywell HG1900 IMU mounted just below it.	127
8.2	Satellite image from Google Maps showing the buildings in the area that the datasets were collected. The building in the middle with the green roof is the Rose St. building and the white building just north of it is the Link building.	129
8.3	Example image of the environment that the datasets were taken in. This image is taken from the dataset shown in figure 8.26 and is of the downstairs area inside the Rose St. building.	130
8.4	Example image of the environment that the datasets were taken in. This image is taken from the dataset shown in figure 8.26 and is of the outside area between the Link building and the Rose St. building.	131
8.5	Estimated trajectory of the sensor platform when moving back and forth between two marked locations on the ground that are 9m apart.	132
8.6	Components of the estimated position for the trajectory shown in figure 8.5. The north, east and down components are in the body frame of the first pose.	133
8.7	Components of the estimated attitude for the trajectory shown in figure 8.5. The roll, pitch and yaw components are in the body frame of the first pose.	133
8.8	Components of the estimated velocity for the trajectory shown in figure 8.5. The north, east and down components are in the body frame of the first pose.	133
8.9	Estimated trajectory of the sensor platform when moving back and forth between two marked locations on the ground that are 9m apart using only inertial observations for comparison to figure 8.5. The initial conditions for the IMU were estimated while stationary but no visual observations were used to aid the inertial after it has started moving.	135
8.10	Estimated trajectory of the sensor platform when moving between the three marked locations in a triangular pattern and the section of 30cm intervals. The section with the 30cm intervals can be seen on the extreme left side of the figure.	136
8.11	Components of the estimated position for the trajectory shown in figure 8.10. The north, east and down components are in the body frame of the first pose.	136
8.12	Components of the estimated attitude for the trajectory shown in figure 8.10. The roll, pitch and yaw components are in the body frame of the first pose.	137

8.13 Components of the estimated velocity for the trajectory shown in figure 8.10. The north, east and down components are in the body frame of the first pose.	137
8.14 Close up showing the east component of the position during the incremental movements between the 30cm intervals from the trajectory shown in figure 8.10.	139
8.15 Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building overlaid onto a satellite image of the area. The trajectory does not line up exactly partly due to the viewpoint of the image not being taken from directly above. . .	141
8.16 Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The trajectory consists of 720 image pairs taken over 115 seconds and is approximately 120 metres in length.	142
8.17 Side view of the estimated trajectory of the first dataset showing the two levels of the Rose St. building. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The back set of stairs can be clearly seen on the right side of the image as well as the misalignment of the two portions of the trajectory on the bottom level.	143
8.18 Estimated position of the platform for each image in the first Rose St. building dataset. The north, east and down components are in the body frame of the first pose.	143
8.19 Estimated velocity of the platform for each image in the first Rose St. building dataset. The north, east and down components are in the body frame of the first pose. Notice how the down component of velocity clearly shows the individual steps taken even though they can not be seen in the position estimates from figure 8.18. . .	144
8.20 Estimated attitude of the platform for each image in the first Rose St. building dataset. The roll, pitch and yaw components are in the body frame of the first pose.	144
8.21 Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building without using edge energy outlier detection. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. Notice the errors in the estimated pose within the front staircase in the bottom right of the figure.	146

8.22 Estimated velocity of the platform for each image in the first Rose St. building dataset. The north, east and down components are in the body frame of the first pose. Notice the spikes in estimated velocity around the 176th and 236th pose caused by incorrect data association within the front staircase.	147
8.23 Number of image features observed for each image in the first Rose St. building dataset. The red line is after the fundamental matrix constraint has been applied to remove outliers and the green line is after observation edge energies have been used as well. The blue line is at 8 features per image which is the minimum number of correctly associated features required for the fundamental matrix technique to work. Notice how the number of observed features drops around the 200th and 550th image corresponding to the front and back staircases respectively where there are very few visual features to observe. This is where fundamental matrix pruning of outliers performs worst.	147
8.24 Sample image inside the front staircase from the first dataset. There are very few available visual features from this image except for corners of the stairs which are indistinguishable from each other and therefore can not be reliably tracked from frame to frame.	149
8.25 Top view of the estimated trajectory of the second dataset between the Rose St. building and the Link building overlaid onto a satellite image of the area. The trajectory does not line up exactly partly due to the viewpoint of the image not being taken from directly above and the drift in the solution towards the end of the trajectory. The bridge between the two buildings is not present in this photo.	150
8.26 Top view of the estimated trajectory of the second dataset between the Rose St. building and the Link building. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The trajectory consists of 1130 image pairs taken over 180 seconds and is approximately 180 metres in length.	151
8.27 Side view of the estimated trajectory of the second dataset showing the two levels of the Rose St. building and the staircase down the back of the Link building. The red cross at (0,0) is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The Link building stair case can be clearly seen on the right side of the figure.	151
8.28 Estimated position of the platform for each image in the second dataset. The north, east and down components are in the body frame of the first pose.	152

8.29	Estimated velocity of the platform for each image in the second dataset. The north, east and down components are in the body frame of the first pose. Notice how the down component of velocity clearly shows the individual steps taken even though they can not be seen in the position estimates from figure 8.28.	152
8.30	Estimated attitude of the platform for each image in the second dataset. The roll, pitch and yaw components are in the body frame of the first pose.	152
8.31	Number of image features observed for each image in the second dataset.	153
8.32	Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building processed with only inertial and monocular observations to assess scale observability. This is the same dataset as used in section 8.4 and figure 8.16 for comparison.	155
8.33	Estimated velocity for the monocular result shown in figure 8.32. The estimated velocity for the stereo case on the same dataset can be seen in figure 8.19.	157
8.34	Estimated distance between consecutive poses for the monocular and stereo result from the first dataset over the two levels of the Rose St. building.	158
8.35	Ratio of the estimated distance between consecutive poses for the monocular and stereo result from the first dataset over the two levels of the Rose St. building.	158
A.1	Trajectory used for simulation results. Red line points in the vehicle frame up direction, blue lines point in the vehicle frame forward direction for each pose that an image is taken at. The black crosses are the randomly distributed visual landmarks.	176

List of Tables

4.1	Derivatives of the inertial prediction equations with respect to estimated states	33
4.2	Derivatives of the inertial delta observation prediction equations with respect to estimated states	50
4.3	Position estimation models considered for comparison	51
4.4	Summary of the four acceleration cases considered	51
4.5	Final position results of the four acceleration cases considered using the three estimation models shown in table 4.3. True final position value is 0.25m for every case	54
5.1	Initial condition results from Monte Carlo runs on first 5 poses. Mean estimate and variance of means calculated from the data presented in figures 5.3 and 5.4	63
5.2	Filter estimates of the variance of the initial velocity and gravity vector estimates after 5 image poses	63
5.3	Mean and variance of the results of 1000 Monte Carlo runs for the pre-integrated inertial delta technique	69
5.4	Mean and variance of the results of 1000 Monte Carlo runs for the standard inertial integration technique	69
5.5	Analysis of the Monte-Carlo results for the pre-integrated inertial delta technique shown in table 5.3	70
5.6	Mean and variance of the results of 1000 Monte Carlo runs for the pre-integrated inertial delta technique using range and bearing landmark observations	72
5.7	Mean and variance of the results of 1000 Monte Carlo runs for the standard inertial technique using range and bearing landmark observations	72
6.1	Derivatives of the scale equation 6.7	86
6.2	Derivatives of the two scale equations derived from observations from 4 consecutive poses	89

6.3	Extreme values in the north, east and down components from 100 Monte-Carlo simulations of the first 20 poses for implicit scale estimation from the trajectory shown in figure 6.9	102
8.1	HG1900 IMU measured specifications	127
8.2	Point Grey Research Bumblebee2 with 2.1mm lenses specifications	128
8.3	Estimated position, velocity and attitude of the sensor platform during the stationary periods at the marked locations in the dataset shown in figure 8.5	134
8.4	Estimated position, velocity and attitude of the sensor platform during the stationary periods at the first marked location in the dataset shown in figure 8.10	138
8.5	Estimated position, velocity and attitude of the sensor platform during the stationary periods at the second marked location in the dataset shown in figure 8.10	138
8.6	Estimated position, velocity and attitude of the sensor platform during the stationary periods at the third marked location in the dataset shown in figure 8.10	139
8.7	Estimated east position of the sensor platform during the stationary periods for the 30cm intervals in the dataset shown in figure 8.10. The estimated and true offset from the first location in this section is also shown	140
8.8	First and last pose estimates for the trajectory in the first Rose St. building dataset	145
8.9	Estimated gravity vector and magnitude for the first and last pose in the first Rose St. building dataset	145
8.10	First and last pose estimates for the trajectory in the second dataset through the Rose St. building and the Link building	153
8.11	Estimated gravity vector and magnitude for the first and last pose in the second dataset through the Rose St. building and the Link building	154
8.12	First and last pose estimates for the trajectory in the first Rose St. building dataset using monocular observations	156
8.13	Estimated gravity vector and magnitude for the first and last pose in the first Rose St. building dataset using monocular observations	157

Chapter 1

Introduction

1.1 Motivating problems

The ability to keep track of the location of a high dynamic 6 degree of freedom (DoF) platform moving in complex environments, such as in and around buildings, is useful for a number of applications.

Urban search and rescue is one of these applications. When emergency response workers enter an unknown building, knowing accurately the location and trajectory of each team members can help to insure their safety as well as to coordinate efforts and ensure the whole building is searched.

Another useful application is for micro unmanned aerial vehicle (UAV) systems where accurate estimates of the velocity and attitude of the vehicle is required for stability and control.

Operating in and around buildings presents particular difficulties for localisation systems. The use of GPS is usually not possible in these environments as the satellites are not visible inside the buildings, and even outside, shadowing of the satellites by surrounding buildings and multi-path reflections make the observations inaccurate. The accuracy of GPS is also usually not sufficient compared to the range to obstacles encountered in such close environments.

Visual SLAM is a possible solution to these types of problems however suffi-

cient visual observations may not always be guaranteed, especially when operating close to blank walls or in dark environments such as during blackouts in rescue situations. The localisation method must also be able to handle high dynamics and be very robust as the rescue workers should not be expected to change their operating behaviour to suite the localisation system.

If an inertial measurement unit (IMU) is combined with visual SLAM then the inertial observations can help to constrain the location estimate during times of low visual observability as well as handle the high dynamics of the system.

The problem with inertial measurement units is that they usually require dedicated initialisation stages where the position, velocity and attitude of the platform needs to be known to a high degree of accuracy in a global navigation frame before the inertial observations can be used.

The requirement for an initialisation stage can cause an inconvenience and may not even be possible in the applications considered. The requirement for accurate velocity and attitude estimates to be maintained can also cause problems as this may not be possible with the available observations.

1.2 Objectives of this thesis

If there was a way to use the inertial observations to aid in visual SLAM without having to know the initial conditions of the system, or maintain an accurate pose estimate throughout the trajectory it would be very useful for the considered applications. A development of a method to achieve this is the objective of this thesis.

The developed system should also be able to gracefully restart its estimation at any point in time, without known pose conditions, allowing for fast recovery from any errors in the navigation filter without requiring user intervention.

Ideally this system should be able to operate with range and bearing observations of landmarks as well as in situations where for practical reasons only bearing observations are available.

1.3 Main contributions of this thesis

In order for these objectives to be met, the following main issues need to be addressed:

1. Develop a method of using inertial observation in a navigation solution when initial state estimates, particularly when initial velocity and attitude, are not available.
2. Allow initial state estimates to be recovered in a linear way once sufficient observations have been made to observe them so that absolute velocity and attitude estimates can be provided.
3. Identify the requirements and techniques to obtain a map scale estimate from bearing only landmark and inertial observations so this information can be fused when range and bearing observations are not available.

These are the main contributions of this thesis.

1.4 Thesis structure

Chapter 2 provides an overview of inertial navigation, specifically focusing on techniques relevant to the work presented in this thesis. The background of strap-down inertial measurement units is given as well as the choices of various reference frames that are commonly used to perform inertial navigation with them. A basic model for strap-down inertial navigation commonly used in SLAM is given and possible ways to correct the inertial solution using other sensors is discussed. The problem of initialisation of inertial navigation solutions and the initial conditions required is mentioned as well.

Chapter 3 contains a literature review of recent visual SLAM research in areas that are relevant to the applications considered in this thesis. A background on the use of visual SLAM to aid inertial navigation is also presented along with the advantages that this combination of sensors provide. Finally the outcomes

arrived at from conducting this literature review and their implications to the applications considered in this thesis are discussed.

Chapter 4 contains the theoretical background and development of the pre-integrated inertial delta observation algorithm that is the core contribution of this thesis. The reasons behind the initial condition requirements for inertial navigation as well as the sources of non-linearity in the integration equations is investigated. From this investigation a method for re-parameterising the inertial navigation equations into a fixed body frame of reference and a way of removing the requirement for the initial conditions of the platform to be known until after integration of the inertial observations is developed.

The algorithms required to implement this form of inertial navigation are presented in pseudo-code form and compared to the standard inertial navigation algorithms commonly used in inertial SLAM.

The Δp^+ component in pre-integrated inertial delta observations, which is not present in standard techniques for inertial navigation, is discussed. Its role in linking the average velocity observed from other sensors, such as cameras, to the instantaneous velocity observed by the IMU is presented. Simulations are conducted to analyse the effect it has under various acceleration profiles compared to other common process models used in SLAM.

Chapter 5 conducts an analysis of the pre-integrated inertial delta observation technique developed in chapter 4. The equations are inspected to identify how they allow the initial conditions, namely initial velocity and the gravity vector in the body frame, to be recovered in a linear way and integrated into the final solution after the inertial observations have already been integrated. Simulations are conducted to confirm that they perform as expected.

The effect on the linearity of the solution by correcting for inertial biases after integration has been performed is investigated and simulations are used to confirm that its effect is negligible.

Finally a comparison of the solution obtained with pre-integrated inertial delta observation is made to that of standard inertial integration techniques.

Chapter 6 introduces a different concept where the inertial observations may be used to estimate the scale of a map constructed with bearing only SLAM. Traditionally the scale of a map from bearing only SLAM is not recoverable which makes use of inertial observations without the initial velocity, or some other scale cue, impossible. This chapter provides an insight into how the scale of this map can be derived from the combination of bearing only landmark and inertial observations so they can still be used to aid each other.

Chapter 7 contains a number of practical considerations that need to be made in order to construct a real-world implementation of visual/inertial SLAM using pre-integrated inertial delta observations.

The concept of observation edge energy with respect to pre-integrated inertial observations is explored and the equations required to calculate it are developed. A comparison between edge energy and innovation is also discussed.

This chapter also provides a discussion as well as a number of possible solutions to the problems for visual feature extraction and matching including outlier rejection as well as pose initialisation. A method for incorporating gyro observations into the calculation of the fundamental matrix for visual data association outlier rejection and pose initialisation is also presented. This technique provides advantages when few matched visual features are available and provides more accurate camera motion estimates as well.

A method for forcing independence between different parts of the trajectory covered by the sensor platform to reduce computations and isolate errors is presented along with the benefits of using this technique.

Chapter 8 presents the sensor suite used to collect real-world data to test the algorithms presented in previous chapters. Results from a number of datasets are presented and analysed to investigate the performance of the implemented algorithms.

Finally, **Chapter 9** summarises the contributions of this thesis and provides suggestions for future research.

Chapter 2

Inertial navigation

2.1 Underlying science

Inertial measurement units (IMU) measure the specific forces and rotation rates experienced by the sensors in the unit. These specific forces and rotations rates are measured relative to an inertial frame of reference.

An inertial frame of reference is one where, in the absence of gravity, a body that is stationary or travelling at a constant velocity with respect to the frame will experience no new specific force or rotation rate. In other words the inertial frame is one in which Newton's first law of motion is valid [44].

When these measured specific forces and rotation rates are integrated, the net change in velocity and change in orientation of the inertial measurement unit can be calculated. If the initial position, velocity and orientation of the unit is known then these results can be combined to obtain estimates of the current position, velocity and orientation of the unit in the chosen navigation frame. This is the underlying idea behind inertial navigation.

As well as the acceleration and rotation of the body there are other sources that affect the observations made by the inertial measurement unit. These effects complicate implementations of inertial navigation solutions as they need to be accounted for in the solution to obtain the correct estimated trajectory.

These additional effects on inertial observations include gravity, earth rotation, centrifugal force, Coriolis force, coning and sculling.

2.1.1 Inertial navigation development

The majority of inertial navigation development started in the 1940's for use in guided missiles to provide a stand alone navigation solution. Since then it has commonly been used in aeronautical applications especially with the aiding of other sensors, such as GPS, allowing lower grade inertial units to be used for accurate long duration navigation.

More recently with the invention of small low-cost strap-down inertial measurement units, using inertial observations to aid in robotic applications has become popular.

2.2 Inertial measurement units

Inertial measurement units come in two basic varieties, traditional gimbaled systems where the accelerometers and gyroscopes are mounted on a gimbaled table that is kept level with respect to the navigation frame, and more recently strap-down inertial systems where the sensors are fixed with respect to the body frame of the vehicle they are mounted on.

Gimbaled inertial systems have been around a lot longer and are used as sensors with smaller dynamic ranges are required and the observations can be more easily integrated to obtain the navigation solution as the sensors themselves keep a constant orientation with respect to the navigation frame. Gimbaled systems generally can provide more accurate estimates however they are significantly more expensive to produce, are heavier and larger and require more power to operate than strap-down systems.

Lower cost strap-down inertial measurement units have been available more recently and have been made possible with the development of low cost micro-electromechanical systems (MEMS) sensors and faster computers that are capable

of performing the more complicated integration calculations required by strap-down systems. As a result strap-down inertial measurement units are generally lower cost, smaller, lighter and require less energy to run than traditional gimbaled systems. An additional advantage of strap-down systems is that as there are no moving parts they can generally operate in situations where higher accelerations and angular velocities are experienced.

The use of MEMS sensors allows the strap-down inertial measurement units to be produced at low cost, however some strap-down systems use ring laser or fibre optic gyroscopes instead to provide more accurate observations.

2.2.1 Inertial reference frames

There are a number of different reference frames that can be used to perform the calculations for inertial navigation. Figure 2.1 shows some of the possible frames that can be used.

The earth inertial frame is fixed to the centre of the earth but does not rotate with respect to the fixed stars. This frame is often used for long duration navigation as it is considered to be an inertial frame of reference.

In contrast the Earth Centred Earth Fixed (ECEF) frame rotates with the earth so that a fixed point on the surface of the Earth has a fixed location in ECEF coordinates. This is convenient when observations of features on the ground are made as the features remain in a fixed location in the reference frame, however the rotation of the earth needs to be accounted for if accurate gyros are used over long periods of time.

The local level frame is often used when the vehicle only operates in a small area over which the curvature of the earth can be ignored. The frame is fixed to the surface of the earth with the local gravity vector facing down. This is a common inertial reference frame for robotic applications where low cost IMUs are used that are not sensitive enough to observe the rotation of the earth.

The body frame has a fixed position and orientation with respect to the vehicle with the IMU. This is a convenient frame to use for strap-down IMUs as the

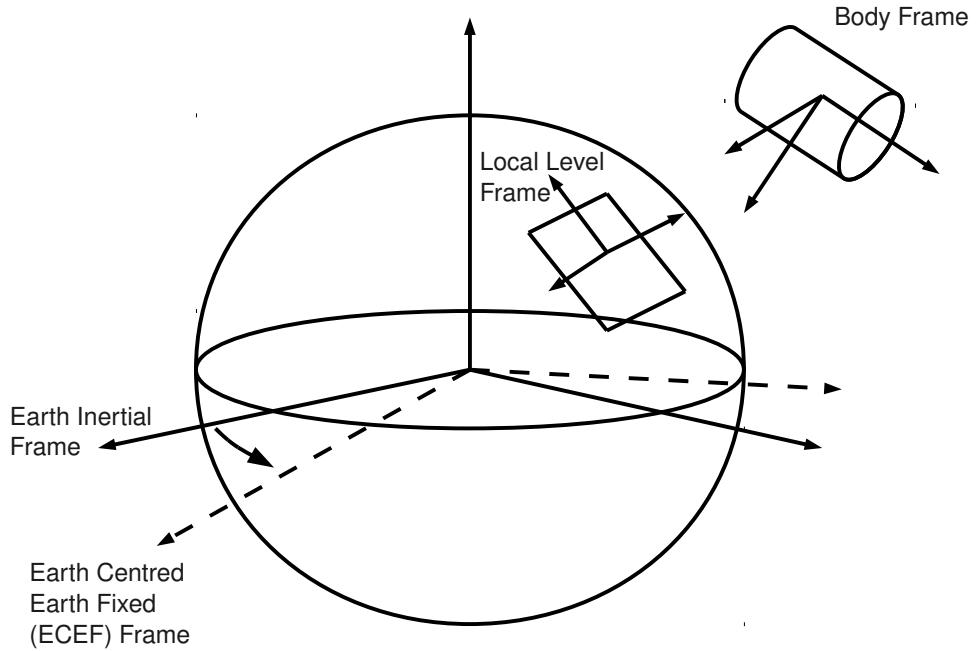


Figure 2.1: Different inertial navigation frames used for integrating inertial observations. The earth inertial frame is fixed to the centre of the earth but does not rotate with it, while the ECEF frame rotates with the earth so a point on the surface of the earth has a fixed location in the ECEF frame. The local level frame is fixed to a point in the area of operation of the vehicle and aligned with gravity vector pointing down. The body frame is fixed to the body of the vehicle.

sensors are fixed to the body frame of the vehicle. In this frame the orientation of the IMU in the frame is known but the gravity vector with respect to the frame is required.

2.2.2 Reference frame selection

The choice of which reference frame to use depends on the sensors and application but they can roughly be categorised into two different types, geographic consistent and platform consistent.

Geographic consistent reference frames are ones in which velocity and attitude

errors are defined in a computed navigation frame. These include the earth inertial frame, the ECEF frame and the local level frame.

Platform consistent reference frames are ones in which the velocity and attitude errors are defined in a body centred reference frame. The body frame is a platform consistent reference frame.

Geographic consistent reference frames generally perform better if navigation frame relative sensors are used, such as GPS. Platform consistent reference frames perform better if body frame relative sensors are used, such as Doppler velocity or vision observations [51].

2.3 Inertial observation models

Inertial measurement units provide observations of accelerations and rates of rotation. They are then integrated to provide predictions of the position, velocity and orientation of the platform.

The equations used to perform this integration differ depending on whether a gimbaled or strap-down IMU is used as well as the reference frame selected. Compensation for earth rotation, centrifugal force, Coriolis force, coning and sculling also add complexity to these equations if used.

The $15^\circ/s$ rotation of the earth can be observed by sensitive IMUs, which is why it is considered in some inertial navigation models. This rotation and the displacement of the vehicle from the centre of the earth also results in a slight centrifugal force, however this force is negligible when compared to gravity. Coriolis force is another artifact of the earth's rotation that causes a body travelling at high velocity parallel to the equator to experience a force pulling it towards the equator. This effect is usually only significant when travelling at high velocities over long distances.

Coning and sculling are side effects of the discrete sampling and integration of strapdown gyroscopes and accelerometers. Coning is caused when the rate at which the gyros are sampled and integrated is close to the rate at which the

rotation rate of the vehicle is changing. Since the integration method chosen has to make assumptions about the rotation rate over the integration interval, such as constant rotation rate, low IMU sampling rates can lead to increasing integration errors. This is only a significant problem when the rotation rate of the vehicle is changing rapidly, such as from high frequency vibration. The worst case scenario is where the axis of rotation sweeps in a cone, which is where the name coning comes from. Sculling is a similar phenomenon but concerning accelerometer integration.

For the kinds of applications considered in this thesis where low cost strap-down IMUs are used and the vehicle operates in a small area over a relatively short period of time, it is generally acceptable to use simplified equations.

For robotic applications, the inertial equations used usually do not take into account earth rotation as the low cost units used often are not sensitive enough to observe it. Coning and sculling are not taken into account as this usually does not cause problems unless the platform experiences vibrations with frequencies close to the sampling rate of the inertial sensors. Coriolis forces are also not considered as robotic platforms usually do not travel fast enough or in a straight line for long enough for this to be a problem.

A local level frame is also often used for robotics as it simplifies the equations and the small area of operation of the robot allows the curvature of the earth to be safely ignored.

2.3.1 Simplified strap-down inertial modelling

The basic simplified equations for integrating observations from a strap-down IMU are shown in equations 2.1, 2.2 and 2.3. These equations are performed in a local level navigation frame and require initial values for the position, velocity and attitude of the platform in that frame.

$$p_{t2}^n = p_{t1}^n + \int_{t1}^{t2} v_t^n dt \quad (2.1)$$

$$v_{t2}^n = v_{t1}^n + \int_{t1}^{t2} (C_{bt}^n f_t^b + g^n) dt \quad (2.2)$$

$$\phi_{t2}^n = \phi_{t1}^n + \int_{t1}^{t2} E_{bt}^n \omega_t^b dt \quad (2.3)$$

The C_b^n term is a rotation matrix that transforms vectors from the IMUs body frame in which the observations are made into the local level navigation frame. The E_b^n term is a rotation rate matrix that transforms rotation rate vectors from the body frame to the navigation frame.

The continuous form for strap-down inertial integration in equations 2.1,2.2 and 2.3 show the theory behind this process. Since inertial navigation is often implemented in digital computers and the IMUs used provide discrete observations at a fixed rate, a discrete form of these equations is required as shown in equation 2.4.

$$\begin{bmatrix} p_{t+1}^n \\ v_{t+1}^n \\ \phi_{t+1}^n \end{bmatrix} = \begin{bmatrix} p_t^n + v_t^n \Delta t + \frac{1}{2} [C_{bt}^n f_t^b + g^n] \Delta t^2 \\ v_t^n + [C_{bt}^n f_t^b + g^n] \Delta t \\ \phi_t^n + E_{bt}^n \omega_t^b \Delta t \end{bmatrix} \quad (2.4)$$

2.4 Aiding inertial navigation

The solution to the inertial integration equations gives an estimate of the position, velocity and attitude of the platform however the unavoidable influence of a number of sources of error will cause that solution to drift from the true value. These sources of error include biases in the inertial sensors, integration errors from inaccurate initial conditions and accumulated integration error of the noise in the inertial observations.

As inertial errors can quickly grow, especially for low cost units or if the initial conditions are incorrect, it is useful to use observations from other sensors to aid the inertial estimates and constrain this drift.

2.4.1 Sources of inertial aiding

A number of different sensors can be used to aid inertial navigation such as GPS for position updates, Doppler radar for velocity updates as well as observations from cameras, laser scanners and magnetometers. In fact any sensor that can observe either the position, velocity or orientation, or a change of one of these states in the navigation frame can be used to aid inertial estimates. If the correlations between these states are maintained then updating one state will lead to an improvement in the accuracy of all states of the vehicle's pose.

2.4.2 How is inertial aiding implemented?

In order for inertial estimates to be optimally fused with observations from other sensors to produce more accurate pose estimates, an estimate of the uncertainty of the estimates and the correlations between them needs to be calculated. This is required in order to know how much to trust the estimate when it is updated with other observations and how the update of one of the estimated states of the vehicle's pose affects the estimates of the other states.

The Kalman filter developed in the early 1960's [32] provides a method to calculate and keep track of the estimate uncertainties and correlations allowing observations to be fused in an optimal way. The Kalman filter is a linear estimator and therefore the non-linear inertial observation equations need to be linearised in order for a Kalman filter to be applied to them.

The use of a linear estimator is one of the main requirements for the initial conditions of the vehicle to be known in order to perform inertial navigation as they are needed in order to determine the linearisation points to be used.

2.5 Initial alignment

Many inertial navigation implementations require an initialisation stage where the initial orientation and velocity of the platform is obtained. As the inertial

integration equations are highly non-linear with respect to orientation estimates this process is usually done as a separate stage operating as a batch solve on the non-linear equations.

The implementation of the initial alignment is heavily dependent on the application and choice of error parameterisation. [2] provides a broad over view of many different coarse and fine initial alignment techniques used in inertial navigation for aerospace applications.

The error parameterisation used for INS systems are mostly separated into a choice of two different frames of reference. The true frame or perturbation approach and the computed frame or psi angle approach. These two approaches are a left over from gimballed inertial navigation systems, their equations have been adapted for use with strapdown IMUs [6] however they have lost their physical distinction and can be shown to be roughly equivalent for these systems.

Coarse initial alignment techniques are usually designed to estimate the initial orientation of the IMU with respect to the navigation frame to within a few degrees at which point small angle approximations can be made to refine the estimate once more information is available. A common form of coarse alignment is to use the accelerometers to sense the gravity vector, giving a rough roll and pitch estimate [51] and then use gyrocompasing to sense earth rotation to obtain the yaw estimate. This last step can take a long time to converge and is only possible with high grade IMUs which is often not the case for robotic applications.

Once coarse alignment has been achieved, fine alignment algorithms are used to fine tune the estimates of the vehicle attitude. [6] provides an overview of the differences in fine alignment equations for the true frame and computed frame parameterisations for inertial navigation solutions in aerospace applications. [58] introduces a dual quaternion over parameterisation of the inertial navigation problem which promises to provide a concise and unified representation facilitating simplified analysis of the navigation errors.

2.5.1 Removing initialisation stage requirements

The initial condition that usually needs to be known to the greatest accuracy due to its highly non-linear effects on the inertial integration equations is the orientation of the vehicle with respect to the inertial frame.

Some work has been done to remove, at least partially, the need for this explicit initialisation step by developing linearised solutions that can deal with large orientation uncertainties around at least one of the three axes [36, 51]. These methods mostly focus on re-parameterising the orientation so that the estimated states have a more linear effect on the inertial integration equations, however they usually also require the introduction of a non-linear constraint.

These methods are mostly concerned with long distance navigation in a geographic consistent reference using GPS observations. As a result they require observations that are referenced to the navigation frame, such as GPS, not body frame referenced observations such as those provided by cameras. Also since the initial orientation estimate errors are large and these filters do not relinearise, they require the filter to have a forgetting factor to remove the effects of poor linearisation of the early observations.

This is not a problem for a pure navigation solution as no states are permanently retained in the filter and the normal process noise will slowly remove the effects of these errors. However in SLAM applications this is not the case as the landmark location estimates, which are retained in the filter for long periods of time, retain the effects of these early linearisations. Also if a delayed state or smoothing implementation is used where past poses are retained in the filter, the effects of these linearisations will continue to affect these estimates as well.

2.6 Pre-integration of inertial observations

IMUs usually run at a high sample rate, on the order of hundreds of samples per second. This usually requires the filter to run at this high prediction rate as well. Inertial observations then traditionally need to be transformed into the

navigation frame at this high rate for inertial integration to be performed [58]. This is a result of the traditional computed or true frame parameterisation of the inertial error models left over from gimballed inertial navigation systems which require integration in these frames.

Instead of running a full update of the navigation solution for every observation from the inertial measurement unit, some work has focused on pre-processing the high rate inertial observations so that the navigation solution can be updated at a lower rate without losing accuracy.

Savage [50] presents what is claimed to be an optimal method for integrating inertial observations that is said to be exact when certain assumptions about the motions of the platform are made. This is true for the assumptions for the motion made in this paper, however for different platforms these assumptions may not be valid so quantitative analysis of this method could be problematic and provide negligible improvement in performance.

In some inertial measurement units, such as the Honeywell HG1900, a number of observations are integrated and processed to account for coning and sculling. These processed results are then made available to the user as delta observations in velocity and orientation only. These integrations are usually performed over short periods of time, for example 10ms, and then when used a constant velocity, constant acceleration, or similar assumption has to be made over that period to be able to update the position estimate of the platform.

Chapter 3

Visual SLAM

The problem of simultaneous localisation and mapping (SLAM) has been around for a number of decades now. It involves a robotic platform operating in an unknown environment using on-board sensors to build up a map of that environment and use that map to keep track of its location. The pose states of the platform as well as the locations of landmarks in the environment composing the map are estimated simultaneously without the need for a priori information.

Figure 3.1 shows an example of the SLAM problem. Sensor observations are made of the true landmark locations from the true vehicle locations and used to estimate the location of these landmarks relative to the initial vehicle location. Because of the inherent sensor noise these estimated feature locations have some uncertainty leading to the estimated landmark location differing from the true location.

As the vehicle moves subsequent observations of these landmarks are used to correct the estimated vehicle location as well as refine the estimate of the landmark location. The errors in the estimated landmark locations and sensor noise from subsequent observations leads to a slight error in the estimated new vehicle location however re-observation of previously observed landmarks can bound this error.

In probabilistic SLAM this can be represented as the estimation of a joint

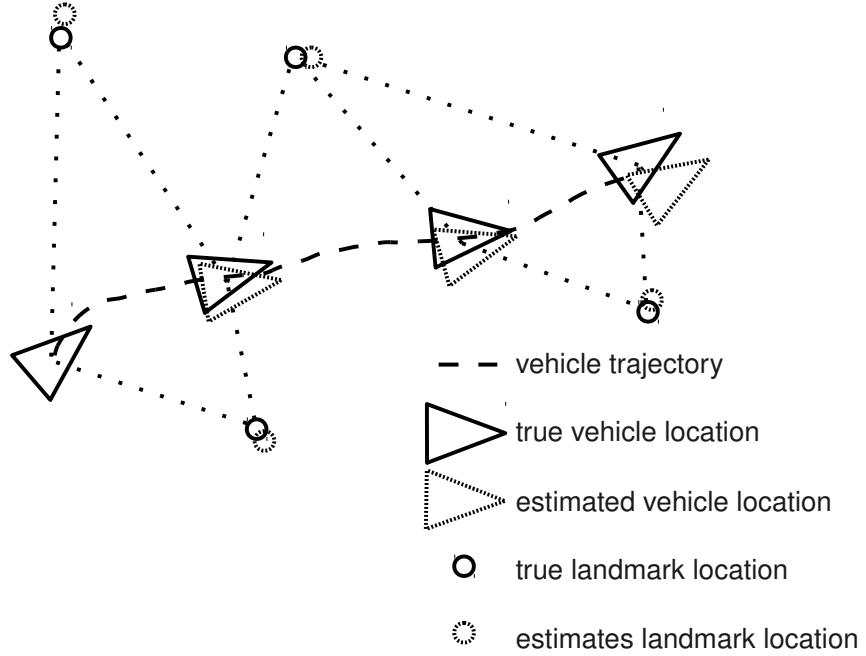


Figure 3.1: Representation of the SLAM problem.

posterior density of the landmark locations and vehicle states given the recorded observations and any initial vehicle state information available. This density is expressed in equation 3.1 with the estimated vehicle states, x , and landmark locations, m , given the observations, z , and a priori information, x_0 .

$$P(x, m|z, x_0) \quad (3.1)$$

An observation model is used to describe the probability of making an observation, z_k , given the current estimated vehicle states and landmark locations. This observation model can be expressed as the conditional density shown in equation 3.2.

$$P(z_k|x, m) \quad (3.2)$$

The joint posterior density forming the SLAM solution can then be updated with this observation information using Bayes' equation as shown in equation 3.3.

$$P(x, m | z_{0:k}, x_0) = \frac{P(z_k | x, m) P(x, m | z_{0:k-1}, x_0)}{P(z_k | z_{0:k-1}, x_0)} \quad (3.3)$$

Further explanation of the SLAM problem can be found in [17].

A number of different sensors can be used for performing SLAM including sonar, laser scanners [45], radar [16] and both monocular [14] or stereo [37] cameras. All these sensors can be used to build up and localise within the map.

A number of other types of sensors such as GPS and inertial measurement units [9, 24] can also be used to aid in SLAM by refining the navigation solution and constraining the robotic vehicle's position within some frame of reference.

Out of these possible sensors, using body frame sensors, such as cameras, to aid in inertial SLAM is the main application considered in this thesis.

3.1 Visual SLAM

Visual SLAM has received a lot of research interest in recent years for a number of reasons. Cameras are cheap, light and have low power requirements compared to other sensors for SLAM such as laser scanners. This makes them attractive sensors for low cost situations or for applications where size and weight need to be tightly controlled, such as for micro aerial vehicles [54].

Cameras are also passive sensors, unlike sonar, radar and laser scanners, which makes them useful in surveillance applications as they are difficult to detect and do not interfere with the environment they are observing.

A large amount of information can also be obtained from a single image or a sequence of images which allows the motion of the platform to be tightly constrained and for highly detailed maps to be constructed. This is especially useful when loop closure occurs as it can be easily detected by comparing the images.

One of the main draw backs of using cameras for SLAM is the large amount of computationally intensive processing that needs to be performed on the images to

extract and match landmarks. Also since cameras are projective sensors providing bearing only observations, observations from a single image can not provide an estimate of the range to features and if a single camera is the only sensor used, the true scale of the map generated is not observable.

The combination of the large number and rate of observations from cameras as well as the non-linear nature of the projective observations creates a huge computational burden on the SLAM filter. A number of approaches have been proposed to reduce this load including defining landmarks in a series of local coordinate frames such as shown in [18]. This method helps by reducing the relative uncertainty in the landmark position when taken in the relative frame and can be likened to the work in [4] and [7]. Submapping can also provide a great reduction in computational load and improve consistency as shown in [57].

In [34] and [35] a different approach to SLAM where the localisation and mapping components are separated is developed, similar to the approach in [56]. However in this case not all camera images are used to make the map. Instead a few distinct key frames are used to construct the map using techniques similar to Structure From Motion (SFM) and then the map is used to place the camera given the current image. This method is mostly beneficial when the camera operates in the same small area.

3.1.1 Stereo implementations

One way to get around the problem of estimating the range to a landmark from the first observation and obtaining the true map scale is to use two or more offset cameras, for example in a stereo rig. Using more than one camera to observe the same feature provides bearing observations from two different locations with a known offset between them allowing the range to be estimated.

This can work well for relatively close features, up to tens of meters depending on the baseline between the cameras, but can not provide sufficiently accurate range estimates of far away landmarks from a single pair of images.

A number of successful SLAM implementations with stereo cameras have been

implemented. Mostly for ground based applications [37, 47] in build up environments where there are always a sufficient number of close landmarks to constrain the camera’s position, and in underwater applications [19], where limited visibility makes it difficult to detect far away landmarks anyway.

Stereo implementations on aerial vehicles are rare, however there are a few isolated examples [38]. This is probably because in micro aerial vehicle applications the additional weight of a second camera is not always justified by the amount of extra information it provides. In larger scale applications the flying altitude is so high compared to the possible baseline of the cameras that the accuracy of the range estimates from stereo observations is very poor, making them virtually monocular.

3.1.2 Monocular implementations

A pioneering work in monocular SLAM where a single camera is the only sensor used for map construction and localisation was performed by Davison [14]. In this work four initial reference landmarks set at known locations are used to resolve the map scale ambiguity and to constrain the estimated camera location until other natural landmarks are initialised.

New landmark initialisation is performed out of the SLAM filter where the possible range to the landmark is represented over a limited range by a particle distribution until it has converged sufficiently to be approximated by a Gaussian distribution at which point it is added into the SLAM filter. This method of delayed landmark initialisation is required as the projective nature of the camera makes it impossible to obtain an estimate of the new landmark range until the camera has moved sufficiently to give an appropriate baseline.

Other methods for initialising new features from bearing only observations include the technique by Bailey in [3]. In this paper the author stores the observations of a potential new feature as well as retaining the poses where they were observed from in the state vector and computes the Kullback-Leibler distance between a non-Gaussian approximation of the feature range estimate and the result

of the feature initialisation. When this divergence reduces to a sufficiently low level the feature is initialised in the SLAM filter.

This method is computationally intensive due to the requirement to construct the non-Gaussian range estimate and the calculation of the Kullback-Leibler distance. In [8] Bryson addressed this issue by constructing a heuristic for initialisation of new features based on simulation of the given problem.

The major drawback of all the above mentioned delayed feature initialisation techniques for monocular SLAM is that the landmarks are not available for estimating the camera location until they are initialised, which may take a long time depending on its motion. As a result other methods of constraining the camera position estimate during these periods must be used. Initially known landmarks in [14] and navigation results from odometry [3] or a GPS/INS solution [8] were used in the other examples.

A novel solution to this problem was presented by Montiel in [42]. Instead of representing the landmarks in the normal Cartesian space, and inverse depth representation was introduced. Landmarks are defined by the location of the camera where they are first observed from, the azimuth and elevation bearing to the feature from that location, and the inverse of the depth from the camera to the new feature.

The inverse depth of the feature is used instead of the depth since for small base lines the probability distribution over the inverse depth of the feature more closely approximates a Gaussian distribution than the depth does.

However this alone does not solve the problem as even though we now have a more accurate way to represent the feature location from a small base line observation, no estimate of the range exists yet. This is solved by applying an initial heuristic on the range estimate after only the first observation with a large variance. Even though this initial heuristic causes biasing in the landmark location and can cause a skewing of the map, in practice it has been shown to work for short periods under certain types of motion.

The initial heuristic also applies a false observation of the average scale of the

map constraining the scale estimate, although there is no relationship between the map scale generated and the true scale of the environment. It merely removes a computational hurdle for the estimator.

Despite all its advantages, the standard inverse depth implementation still has a few drawbacks. While the inverse depth representation is more accurately approximated by a Gaussian distribution for small base lines, once the base line of observations of a feature grows the Cartesian parameterisation becomes more accurate. Therefore at some point a change from inverse depth to Cartesian representation should be preformed.

The biasing and skewing effect of the initial heuristic range estimate could also cause problems when operated over longer periods of time. It would be useful to be able to remove this prior at a later time once the landmark range estimate is more refined however then the scale observability issue becomes a problem again.

3.1.3 Structure from motion

A different approach to the problem of constructing a map from camera observations has been taken by the computer vision community in what is referred to as Structure From Motion (SFM). A comprehensive overview of this field of work can be found in [28].

Rather than using some kind of process model or navigation sensors to predict the location of the camera when a new image is taken and then use the image observations to refine that estimate, SFM techniques generally attempt to use the image observations alone to jointly solve for the new camera and feature locations in a single step.

This technique has the advantage of being able to estimate the relative rotation and direction of motion of the camera between two images alone without requiring any prior information or additional sensors or motion models. The distance of translation can not be determined due to the same scale observability problem discussed in section 3.1.2.

The simplest SFM technique for reconstruction of a map and trajectory from

two images is to construct the fundamental matrix. The fundamental matrix is a way of representing the geometry between the view points of two projective cameras. It is independent of the structure of the scene being observed but can be calculated purely from matched feature observations between the two views [29]. This can be done from as few as seven feature correspondences, the result of which can be used to obtain the camera rotation, translation direction and feature locations up to an arbitrary scale factor.

The fundamental matrix operates through the fundamental matrix constraint shown in equation 3.4.

$$x'^T F x = 0 \quad (3.4)$$

x and x' are the image locations of a pair of matched features in the first and second image respectively defined in homogeneous coordinates and F is the fundamental matrix.

There are even linear methods for recovering the fundamental matrix from eight or more observations [27] which can quickly be performed as part of a RANSAC routine to reject outliers in the feature association at the same time. This technique can also be performed on consecutive image pairs in a sequence of images if longer trajectories are to be recovered. A trifocal tensor can also be used on triplets of images in a similar way.

All of these methods are useful to provide an initial starting point for a more accurate iterative refinement stage if desired.

The advantage of this SFM technique over the undelayed inverse depth initialisation method mentioned earlier is that it does not impose a prior range heuristic so the landmark locations are not biased. However, at least 7 feature correspondences are required between each frame for this technique to work as no process model or initial feature location estimate is used.

Theoretically only 3 landmarks are required to be observed to constrain the camera location if the landmark locations are already known, and even less over a short period of time if an accurate process model is available to maintain the

estimate between observations.

3.2 Vision aided inertial

The use of observations from an IMU in vision based SLAM was first presented in [33]. Since then it has received attention by a number of people such as those presented in [8, 9, 20, 24, 48].

The addition of IMU observations to visual SLAM provides many advantages as they are complimentary sensors. While visual landmark observations provide low rate, non-drifting corrections to the estimated vehicle position, the IMU observations provide a high rate capturing of the vehicle motions that are important for control applications as well as for the prediction step on high dynamic platforms.

The IMU can also help in constraining the vehicle position estimate for short periods of time if there are insufficient visual observations to do so in a platform independent way without the need for a vehicle process model. The observation of the gravity vector from the accelerometers can provide a drift free constraint for the roll and pitch of the vehicle as well.

Along with these advantages, it is also possible to use the accelerometer observations along with the visual map generated from monocular visual SLAM to observe the scale factor of the visual map as shown in [41], which is not recoverable from monocular visual observations alone.

Traditionally inertial observations can only be integrated into a SLAM filter if the initial velocity and attitude of the platform are known to a high degree of certainty by using long established techniques [8], mostly derived from inertial navigation algorithms from the aerospace community [55].

3.3 Graphical SLAM

The majority of SLAM implementations have traditionally been developed using an extended Kalman filter (EKF) such as those presented in [8, 14, 16, 33, 48]. The extended form of the Kalman filter was used due to the non-linearity of the observation models requiring them to be linearised.

The standard EKF implementation uses what is called a filtering approach where past vehicle poses are marginalised out so that the joint probability density estimated contains only the current vehicle pose and the landmark locations, $P(x_i, m|z, x_0)$, where the i subscript indicates only the i th vehicle pose is present.

Some of the problems with using EKFs for SLAM include the $O(n^2)$ computational complexity, where n is the number of estimated states, which causes problems with large scale applications as well as the fixing of observation model linearisations at each step resulting in an accumulation of linearisation errors as the filter runs that can lead to divergence as presented in [10] and [31]. Data association can also only be performed on observations made from the current vehicle pose and can not be re-assessed at a later time if past vehicle states are not maintained.

In an effort to overcome some of these difficulties associated with EKF based SLAM a number of different filter structures have been explored in recent years [15, 19, 22, 26, 37, 57]. Of these the Graphical SLAM method presented by Folkesson and Christensen in [22] is of most interest for the applications considered in this thesis.

Graphical SLAM maintains individual estimates of each past vehicle pose at which observations are made, instead of just the most recent pose as in the EKF example. This approach is called smoothing or delayed state filtering as it keeps all the past states and smooths their estimates based on future observations. The estimated probability density can therefore be represented as $P(x_{1:i}, m|z, x_0)$ as all vehicle poses up to the current time i are maintained in the joint density.

Instead of representing the SLAM problem as a vector of estimated state

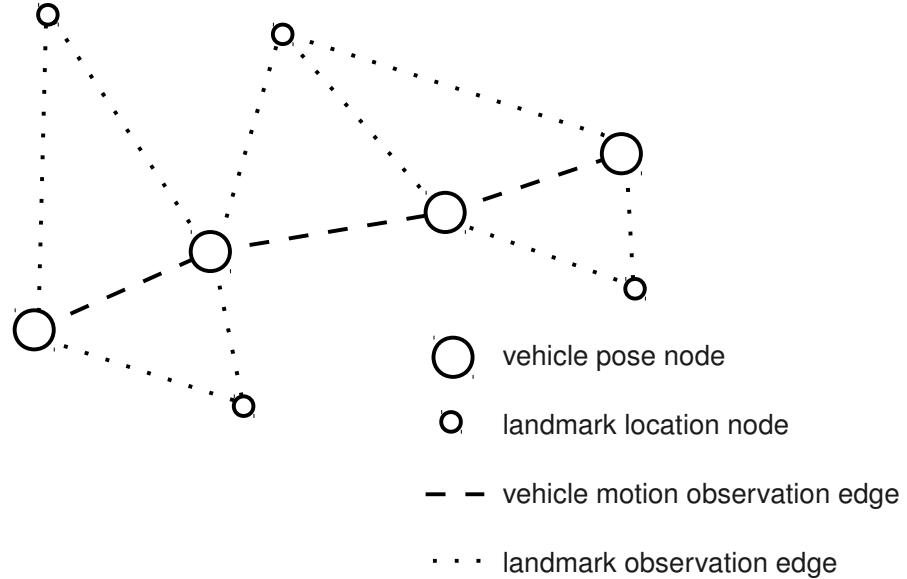


Figure 3.2: Representation of the Graphical SLAM formulation. Notice the similarity to the representation of the SLAM problem in figure 3.1.

means and a covariance matrix containing the correlations between these estimates, Graphical SLAM stores the vehicle pose and map estimates as a graph. The states of the vehicle at various times in the trajectory as well as the landmarks are stored as nodes in the graph with edges between these nodes representing the observations made.

An example graphical representation of a SLAM problem can be seen in figure 3.2 where the nodes containing the estimated states and the edges between them representing the observations made can be seen. Notice the similarity to the representation of the SLAM problem in figure 3.1. This structure allows for a number of operations to be performed on the graph that are not possible with normal EKF based SLAM.

Since all past poses are maintained in the graph and the observations are

stored separately as individual edges, relinearisation of the observations as more accurate state mean estimates become available as well as re-evaluation of data association at a later time now becomes possible. This leads to more consistent estimates as the accumulation of errors from poor linearisations and incorrect data association is reduced.

As new observations are made, new edges are added to the graph to represent these observations. At this point they are included in the graph but have not yet influenced the solution. This is the point where graph relaxation is performed.

Graph relaxation refers to the step where the estimated means of the node states are adjusted so that they agree more closely to the observations made. There are many different ways that graph relaxation can be performed such as direct solving by QR factorisation [15], belief propagation [43] or gradient descent [46] just to name a few. Graph relaxation can be performed at any time and does not have to be synchronised to the observation rates of the sensors used. Different sections of the graph can also be conditioned off and relaxed independently allowing a lot of flexibility in how the solution to a Graphical SLAM problem is managed.

Another advantage is that since correlations between distant features and poses do not need to be calculated as no marginalisation of states is performed, prediction steps become a constant time operation. This can also improve the computational complexity of update steps as adding new observations usually only significantly affect the estimated states that are close to the states involved in the observations. Using smart graph relaxation techniques or node grouping, such as the star nodes in [22], updates not involving loop closure can approach constant time complexity.

3.4 Outcomes of the literature review

From the available past research it is apparent that even though large scale metric SLAM over long trajectories is possible [11, 20], this is not where its strength lies.

In fact in many applications, even where the robot is operating over large areas, to quote [24], “One is concerned with local accuracy only and the global solution is not important.”

The most benefit can be obtained from metric SLAM where the vehicle spends a lot of time in the one area or the same area is revisited many times. This is where the multiple observations of the same landmarks can be combined to increase the estimation accuracy of the map and trajectory.

Over long trajectories where the vehicle is not revisiting the same area the results from metric SLAM algorithms have little advantage over less computationally intensive dead reckoning techniques such as [30]. In these situations, even if loop closure does occur the metric links are so weak due to the long trajectory that the local accuracy of the maps is not improved significantly.

There has been a trend in recent years where metric localisation is only used locally for control and aiding the robot in performing tasks while global localisation is implemented in a topological way [12, 25]. For large scale metric localisation, navigation with absolute position sensors, such as GPS-inertial navigation [9, 51, 55] seems to be much more useful. Metric SLAM is most beneficial when used on the smaller scale [14].

The use of graphical smoothing techniques seem to be well suited to this problem as it reduces some of the linearisation and data association issues associated with normal EKF SLAM. Various optimised graph relaxation techniques used to derive an estimate of state means on a graph can also be used that reduce the computation complexity of an update to as low as constant time. These updates enforce the local metric consistency that is desired but may not maintain global metric consistency in the map, but this is in line with the aims set out above.

In light of these observations, the focus of this research will be on developing a visual-inertial SLAM algorithm for use in small scale metric maps with the possibility for integration into a larger scale topological or similar mapping framework.

Chapter 4

Inertial Initialisation and Pre-Integration

4.1 Inertial aiding and estimation

4.1.1 Importance of linearity

Linear estimators, such as the Kalman filter (KF), are preferred for large problems due to their computational simplicity, fast convergence rates and stability of the estimator in the case of incorrect initial conditions. The down side of linear estimators is that they can only be used to solve systems of linear equations.

Many of the equations used in navigation and mapping are non-linear due to the nature of the sensors and process models used. One way to get around the non-linearity in the equations is to linearise them about a point and then treat the problem as a system of linearised equations which can be solved using a linear estimator. Such as in the extended Kalman filter (EKF).

The problem with this method is that the choice of an incorrect point to linearise the equation around will cause the derivative of the linearised equation to be different from the derivative of the non-linear equation resulting in the estimator converging to the wrong solution.

The correct point to linearise the equation around is the final solution however

this is not available at the time of linearisation. Assuming that the equation to be linearised is sufficiently smooth then an initial estimate that is close to the solution is normally used to provide the linearisation point. How close this point has to be to the final solution to provide sufficient results depends on the smoothness of the non-linear equation.

In SLAM or navigation situations a prior estimate of the solution is usually available from the result of the previous iteration of the algorithm or from initial conditions provided to the estimator. This initial estimate can be used to determine the linearisation points for the non-linear observation and process model equations.

Providing that these estimates are sufficiently accurate the estimator should converge to approximately the correct solution. However in cases where the filter has an infinite memory horizon, such as in SLAM, the slow accumulation of small linearisation errors will eventually result in inconsistent estimates. This is generally not a problem for pure navigation implementations without mapping as there are no states that are permanently retained in the filter so the accumulation of linearisation errors do not build up indefinitely.

There are methods that can be used to further reduce the effect of linearisation error accumulation at a cost of further computational complexity. The iterated extended Kalman filter (IEKF) is an example which performs an update with an initial state estimate and then re-performs the update linearising about the result of the first update. Information smoothing techniques such as square root information filtering [15] and Graphical SLAM [22] allow the equations to be re-linearised at any point in the running of the filter, not just after a single update, but at the expense of larger computational and memory requirements. The use of stabilising noise on the landmark states is another technique that can be used to reduce the chances of inconsistency from accumulated linearisation errors but at the expense of less accurate estimates.

All these techniques for linearising the non-linear system of equations for use in a linear estimator only work if a prior estimate of the states is available to be

used to obtain a linearisation point. In cases where no initialisation procedure is available these initial prior estimates may not exist. In situations like this they must either be obtained by a non-linear batch initialisation technique, if one is available, or the problem may be able to be re-parameterised to remove the non-linearity from the equations which means they can be estimated without the need for a prior to linearise about.

If the problem can be re-parameterised to remove the non-linearity or define the initial conditions as known, this not only creates an advantage by removing the requirement for initial conditions or an initialisation routine to be performed, but it also reduces the problem of accumulated linearisation errors during the normal estimation process as discussed above.

4.1.2 Where does the non-linearity come from?

The equations for integrating a single inertial observation with biases into a current vehicle state estimate is shown in equation 4.1.

$$\begin{bmatrix} p_{t+1}^n \\ v_{t+1}^n \\ \phi_{t+1}^n \end{bmatrix} = \begin{bmatrix} p_t^n + v_t^n \Delta t + \frac{1}{2} [C_{bt}^n (f_t^b - bias_f^{obs}) + g^n] \Delta t^2 \\ v_t^n + [C_{bt}^n (f_t^b - bias_f^{obs}) + g^n] \Delta t \\ \phi_t^n + E_{bt}^n (\omega_t^b - bias_\omega^{obs}) \Delta t \end{bmatrix} \quad (4.1)$$

When the derivatives of these equations are taken with respect to the vehicle states the non-linearities of the system can be seen and are shown in table 4.1. The non-linearity of these equations with respect to the observations is not important for a linear estimator as the observation values are already known.

Non-linearities occur when derivatives of the prediction with respect to the estimated states contain variables that are dependant on the estimated states. This causes non-linearities as the derivatives of the prediction change with the estimate of the state, thus making the values in the prediction Jacobian matrix dependant on the state estimate. If they are dependant then they must be recalculated when the estimate of the previous state changes, such as after an update. These non-linearities arise from the multiplication of two (or more) terms which

Table 4.1: Derivatives of the inertial prediction equations with respect to estimated states

	p_t^n	v_t^n	ϕ_t^n	$bias_f^{obs}$	$bias_\omega^{obs}$	g^n
p_{t+1}^n	1	Δt	$\frac{1}{2} \frac{dC_{bt}^n}{d\phi_t^n} (f_t^b - bias_f^{obs}) \Delta t^2$	$-\frac{1}{2} C_{bt}^n \Delta t^2$	0	$\frac{1}{2} \Delta t^2$
v_{t+1}^n	0	1	$\frac{dC_{bt}^n}{d\phi_t^n} (f_t^b - bias_f^{obs}) \Delta t$	$-C_{bt}^n \Delta t$	0	Δt
ϕ_{t+1}^n	0	0	$1 + \frac{dE_{bt}^n}{d\phi_t^n} (f\omega_t^b - bias_\omega^{obs}) \Delta t$	0	$-E_{bt}^n \Delta t$	0

are dependent on estimated states in the original inertial observation equation.

Non-linear derivatives are also present in the C_{bt}^n and E_{bt}^n terms as the rotation and rotation rate matrices are formed with non-linear trigonometric terms that are dependent on the estimated vehicle attitude.

Given the time interval, Δt , is known, the non-linearity appears in the derivatives of the predicted states with respect to the estimated attitude and biases only. Therefore if the previous attitude and biases can be known with complete certainty then the correct linearisation points for these terms will lead to better performance in a linear estimator as the remaining unknown states all have linear observation equations.

4.1.3 Importance of initial conditions

In unaided inertial navigation accurate initial conditions are required as a starting point for the integration of the inertial observations. An initial attitude estimate is vital as the effect of gravity has to be removed from the accelerometer observations, as shown in equation 4.1, before they can be integrated to obtain a navigation solution. If the initial attitude estimate is incorrect, the incorrect adjustment for the gravity vector will result in the integration of consistently incorrect acceleration observations resulting in rapid divergence of the solution. An accurate initial velocity estimate is also important as a starting point for the acceleration integration.

In aided inertial navigation initial conditions are still important due to the

non-linearity of the inertial integration equations as described in section 4.1.2. Jacobian matrices for the inertial observations are required in order to fuse inertial observations with other sensor information for aiding the inertial solution as covariance estimates for the estimated states are required. For this the derivatives of the inertial prediction equations must be calculated. As the prediction equations are non-linear the derivatives of the inertial equations are dependant on the initial conditions.

If these initial conditions are not correct then the derivatives will be wrong and if the error is too large when other sensor observations are fused the solution may fail to converge. Again this is most important for the initial attitude as an accurate attitude estimate is required to correct for the gravity vector and is the largest source of non-linearity in the inertial equations as show in table 4.1.

4.1.4 How are initial conditions obtained?

Initial conditions for inertial navigation and SLAM implementations are usually obtained by performing some predefined initialisation routine, such as placing the unit at a known stationary position and orientation [9], or by collecting data from the inertial unit as well as other sensors for a period of time while the unit is moving and then performing a non-linear batch solving technique to obtain the initialisation [36].

The problem with using the initialisation routine to obtain the initial conditions is that a specific initialisation stage must be performed before the unit can be used which may not be convenient or even possible in many real world situations. With the non-linear batch initialisation a navigation solution for the platform is not available for some time until the non-linear solver has converged which makes it un-useable for autonomous systems where the navigation estimate may be required for control and stability purposes.

In many SLAM implementations the initial conditions of the vehicle are usually arbitrarily defined with respect to the initial vehicle states. This is done when no globally consistent information about location is available and is convenient

for most small scale applications. The problem with this approach when using inertial navigations is that the initial orientation of the vehicle needs to be known with respect to the gravity vector and therefore can not traditionally be simply set to the initial vehicle orientation. Also if globally referenced sensors, such as GPS are used, the location of the vehicle in this globally referenced frame must also be known.

4.2 Achieving linearity

As the non-linearity in the inertial observation equations comes from the multiplication of two terms that are dependent on estimated states as well as the trigonometric terms in the rotation and rotation rate matrices, as mentioned in section 4.1.2, the performance in a linear estimator can be improved by re-parameterising the problem to fix one of these terms thereby providing the filter with the correct linearisation of the equation.

It can be seen from table 4.1 that the majority of the non-linearity comes from the rotation of the inertial observations from the body frame into the navigation frame through the uncertain C_b^n and E_b^n matrices. Figure 4.1 shows the reference frames used in this type of global navigation frame inertial integration. Each of the inertial observations are transformed into the navigation frame through an uncertain rotation. The uncertainty in this rotation is where the majority of the non-linearity comes from. If the problem can be re-parameterised to fix these rotation matrices then the major source of linearisation uncertainty can be removed.

If performed, this re-parameterisation will also simultaneously fix the linearisation of the inertial integration equations with respect to the IMU biases as a convenient side effect. The uncertainty about the IMU biases only affects the linearity of the inertial integration equations through the estimate of the attitude of the inertial unit as shown in table 4.1. As the re-parameterisation will result in the attitude of the platform being known with complete certainty, it is no

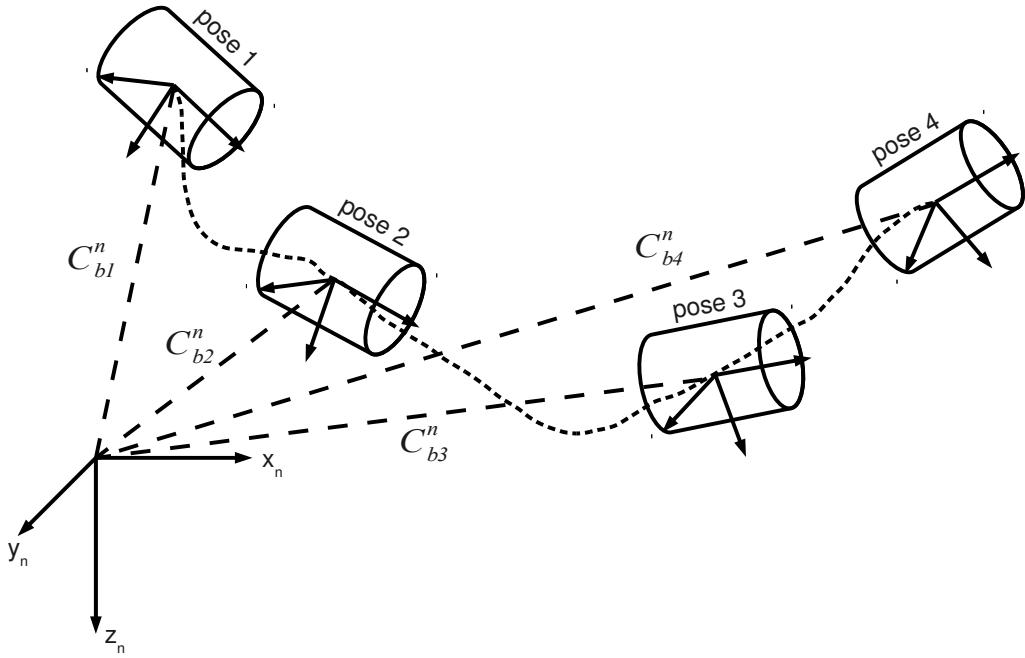


Figure 4.1: Illustration of global frame navigation where the inertial observations are transformed into the global frame for integration.

longer an estimated state of the system and thus this cause of non-linearity is also removed.

Even though now the initial attitude of the vehicle in the navigation frame is known with complete certainty, this knowledge comes with a trade off. In the new body referenced navigation frame the direction of the gravity vector is now unknown. Therefore the gravity vector now becomes an unknown state that must be estimated in the filter in place of the initial attitude. The advantage of this trade off when using a linearised estimator is that the gravity vector estimate, unlike the initial attitude estimate, is completely linear as can be seen from the last column of table 4.1.

Figure 4.2 shows the proposed body reference frame for integration of the inertial observations. Instead of transforming the inertial observations into some absolute navigation frame as in figure 4.1, the observations are transformed into the body frame of the first pose. As the relationship between the attitude of the

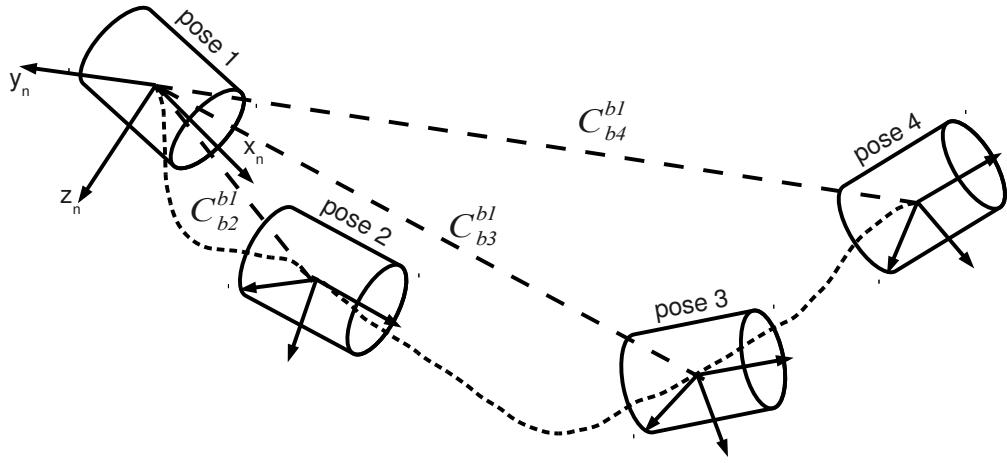


Figure 4.2: Illustration of body frame navigation where the inertial observations are transformed into the body frame of the first pose for integration.

current body frame and the first pose body frame can be known accurately from integrating gyro observations, this rotation can be done to a high degree of certainty removing much of the linearisation uncertainty from the inertial integration equations.

4.2.1 Re-parameterisation to determine initial conditions

The way to re-parameterise the estimated states so that the initial attitude of the inertial unit is known with complete certainty is to transform the navigation solution from the global navigation frame into the body frame of the inertial measurement unit itself at the first time step. This way the initial attitude of the IMU can be known as $[0, 0, 0]^T$ for roll, pitch and yaw respectively with complete certainty.

This parameterisation then fixes the C_b^n and E_b^n matrices which means they are no longer dependent on the estimated states and therefore no longer cause non-linearity in the inertial integration equations. Effectively this is equivalent to integrating the IMU observations in the body frame instead of the global frame

as is normally done [9, 24, 36, 55].

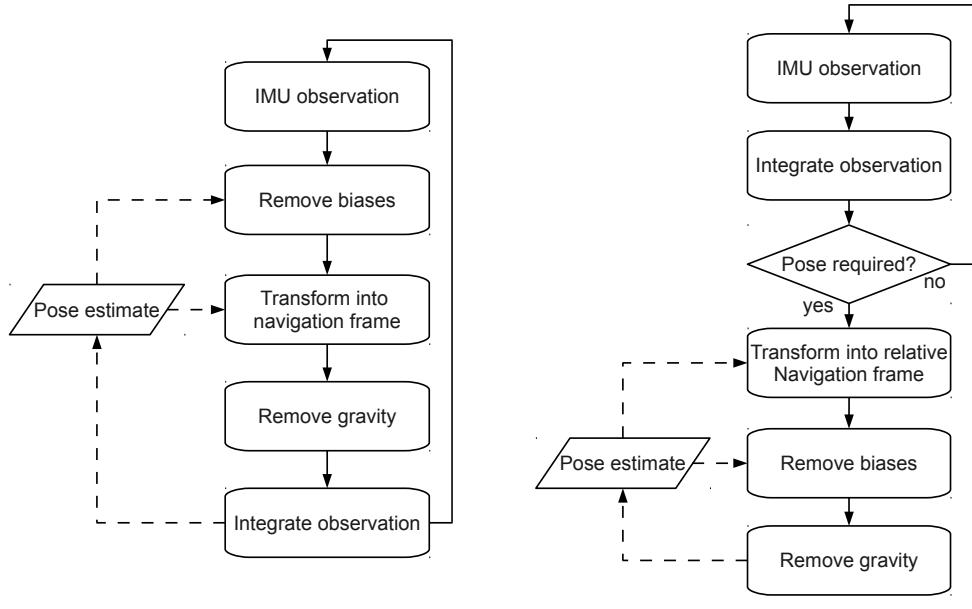
This fixing only applies to the first IMU observation that is integrated as this is the only observation that is fixed in this new frame. Subsequent inertial observations are taken in a body frame that does not necessarily align with this new navigation frame as the platform may have rotated since then. The uncertainty of subsequent attitude estimates will slowly introduce linearisation uncertainty back into the inertial integration equations.

Even though this is true the uncertainty is now relative to the initial attitude, not the global navigation frame, so integration of the gyro measurement will provide an estimate of these states which will be known to a fairly high degree of certainty such that non-linearity will not be a problem. The relative frame can also be redefined into the current vehicle frame at later points in time if this relative attitude uncertainty grows too large. This is usually done every time the pose of the vehicle in the navigation frame is desired, such as when observations from other sensors are taken.

Figure 4.3 shows a comparison of the re-parameterised body frame inertial integration technique to the commonly used global navigation frame inertial integration technique from [9]. It can be seen from this figure that the same basic steps are performed in both cases with only the order changed. Predominately in the global navigation frame technique, integration of the inertial observations is performed after they have been transformed into the navigation frame and gravity is compensated for. In the re-parameterised technique, inertial integration happens as the first step in the body frame and then transformation into the navigation frame and compensation for gravity is only performed when a vehicle pose estimate is required.

4.2.2 Removing initial condition requirements

Redefining the orientation of the navigation frame to align with the initial attitude of the IMU removes the initial attitude as a state that needs to be estimated in the filter. Therefore the initial attitude is no longer needed as an initial condition



(a) Standard inertial integration process as done in [9] (b) Re-parameterised inertial integration to improve linearity

Figure 4.3: Flowcharts comparing the steps in standard inertial integration to those of the re-parameterised form.

for the inertial integration.

In fact it can be seen from figure 4.3(b) that the integration of the inertial observations can be performed without any knowledge of any of the estimated states in the filter. Therefore the initial conditions of the vehicle are no longer needed for inertial integration, however the position and velocity of the vehicle in the navigation frame as well as the IMU biases are still unknown.

As well as these states, the fixing of the IMU attitude now requires the gravity vector in the new frame of reference to be estimated as well. This is because the removal of the effect of gravity still needs to be performed before inertial integrations can be used to give an estimate of the vehicle position in the navigation frame, as can be seen in the flowchart in figure 4.3(b). The direction of the gravity vector in the new navigation frame is unknown because the alignment of the new body frame in the inertial navigation frame is unknown at this time.

Therefore even though the initial conditions of the platform are no longer required for integration of the inertial observations, the initial position, velocity, IMU biases and gravity vector still need to be estimated if a global navigation frame estimate of the vehicle pose is required. However this is not a problem if the inertial integration is done in the body frame as these states now only affect the inertial observation equations in a linear way. Therefore a linear estimator can be used to estimate these states without the need for prior initial condition estimates.

If only relative navigation is required, and absolute position is not observable from the aiding sensors this is even easier as the navigation frame can be fixed to the body frame of the IMU at the first time step, giving a known transform between the body frame and the navigation frame.

4.3 Inertial pre-integration

Once the attitude of the IMU has been fixed and compensation for the effect of gravity has been delayed until a pose estimate is required, it is possible for the inertial observations from the IMU to be integrated before being incorporated into the filter and without the initial conditions being known. This will be referred to as inertial pre-integration.

By pre-integrating the inertial observations it is possible to treat a number of high rate inertial observations as a single observation without losing any information. This results in a significant reduction in the number of prediction steps required in a filtering application, and a similar reduction in the number of poses for smoothing.

Pre-integration also provides a simple way for inertial observations to be incorporated into a filter when initial conditions are not known as vehicle position, velocity and attitude information is no longer required for integration.

The main advantage of pre-integrating the inertial observations is the linearity of the resulting equations. The high degree of linearity in the pre-integrated in-

ertial observation equations results in more accurate observation Jacobian calculations which increases the consistency of the estimation. The increased linearity and more accurate Jacobians result in the estimation process converging faster and being more stable.

This linearity also reduces the need for re-linearisation if information smoothing based estimation is used reducing processing time.

4.3.1 Inertial delta components

If the inertial integration equations used to derive the discrete inertial update step shown in equation 4.1 are written in their continuous form, equations 4.2, 4.3 and 4.4 are obtained. These equations can be used to integrate any number of inertial observations for any time interval where $t2 > t1$.

$$p_{t2}^n = p_{t1}^n + \int_{t1}^{t2} v_t^n dt \quad (4.2)$$

$$v_{t2}^n = v_{t1}^n + \int_{t1}^{t2} (C_{bt}^n (f_t^b - bias_f^{obs}) + g^n) dt \quad (4.3)$$

$$\phi_{t2}^n = \phi_{t1}^n + \int_{t1}^{t2} E_{bt}^n (\omega_t^b - bias_\omega^{obs}) dt \quad (4.4)$$

Equation 4.2 can also be written as shown in equation 4.5.

$$p_{t2}^n = p_{t1}^n + (t2 - t1)v_{t1}^n + \int_{t1}^{t2} (C_{bt}^n (f_t^b - bias_f^{obs}) + g^n) dt^2 \quad (4.5)$$

These equations require the initial conditions of the system to be known in order to perform the integrations. However the components of the integrations that do not require the initial conditions can be separated out from the components that do.

These separated components without initial condition requirements are show in equations 4.6, 4.7 and 4.8 for the position, velocity and attitude integration respectively. These components can be formed without any information about

the initial conditions of the system and will be referred to as inertial delta components.

$$\Delta p_{t2}^{+t1} = \int_{t1}^{t2} C_{bt}^{t1} (f_t^b - bias_f^{obs}) dt^2 \quad (4.6)$$

$$\Delta v_{t2}^{t1} = \int_{t1}^{t2} C_{bt}^{t1} (f_t^b - bias_f^{obs}) dt \quad (4.7)$$

$$\Delta \phi_{t2}^{t1} = \int_{t1}^{t2} E_{bt}^{t1} (\omega_t^b - bias_\omega^{obs}) dt \quad (4.8)$$

Note that for these delta components, the C_{bt}^{t1} and E_{bt}^{t1} matrices are formed from the estimated attitude of the platform at time t with respect to the attitude at time $t1$. This is in contrast to the C_{bt}^n and E_{bt}^n matrices from equations 4.2, 4.3 and 4.4 which are with respect to the navigation frame. This way the initial attitude of the platform in the navigation frame at time $t1$ does not need to be known.

Once these delta components have been formed the equivalent results from equations 4.2, 4.3 and 4.4 can be obtained from substituting the delta components back into the original equations to obtain equations 4.9, 4.10, and 4.11.

$$p_{t2}^n = p_{t1}^n + (t2 - t1)v_{t1}^n + C_{bt1}^n \Delta p_{t2}^{+t1} + \frac{1}{2}(t2 - t1)^2 g^n \quad (4.9)$$

$$v_{t2}^n = v_{t1}^n + C_{bt1}^n \Delta v_{t2}^{t1} + (t2 - t1)g^n \quad (4.10)$$

$$\phi_{t2}^n = EulerFromRotationMatrix(C_{bt1}^n \Delta C_{t2}^{t1}) \quad (4.11)$$

For equation 4.11 the rotation matrices are formed from the estimate of the vehicle attitude at time $t1$ to form C_{bt1}^n and from the delta attitude component $\Delta \phi_{t2}^{t1}$ to form ΔC_{t2}^{t1} . These rotations matrices are then multiplied together to obtain the updated attitude at time $t2$ instead of using the delta Euler angles from

Algorithm 4.1 Inertial Delta Observation Creation

```

 $\Delta p_t^+ = 0$ 
 $\Delta v_t = 0$ 
 $\Delta \phi_t = 0$ 
for  $t1 < t < t2$  do
     $\Delta t = t_{t+1} - t_t$ 
     $f_t^{t1} = C_t^{t1} (f_t^b - bias_f^{obs})$ 
     $\Delta v_{t+1} = \Delta v_t + f_t^{t1} \Delta t$ 
     $\Delta p_{t+1}^+ = \Delta p_t^+ + \Delta v_t \Delta t$ 
     $\Delta \phi_{t+1} = \Delta \phi_t + E_t^{t1} (\omega_t^b - bias_\omega^{obs}) \Delta t$ 
end for
observation = 
$$\begin{bmatrix} \Delta p_t^+ \\ \Delta v_t \\ \Delta \phi_t \end{bmatrix}$$


```

equation 4.4 and a rotation rate matrix. This is done because the attitude delta is a difference in angles, not a rotation rate, also the change in orientation in the attitude delta may be large as the integration period may be long compared to the vehicle dynamics. As a result of these differences the small angle approximation made by the use of the rotation rate matrix, E_{bt}^n , in equation 4.1 may not be valid.

4.3.2 Inertial delta observation creation

Algorithm 4.1 shows an example of how the inertial delta components derived in section 4.3.1 can be formed from a series of ordinary inertial observations from an IMU. These resulting components can be used as a single observation in place of the number of inertial observations used to form them and therefore will be referred to as an inertial delta observation.¹

The $bias_f^{obs}$ and $bias_\omega^{obs}$ bias terms are the estimated IMU biases at the time the inertial delta observations are formed. In sections 4.3.3 the derivatives of the inertial delta observations with respect to these biases will be calculated so that small changes in the estimated bias from these values can be compensated for. In

¹The example in algorithm 4.1 uses first order Euler integration for clarity however higher order integration techniques can also be used

Algorithm 4.2 Standard Inertial Prediction

```

 $p_t = p_{t1}^n$ 
 $v_t = v_{t1}^n$ 
 $\phi_t = \phi_{t1}^n$ 
for  $t1 < t < t2$  do
     $\Delta t = t_{t+1} - t_t$ 
     $f_t^n = C_t^n (f_t^b - bias_f)$ 
     $v_{t+1} = v_t + f_t^n \Delta t + g^n \Delta t$ 
     $p_{t+1} = p_t + v_t \Delta t$ 
     $\phi_{t+1} = \phi_t + E_t^n (\omega_t^b - bias_\omega) \Delta t$ 
end for
prediction =  $\begin{bmatrix} p_t \\ v_t \\ \phi_t \end{bmatrix}$ 

```

many situations simply setting $bias_f^{obs}$ and $bias_\omega^{obs}$ to zero during delta creation and correcting for biases in the later estimation process is sufficient.

The delta observations calculated are in the reference frame of the vehicle at time $t1$ and therefore will be different for each inertial delta observation created. This relative reference frame that moves with the vehicle is what allows the initial conditions for the delta observation (p_{t1}^+ , v_{t1} and ϕ_{t1}) to be set to zero therefore removing the requirement for them to be known a-priori.

The downside of the moving reference frame is that the inertial delta observations all need to be transformed into the same frame before they can be combined in a filter but this can be easily done as will be explained in section 4.3.4.

Algorithm 4.2 shows the steps for a normal inertial prediction between two times $t1$ and $t2$, from equation 4.1², for comparison with the inertial delta observation steps in Algorithm 4.1.

It can be seen that the two algorithms are almost identical except for the initial conditions, the frame of reference and gravity vector correction. For the standard inertial prediction the estimated position, velocity and attitude at time $t1$ are required as priors but prior values of zero are used for the inertial delta

²Algorithm 4.2 uses first order Euler integration for direct comparison to the example in algorithm 4.1, therefore the position prediction step differs slightly from that in equation 4.1

observation creation. Also the inertial specific force (f_t^b) and rotation rate (ω_t^b) observations are transformed into the navigation frame for the standard inertial prediction steps by the C_t^n and E_t^n matrices, while in the inertial delta observation creation they are transformed into the body frame at time t_1 by the $C_t^{t_1}$ and $E_t^{t_1}$ matrices. Also for the pre-integrated method the gravity vector correction does not need to be considered until after the observations have been integrated, while for the standard technique this is done before integration.

4.3.3 Inertial delta Jacobian and covariance creation

The inertial delta observations created in algorithm 4.1 are fine for inertial navigation, however if inertial aiding is to be performed the Jacobian and covariance matrices for the delta observations are required so that they can be fused optimally with observations from other sensors.

The Jacobian and covariance matrices will be different for each individual inertial delta observation due to the different combination of rotations performed by the vehicle during the sampling period of the delta observation. Therefore these matrices must be individually created for each delta observation and can not be precomputed.

Algorithm 4.3 shows an example of how the Jacobian and covariance matrices for an inertial delta observation created using algorithm 4.1 can be formed. This procedure is inspired from the covariance calculation in the EKF prediction step, shown in algorithm 4.4 for comparison. The variable ordering used is position, velocity, attitude, accelerometer bias and gyro bias.

As algorithm 4.3 requires intermediate attitude values from algorithm 4.1 to produce the $C_t^{t_1}$ and $E_t^{t_1}$ matrices and their derivatives, these two algorithms should be calculated simultaneously for efficiency.

The J matrix is the Jacobian for the inertial delta observation with respect to the initial vehicle states and bias estimates. As the initial vehicle states are set to zero at the beginning of each delta, only the last two columns of this matrix, relating to the bias terms, are important for the result but the whole matrix is

Algorithm 4.3 Inertial Delta Jacobian and Covariance Creation

```

 $J_t = \mathbf{I}_{15}$ 
 $R_t = \mathbf{0}_{15}$ 
for  $t1 < t < t2$  do
     $\Delta t = t_{t+1} - t_t$ 
     $\alpha = \frac{dC_t^{t1}(f_t^b - bias_f^{obs})}{d\phi_t}$ 
     $\beta = \frac{dE_t^{t1}(\omega_t^b - bias_\omega^{obs})}{d\phi_t}$ 
     $F_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3\Delta t & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \alpha\Delta t & -C_t^{t1}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \beta\Delta t & \mathbf{0}_3 & -E_t^{t1}\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$ 
     $G_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ C_t^{t1}\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & E_t^{t1}\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$ 
     $J_{t+1} = F_t J_t$ 
     $R_{t+1} = F_t R_t F_t' + G_t Q G_t'$ 
end for
 $J_{t1}^{t2} = J_t$ 
 $R_{t1}^{t2} = R_t$ 

```

required for the internal calculations.

The R matrix is the covariance matrix for the inertial delta observation where the Q matrix is the inertial sensor noise covariance matrix for the IMU. The resultant covariance matrix will be of dimension 9×9 (only the first 9 states are used) however it will only be full rank if two or more inertial observations are integrated to form the delta observation, otherwise it will be of rank 6.

From comparison of algorithms 4.3 and 4.4 it can be seen that, similar to the comparison for the mean prediction step, the two main difference are the initial conditions and the frame of reference used. The standard inertial prediction covariance equations require the state covariance matrix at time $t1$, P_{t1} , as an initial condition whereas the inertial delta observations covariance calculations start with a zero covariance matrix and an identity Jacobian matrix. Also the

Algorithm 4.4 Standard Inertial Prediction Covariance Calculation

```

 $P_t = P_{t1}$ 
for  $t1 < t < t2$  do
     $\Delta t = t_{t+1} - t_t$ 
     $\alpha = \frac{dC_t^n(f_t^b - bias_f)}{d\phi_t}$ 
     $\beta = \frac{dE_t^n(\omega_t^b - bias_\omega)}{d\phi_t}$ 
     $F_t = \begin{bmatrix} \mathbf{I}_3 & \mathbf{I}_3\Delta t & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{I}_3 & \alpha\Delta t & -C_t^n\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 + \beta\Delta t & \mathbf{0}_3 & -E_t^n\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix}$ 
     $G_t = \begin{bmatrix} \mathbf{0}_3 & \mathbf{0}_3 \\ C_t^n\Delta t & \mathbf{0}_3 \\ \mathbf{0}_3 & E_t^n\Delta t \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix}$ 
     $P_{t+1} = FP_tF' + GQG'$ 
end for
 $P_{t2} = P_t$ 

```

calculations are performed in the body frame at time $t1$ for the inertial delta observation and in the navigation frame for the standard technique.

Another difference between the standard covariance calculations and the calculations for the inertial delta observation covariance is the requirement for the observation Jacobian matrix, J in algorithm 4.3, which is not required for the standard method. Elements from this matrix are required in case the inertial delta observations need to be modified if the estimate of the IMU biases change, this will be explained in section 4.3.4.

4.3.4 Inertial delta observation implementation

If pre-integrated inertial delta observations are to be used in the prediction step of a state space estimator, such as an EKF, the state prediction equations 4.9, 4.10, and 4.11 can be used directly. Then the observation covariance matrix R_{t1}^{t2} from algorithm 4.3 can be used for the process noise variance as normal.

However, in order for the inertial delta observations to be fused with other sensor information in an inertial aiding implementation, a method to predict the expected value of inertial delta observations given current estimates of the vehicle state is required.

Such a method can be found from rearranging equations 4.9, 4.10, and 4.11 to obtain equations 4.12, 4.13 and 4.14.

$$\begin{aligned} E(\Delta p_t^+ | \hat{x}_t) = h_{\Delta p+}(\hat{x}_t) &= \hat{C}_{bt1}^{n-1} \left(\hat{p}_{t2} - \hat{p}_{t1} - \hat{v}_{t1} \Delta t - \frac{1}{2} \hat{g} \Delta t^2 \right) \\ &+ \frac{d\Delta p_t^+}{dbias_f} \left(\widehat{bias}_f - bias_f^{obs} \right) \\ &+ \frac{d\Delta p_t^+}{dbias_\omega} \left(\widehat{bias}_\omega - bias_\omega^{obs} \right) \end{aligned} \quad (4.12)$$

$$\begin{aligned} E(\Delta v_t | \hat{x}_t) = h_{\Delta v}(\hat{x}_t) &= \hat{C}_{bt1}^{n-1} (\hat{v}_{t2} - \hat{v}_{t1} - \hat{g} \Delta t) \\ &+ \frac{d\Delta v_t}{dbias_f} \left(\widehat{bias}_f - bias_f^{obs} \right) \\ &+ \frac{d\Delta v_t}{dbias_\omega} \left(\widehat{bias}_\omega - bias_\omega^{obs} \right) \end{aligned} \quad (4.13)$$

$$\begin{aligned} E(\Delta \phi_t | \hat{x}_t) = h_{\Delta \phi}(\hat{x}_t) &= EulerFromRotationMatrix(\hat{C}_{bt1}^{n-1} \hat{C}_{bt2}^n) \\ &+ \frac{d\Delta \phi_t}{dbias_\omega} \left(\widehat{bias}_\omega - bias_\omega^{obs} \right) \end{aligned} \quad (4.14)$$

Where $E(\cdot)$ is the expected value operator, $h(x)$ is the observation prediction function and \hat{x}_t is the current estimate of the mean of the vehicle states. The $bias_f^{obs}$ and $bias_\omega^{obs}$ variables are the estimated IMU biases used in the creation of the deltas in algorithm 4.1 where as the \widehat{bias}_f and \widehat{bias}_ω variables are the filter's current estimate of these biases. \hat{p} , \hat{v} and \hat{g} are the current estimates of the vehicle's position and velocity as well as the gravity vector in the navigation

frame respectively.

The $\frac{d\Delta}{dbias_f}$ and $\frac{d\Delta}{dbias_\omega}$ derivatives are obtained from the last two columns of the delta observation Jacobian matrix, J , calculated in algorithm 4.3. These values can be easily extracted as the final Jacobian matrix from this algorithm has the form shown in equation 4.15.

$$J = \begin{bmatrix} \frac{d\Delta p_{t2}^+}{dp_{t1}^{t1}} & \frac{d\Delta p_{t2}^+}{dv_{t1}^{t1}} & \frac{d\Delta p_{t2}^+}{d\phi_{t1}^{t1}} & \frac{d\Delta p_{t2}^+}{dbias_f} & \frac{d\Delta p_{t2}^+}{dbias_\omega} \\ \mathbf{0}_3 & \frac{d\Delta v_{t2}}{dv_{t1}^{t1}} & \frac{d\Delta v_{t2}}{d\phi_{t1}^{t1}} & \frac{d\Delta v_{t2}}{dbias_f} & \frac{d\Delta v_{t2}}{dbias_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \frac{d\Delta \phi_{t2}}{d\phi_{t1}^{t1}} & \mathbf{0}_3 & \frac{d\Delta \phi_{t2}}{dbias_\omega} \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{dbias_f}{dbias_f} & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \frac{dbias_\omega}{dbias_\omega} \end{bmatrix} \quad (4.15)$$

These derivative terms with respect to the bias estimates, found in the last two columns, are the only values from this Jacobian matrix that are used.

Finally only the inertial delta observation prediction Jacobian matrix, H_t , needs to be calculated. This matrix defines how the predicted inertial delta observations, calculated in equations 4.12, 4.13 and 4.14, change with respect to a change in the estimated states used to compute them. This matrix is created by taking the derivatives of equations 4.12, 4.13 and 4.14 with respect to the estimated states. These derivatives are shown in table 4.2.

The equivalent derivatives for equation 4.14 are not shown as, due to the discontinuities and multiple solutions of the rotation matrix to Euler angle algorithm, a closed form of these derivatives does not exist. In practice derivatives for this equation can be found numerically or analytically for each piecewise continuous section.

Where $\alpha = \frac{d\hat{C}_{bt1}^{n-1}}{d\phi_{t1}} (\hat{p}_{t2} - \hat{p}_{t1} - \hat{v}_{t1}\Delta t - \frac{1}{2}\hat{g}\Delta t^2)$ and $\beta = \frac{d\hat{C}_{bt1}^{n-1}}{d\phi_{t1}} (\hat{v}_{t2} - \hat{v}_{t1} - \hat{g}\Delta t)$

4.4 Importance of the Δp^+ component

The Δp^+ component of the inertial delta observation is important especially when the integration of the delta is performed over long periods of time.

Table 4.2: Derivatives of the inertial delta observation prediction equations with respect to estimated states

	p_{t1}	v_{t1}	ϕ_{t1}	p_{t2}	v_{t2}	ϕ_{t2}	$bias_f^{obs}$	$bias_\omega^{obs}$	g^n
$h_{\Delta p^+}(\hat{x}_t)$	$-\hat{C}_{bt1}^{n-1}$	$-\hat{C}_{bt1}^{n-1}\Delta t$	α	\hat{C}_{bt1}^{n-1}	0	0	$\frac{d\Delta p_t^+}{dbias_f}$	$\frac{d\Delta p_t^+}{dbias_\omega}$	$-\frac{1}{2}\Delta t^2$
$h_{\Delta v}(\hat{x}_t)$	0	$-\hat{C}_{bt1}^{n-1}$	β	0	\hat{C}_{bt1}^{n-1}	0	$\frac{d\Delta v_t}{dbias_f}$	$\frac{d\Delta v_t}{dbias_\omega}$	$-\Delta t$

In some inertial measurement units a number of observations are integrated and processed to account for coning and sculling. These processed results are then made available to the user as delta observations in velocity and orientation. These integrations are usually preformed over short periods of time, for example 10ms, and a constant velocity, constant acceleration, or similar assumption has to be made over that period to be able to update the position estimate of the platform.

The requirement for this assumption to be made is usually not a problem if the platform has slow dynamics such that the constant acceleration assumption over this period of time is valid, such as for an aircraft. However if longer integration times are used, such as in this thesis, or the application is on a platform with higher dynamics, such as a human, this assumption can cause problems.

The Δp^+ component is calculated and used in the estimation process to overcome these problems. It is simply a corrective factor that can be applied to what is effectively a constant velocity assumption to account for the acceleration profile experienced by the platform over the delta observation integration interval. It can be thought of as an observation connecting the average velocity over the integration interval with the instantaneous velocity at the beginning of the interval. This is important as inertial integration operates using instantaneous velocities but many sensors, such as stereo cameras, can only observe changes in position and therefore average velocities.

To analyse the benefit of the Δp^+ component a simplified 1-dimensional model where there is no rotation or gravity will be considered. The difference in the

position of the platform, Δp over a single integration interval using a constant velocity, constant acceleration and Δp^+ model will be used.

The equations expressing these models can be seen in table 4.3.

Table 4.3: Position estimation models considered for comparison

Δp equation	model
$\Delta p = v_1 dt$	constant velocity
$\Delta p = \frac{v_1 + v_2}{2} dt$	constant acceleration
$\Delta p = v_1 dt + \Delta p^+$	Δp^+

For this example an integration period of 0.25 seconds will be used with an IMU sample rate of 600Hz. Four different cases will be considered, these will be constant velocity, constant acceleration, increasing acceleration and decreasing acceleration.

To make the four cases comparable, all the cases involving acceleration will have an average acceleration of $5ms^{-2}$ and the distance between the two poses will be 0.25m. As a result the initial velocity in the four cases will have to be adjusted to satisfy these constraints. The profile of the increasing and decreasing acceleration follows a cosine curve with a period of 1.25 seconds.

Table 4.4 shows a summary of the important attributes of the four acceleration cases considered and figure 4.4 shows the acceleration profiles of these four cases. Figure 4.5 shows the position profiles for these cases as well.

Table 4.4: Summary of the four acceleration cases considered

acceleration case	initial v ms^{-1}	final v ms^{-1}	average acc^n ms^{-2}	Δv ms^{-1}	Δp m	Δp^+ m
constant velocity	1	1	0	0	0.25	0
constant acceleration	0.379	1.629	5	1.25	0.25	0.155
increasing acceleration	0.680	1.930	5	1.25	0.25	0.080
decreasing acceleration	0.072	1.322	5	1.25	0.25	0.232

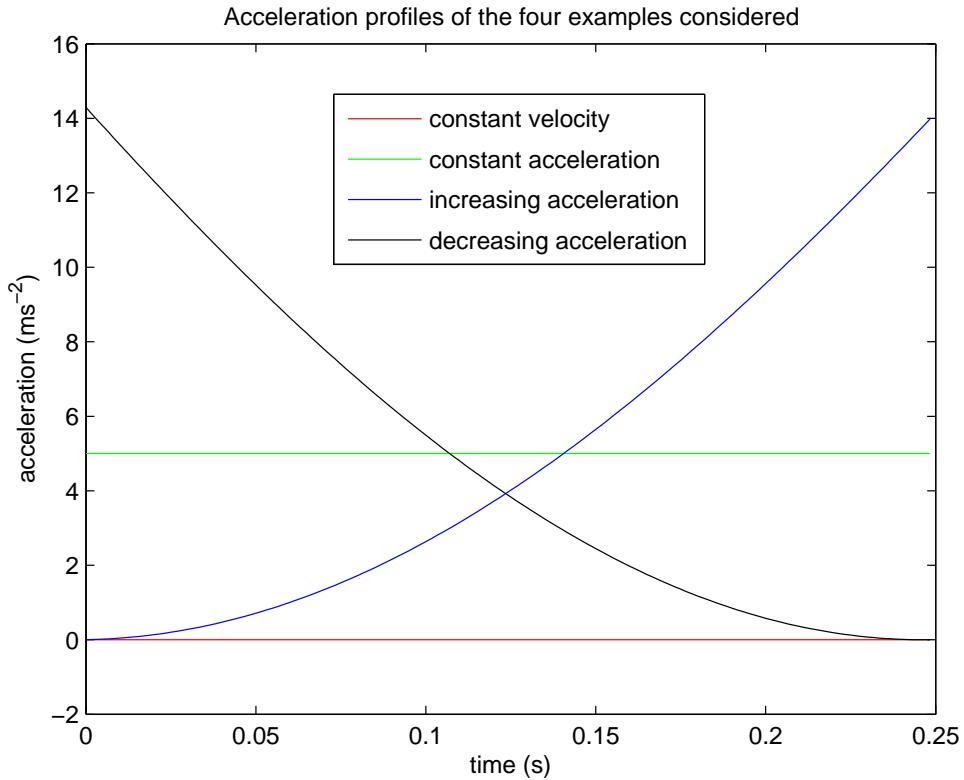


Figure 4.4: Acceleration profiles with respect to time for the four cases considered when analysing the effect of the Δp^+ component when compared to constant velocity and constant acceleration methods.

Note in table 4.4 that the change in position (Δp) is the same for all four cases considered, this can also be seen in figure 4.5. For the three cases with acceleration, the average acceleration and therefore the total change in velocity (Δv) are also the same. However the Δp^+ values are different for all four cases which is expected as this component of the inertial delta observation describes the shape of the acceleration profile within the integration period.

Table 4.5 show a comparison of the estimated final position of the simulated platform for the four acceleration cases described using the inertial delta observations and the three estimation models shown in table 4.3.

It can be seen from this table that for the constant velocity profile all three models successfully estimate the final position of $0.25m$. For the constant accel-

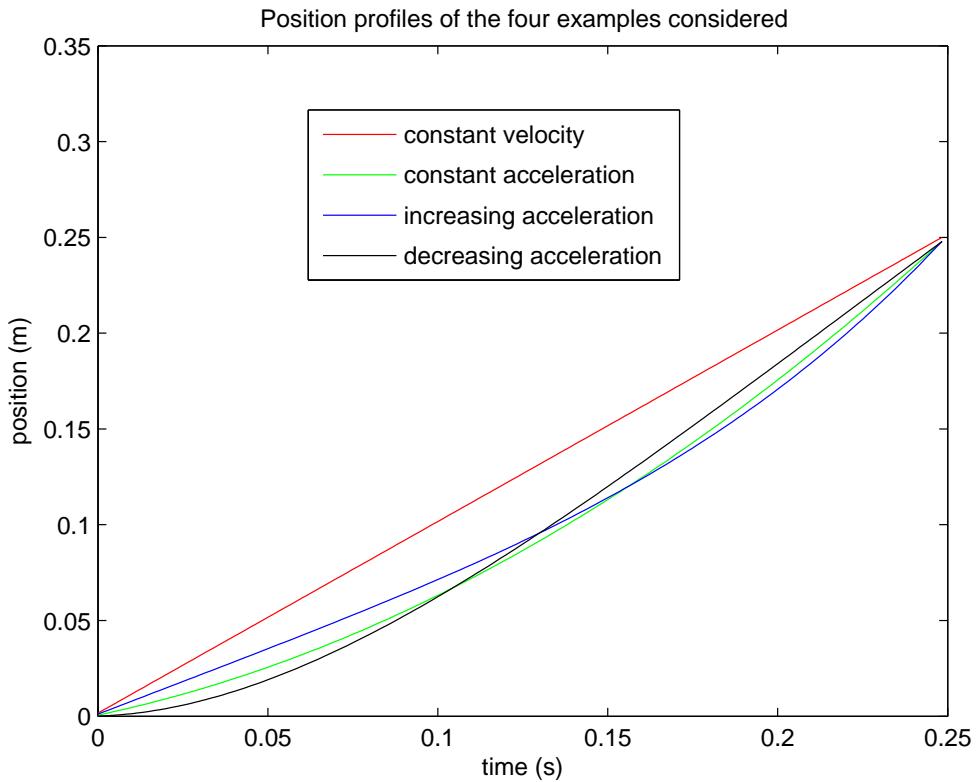


Figure 4.5: Position profiles with respect to time for the four cases considered when analysing the effect of the Δp^+ component when compared to constant velocity and constant acceleration methods.

eration case the constant velocity model under-estimates the displacement of the platform as $0.0948m$ while the other two models obtain the correct answer.

Interesting results are obtained with the non-constant acceleration profiles where the constant velocity model under-estimates the displacement every time. The constant acceleration model over-estimates the platform displacement when increasing acceleration is experienced, and under-estimates the displacement when decreasing acceleration is experienced.

In all four acceleration profiles examined the Δp^+ estimation model obtains the correct result.

Table 4.5: Final position results of the four acceleration cases considered using the three estimation models shown in table 4.3. True final position value is $0.25m$ for every case

acceleration case	constant velocity model (m)	constant acceleration model (m)	Δp^+ model (m)
constant velocity	0.250	0.250	0.250
constant acceleration	0.0948	0.250	0.250
increasing acceleration	0.170	0.3262	0.250
decreasing acceleration	0.0179	0.1741	0.250

4.5 Chapter summary

This chapter discussed the challenges faced when trying to use linear estimators for inertial navigation due to the inherent non-linearities in inertial integration equations and the reliance of those equations on knowing the initial conditions of the vehicle. A re-parameterisation of the inertial equations into a body frame fixed coordinate system was presented. This re-parameterisation is shown to allow the IMU observations to be integrated without the need for initial conditions.

Given this ability to integrate inertial observations without initial conditions, a novel method for processing and fusing them into SLAM implementations was presented. The new method allows grouping of inertial observations between required poses and, more importantly, makes it possible for inertial observations to be used in SLAM where the initial conditions are not known and allows them to be recovered in a linear way.

The improvements in linearity and the removal of the need for any initialisation stage for the inertial measurement unit gives this algorithm many advantages over conventional inertial SLAM algorithms.

A step by step comparison of the new integration equations to the standard inertial SLAM equations is presented as the algorithm is developed. A detailed explanation of how to preform the calculations and the algorithms behind a practical implementation of the pre-integrated inertial delta observations is also provided.

Finally, The role that the Δp^+ component of an inertial delta observation plays was examined and how it provides a link between the instantaneous and average velocity of the platform. Comparisons between the performance of a constant velocity model, constant acceleration model and a model using Δp^+ components under various acceleration profiles was examined.

Chapter 5

Analysis of pre-integrated inertial delta observations

In chapter 4 a new method for processing and using observations from an IMU was developed. This technique allows inertial information to be integrated and used before the initial conditions of the platform are known. These initial conditions can then be recovered in a linear way.

This chapter analyses the performance of this new technique to assess the benefits of the increased linearity and how the initial conditions can be recovered in a linear way. Comparison to the performance of standard inertial integration techniques will also be conducted.

5.1 Initial condition recovery

5.1.1 Initial velocity estimation

In section 4.3 an algorithm was developed that allowed inertial observations to be pre-integrated without an estimate of initial velocity. In some situations however, such as in control applications, an estimate of the initial velocity of the platform may be desirable.

By rearranging equation 4.9 it is possible to show that the initial velocity

of the platform is observable give two consecutive relative position estimates, a gravity vector estimate and the inertial observations between them. This is shown in equation 5.1.

$$v_{t1}^n = \frac{p_{t1}^n - p_{t2}^n + C_{bt1}^n \Delta p_{t2}^{+t1} + \frac{1}{2}(t2 - t1)^2 g^n}{(t2 - t1)} \quad (5.1)$$

Therefore an estimate of the initial velocity of the platform can be obtained in a linear way without a prior estimate by using the inertial pre-integration technique.

5.1.2 Gravity vector estimation

One criterion for the initial velocity observability shown in section 5.1.1 is that an estimate of the gravity vector be available. For the inertial pre-integration process to be truly free of initialisation requirements the gravity vector should be observable as well.

In order for both the initial velocity and the gravity vector to be estimated, aiding relative position observations from at least three different poses are required.

Taking equation 4.9 over two consecutive inertial delta observations and the velocity equation from the first delta observations from equation 4.10 results in the set of equations 5.2.

$$\begin{aligned} p_{t2}^n &= p_{t1}^n + (t2 - t1)v_{t1}^n + C_{bt1}^n \Delta p_{t2}^{+t1} + \frac{1}{2}(t2 - t1)^2 g^n \\ p_{t3}^n &= p_{t2}^n + (t3 - t2)v_{t2}^n + C_{bt2}^n \Delta p_{t3}^{+t2} + \frac{1}{2}(t3 - t2)^2 g^n \\ v_{t2}^n &= v_{t1}^n + C_{bt1}^n \Delta v_{t2}^{t1} + (t2 - t1)g^n \end{aligned} \quad (5.2)$$

Note that in equations 5.2 the gravity vector g^n is defined in the navigation frame. This frame is usually defined as the body frame of the first pose being estimated which may or may not be the pose at time $t1$. Therefore the navigation

frame notation is used, instead of body frame notation, to maintain generality in the equations however these can be considered to be equivalent in this example.

Substituting the velocity equation into the second position equation makes equation 5.3.

$$\begin{aligned}
 p_{t3}^n &= p_{t2}^n + (t3 - t2) (v_{t1}^n + C_{bt1}^n \Delta v_{t2}^{t1} + (t2 - t1) g^n) \\
 &\quad + C_{bt2}^n \Delta p_{t3}^{+t2} + \frac{1}{2} (t3 - t2)^2 g^n \\
 &= p_{t2}^n + (t3 - t2) (v_{t1}^n + C_{bt1}^n \Delta v_{t2}^{t1}) + C_{bt2}^n \Delta p_{t3}^{+t2} \\
 &\quad + (t3 - t2) \left(\frac{1}{2} (t3 - t2) + (t2 - t1) \right) g^n
 \end{aligned} \tag{5.3}$$

Rearranging the first velocity equation into the form of equations 5.1 and substituting into equation 5.3 makes equation 5.4.

$$\begin{aligned}
 p_{t3}^n &= p_{t2}^n + (t3 - t2) \left(\left[\frac{p_{t1}^n - p_{t2}^n + C_{bt1}^n \Delta p_{t2}^{+t1} + \frac{1}{2} (t2 - t1)^2 g^n}{(t2 - t1)} \right] + C_{bt1}^n \Delta v_{t2}^{t1} \right) \\
 &\quad + C_{bt2}^n \Delta p_{t3}^{+t2} + (t3 - t2) \left(\frac{1}{2} (t3 - t2) + (t2 - t1) \right) g^n
 \end{aligned} \tag{5.4}$$

This can be rearranged into an expression for the gravity vector containing only the relative position estimates and inertial delta observations as terms as shown in equation 5.5

$$g^n = \frac{p_{t2}^n - p_{t3}^n + (t3 - t2) \left(\left[\frac{p_{t1}^n - p_{t2}^n + C_{bt1}^n \Delta p_{t2}^{+t1} + \frac{1}{2} (t2 - t1)^2 g^n}{(t2 - t1)} \right] + C_{bt1}^n \Delta v_{t2}^{t1} \right) + C_{bt2}^n \Delta p_{t3}^{+t2}}{(t3 - t2) \left(\frac{1}{2} (t3 - t2) + (t2 - t1) \right)} \tag{5.5}$$

Therefore with inertial pre-integration it is possible to obtain estimates for both the initial velocity of the platform and the gravity vector after just three relative position observations in a linear way without the need for initial conditions.

A beneficial side effect of gravity vector observability is that with this estimate

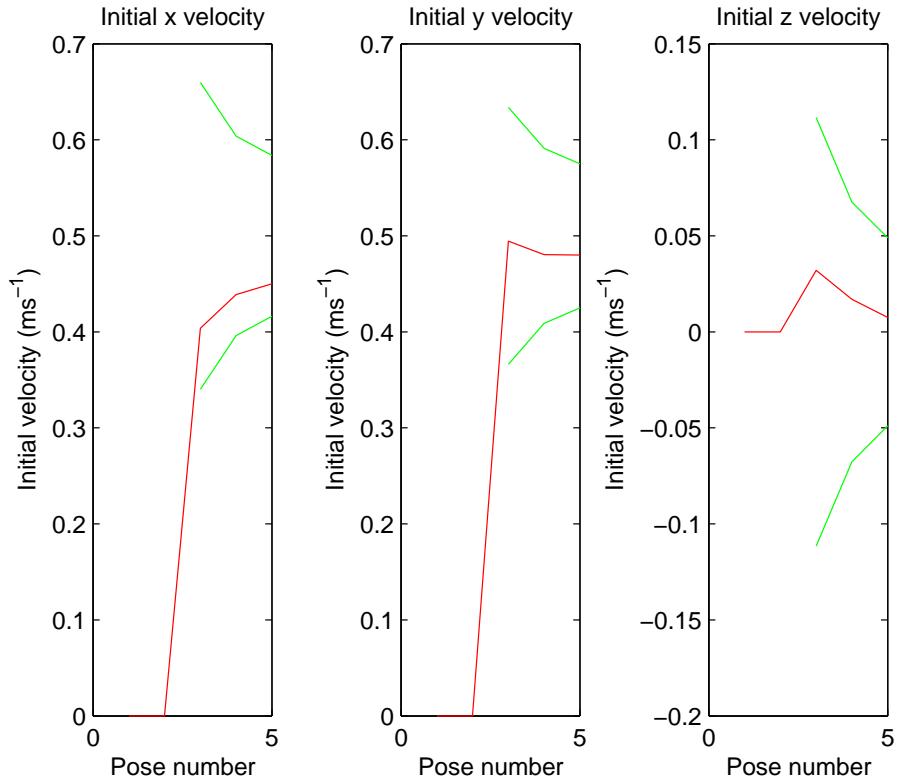


Figure 5.1: Sample estimate convergence over the first 5 poses for the initial velocity estimate. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.

the absolute roll and pitch of the platform in the inertial frame can be extracted as well.

5.1.3 Monte-Carlo simulation for initial conditions

In order to test the observability of the initial velocity and gravity vector when using pre-integrated inertial delta observations in a visual SLAM implementation, a Monte-Carlo test was performed using simulated data. The simulation set-up used is described in appendix A.

1000 simulations were performed with the same trajectory but different random landmark placements, observation noises and IMU biases. Five consecutive

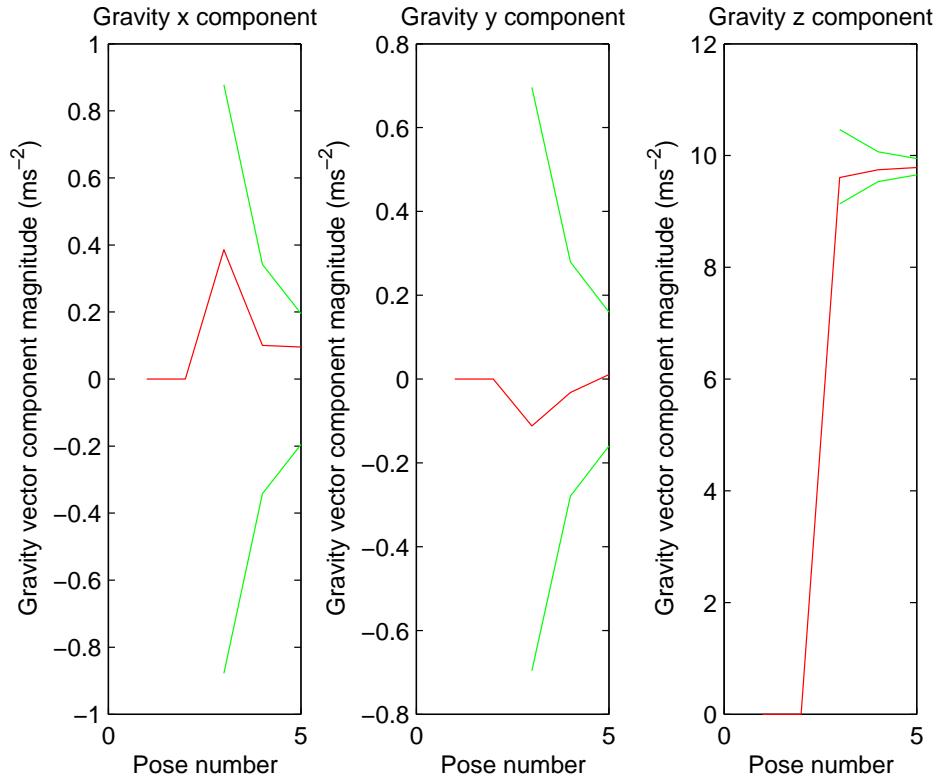


Figure 5.2: Sample estimate convergence over the first 5 poses for the gravity vector estimate. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.

poses were simulated which uses data collected over less than a second so that the use of pre-integrated inertial delta observations to provide fast initialisation for control can be analysed. Since biases also need to be estimated in this simulation more than three poses are required to obtain an accurate estimate of the initial conditions, which is why five poses are used instead of just three.

An uninformed all zero prior for the initial velocity and gravity vector estimates as well as bias estimates is used so that the estimated means are completely a product of the observations made. Estimation is performed in the first pose body frame which is afterwards rotated into the navigation frame to make analysis easier as the gravity vector and initial velocity provided by the simulation

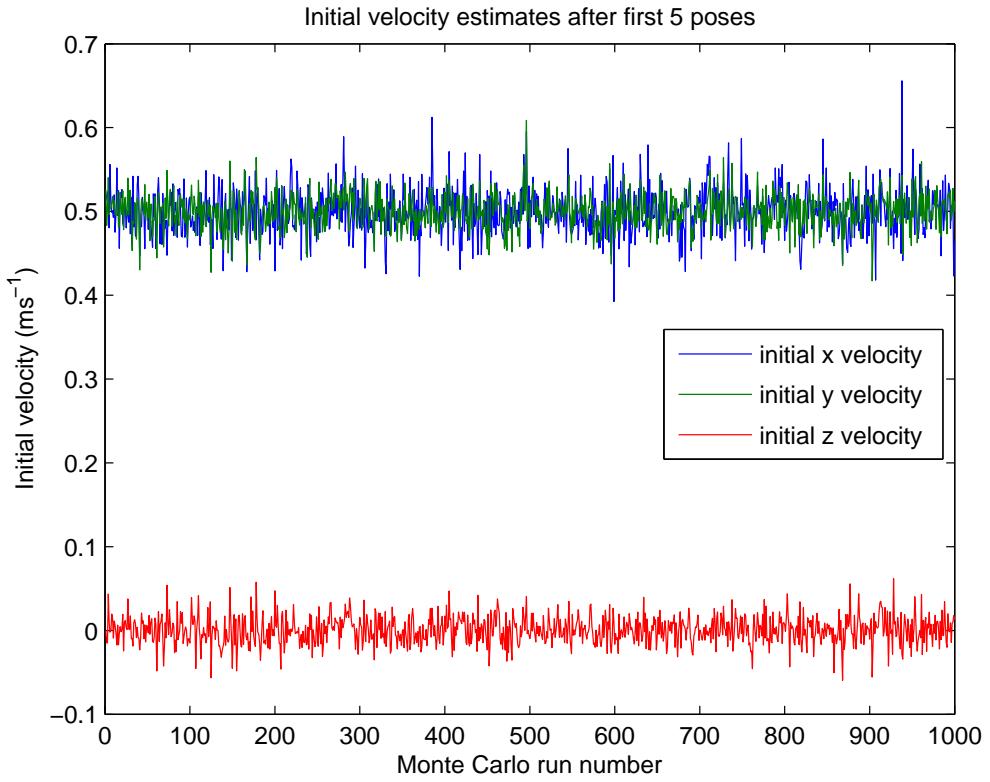


Figure 5.3: Monte Carlo results for the initial velocity estimates after the first 5 poses over 1000 runs. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.

trajectory generator are provided in the navigation frame. This rotation would normally not be necessary for a practical implementation.

Figures 5.1 and 5.2 show a sample of the convergence of the estimated initial velocity and gravity vector respectively with 2σ uncertainty bounds. The initial estimates start as all zero. Uncertainty bounds for the estimates are not available until the third pose as a minimum of three poses are required for the initial velocity and gravity vector to be jointly observable as discussed in section 5.1.2.

Figures 5.3 and 5.4 show the estimated initial conditions for the 1000 simulated runs. The true simulated values for the initial velocity and gravity vector along with the mean and variance of the 1000 estimated initial values can be seen in

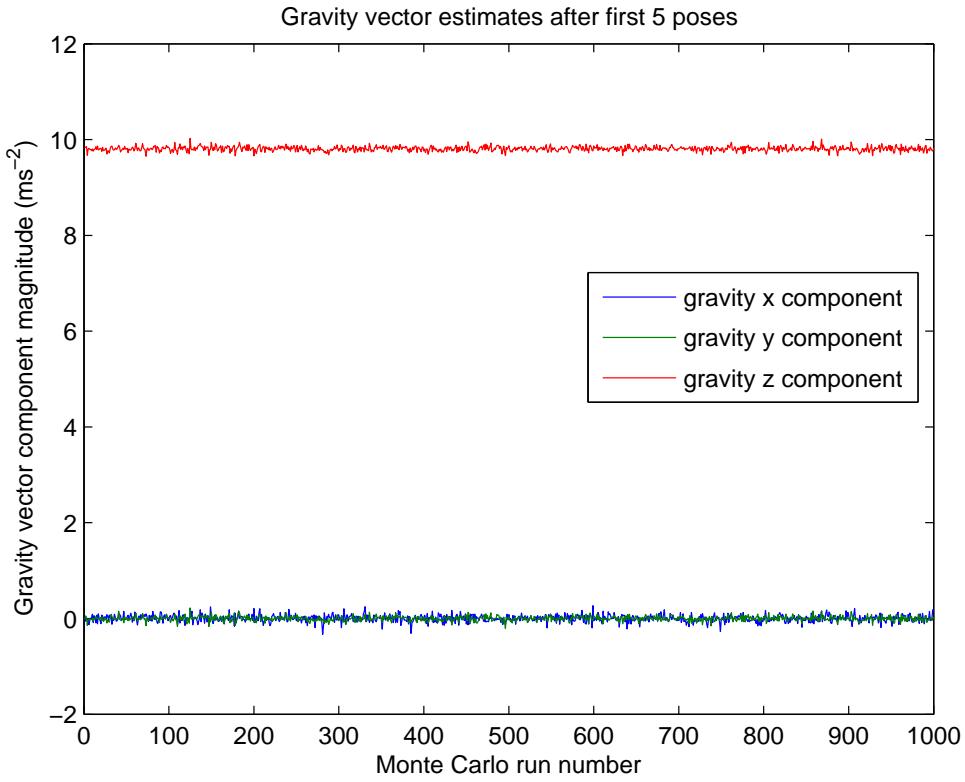


Figure 5.4: Monte Carlo results for the gravity vector estimates after the first 5 poses over 1000 runs. Results have been transformed back into the absolute reference frame for comparison in tables 5.1 and 5.2, estimation is done in the first pose relative frame.

table 5.1.

From inverting the information matrix produced by the information filter, the estimated variance of the initial velocity and gravity vector estimates for each run can be found. These variances are shown in table 5.2 along with the standard deviation divided by the square root of the number of Monte Carlo runs and the error in the mean values produced by those runs. This information can be used to assess if the filter is performing as expected.

The estimated uncertainty from the filter, $\hat{\sigma}_\mu^2$, shown in table 5.2 is taken from one sample run. The estimated uncertainty can change from run to run as the landmarks used are randomly placed and are different in each of the Monte Carlo

Table 5.1: Initial condition results from Monte Carlo runs on first 5 poses. Mean estimate and variance of means calculated from the data presented in figures 5.3 and 5.4

State	Truth	Mean estimate	Variance of means
	μ	$\bar{\mu}$	$\sigma_{\bar{\mu}}^2$
Initial x velocity (ms^{-1})	0.4999999	0.5012808	0.0007899
Initial y velocity (ms^{-1})	0.4997916	0.5006600	0.0004331
Initial z velocity (ms^{-1})	-0.0002083	0.0008143	0.0002835
x gravity (ms^{-2})	0.0000000	-0.0023072	0.0054858
y gravity (ms^{-2})	0.0000000	-0.0015679	0.0025088
z gravity (ms^{-2})	9.8100000	9.8080067	0.0025320

Table 5.2: Filter estimates of the variance of the initial velocity and gravity vector estimates after 5 image poses

State	$\hat{\sigma}_{\mu}^2$	$\hat{\sigma}_{\mu}/\sqrt{(n)}$	$\hat{\mu} - \mu$
Initial x velocity (ms^{-1})	0.0013395	0.0011574	0.0012809
Initial y velocity (ms^{-1})	0.0003667	0.0006055	0.0008683
Initial z velocity (ms^{-1})	0.0004272	0.0006536	0.0010227
x gravity (ms^{-2})	0.0088063	0.0029675	-0.0023072
y gravity (ms^{-2})	0.0027589	0.0016610	-0.0015679
z gravity (ms^{-2})	0.0034277	0.0018514	-0.0019933

runs performed leading to a different distribution of visual observations in each run. This is why the estimated variance for this one particular run may differ from the Monte Carlo variance of the means for the 1000 runs presented in table 5.1.

From table 5.2 it can be seen that after only five poses the components of initial velocity are estimated with an expected standard deviation of less than $0.04ms^{-1}$ ($\sigma^2 < 0.0016$). This value is consistent with the Monte-Carlo simulation variance of the initial velocity estimates shown in table 5.1.

Similarly the gravity component estimates are known to an estimated standard deviation of less than $0.1ms^{-2}$ ($\sigma^2 < 0.01$). This is also consistent with the Monte-Carlo simulation values in table 5.1.

The estimated mean standard deviation for the Monte-Carlo runs ($(\hat{\sigma}_\mu / \sqrt{n})$) gives the expected standard deviation of the mean estimate for the initial conditions taken over all the 1000 simulated runs. This value can be compared to the Monte-Carlo mean error ($\hat{\mu} - \mu$) to assess if the estimated initial conditions are unbiased. It can be seen from table 5.2 that all the mean errors are within two standard deviations as expected. This result suggests that the estimates are unbiased.

5.2 Remaining non-linearity

Even though pre-integrating the inertial observations in the body frame results in highly linear observation equations, there are still two small remaining sources of non-linearity. The effect of uncertainty in the gyro bias estimates is one of these sources, the other is the drift caused from the integration of the gyro sensor noise.

5.2.1 Non-linearity from bias correction

As can be seen in equation 4.8 the gyro bias terms change the intermediate delta attitude estimate and therefore the C_t^{t1} and E_t^{t1} matrices used to perform the inertial pre-integration in algorithm 4.1. This not only affects the $\Delta\phi$ term of the pre-integrated inertial observation, but also the Δp^+ and Δv terms as well through the intermediate C_t^{t1} matrix.

If the bias is known at the time the inertial observations are pre-integrated they can be accounted for as shown in equations 4.6, 4.7 and 4.8 as well as in algorithm 4.1.

After inertial pre-integration, if a more accurate estimate of the biases is obtained, a correction can be made to the pre-integrated delta observations to account for this as part of equations 4.12, 4.13 and 4.14.

In order to make this correction the $\frac{d\Delta}{dbias}$ terms calculated in algorithm 4.3 (and shown in equation 4.15) must be used. These terms keep track of how the

biases affect the values of the pre-integrated delta observation.

For accelerometer biases this effect is linear, however the gyro biases as well as attitude estimate drift during the integration period affect the delta observations in a non-linear way. The $\frac{d\Delta}{dbias}$ term only allows for a linear correction using a small angle approximation so there is a small remaining non-linearity with respect to accelerometer biases as well. Given that residual gyro biases are generally small, or can be made small through estimation, and that the attitude drift due to gyro noise over the integration interval is small, a small angle approximation to the error can be made over the pre-integration period. Therefore it is assumed that this non-linearity will in practice not cause a problem.

5.2.2 Monte-Carlo simulation of non-linearity from bias correction

To test the theory that non-linearity from gyro bias correction is not a practical problem a Monte-Carlo simulation for pre-integrating inertial observations and correcting for biases after pre-integration was performed to compare them to the result when biases were corrected for before pre-integration.

100 simulations were run over a 100 second period with the simulation set-up described in appendix A. Random accelerometer and gyro biases were applied to each run. Integration was performed with inertial observations only and no other sensor observations were used so that the effect of biases could be isolated.

The simulated inertial observations were processed in three ways. Firstly the inertial observations were pre-integrated to form inertial delta observations with the true bias values used in algorithm 4.1, this provided the optimal solution. Secondly the inertial observations were pre-integrated with a bias estimate of 0 for all values, this will show the result of integrating the inertial observations without bias compensation. Finally the uncompensated inertial delta observations were corrected after pre-integration using the true bias values and the calculated $\frac{d\Delta}{dbias}$ terms.

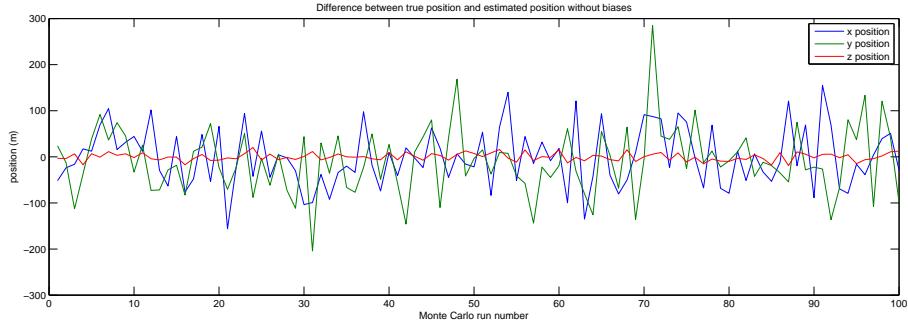


Figure 5.5: Difference in estimated and true position after navigating using pre-integrated inertial delta observations without bias compensation for a period of 100 seconds.

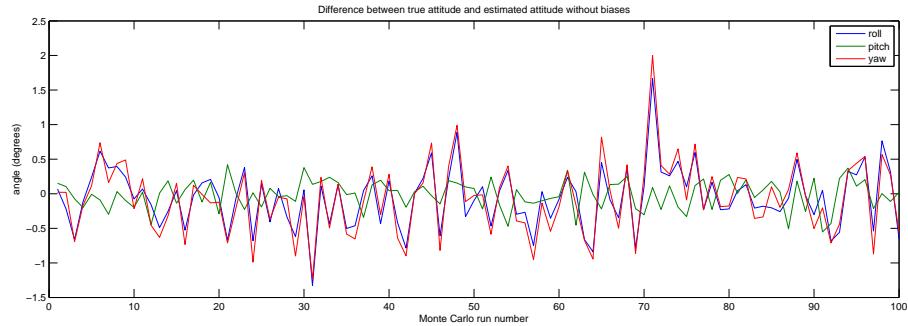


Figure 5.6: Difference in estimated and true attitude after navigating using pre-integrated inertial delta observations without bias compensation for a period of 100 seconds.

The optimal solution calculated first is used to compare the performance of the other two solutions. Figures 5.5 and 5.6 show the difference between the final position and attitude estimates respectively from integrating the inertial observations without bias estimates to the optimal value using the correct biases.

Note how in figure 5.5 the error in the z position is generally of a smaller magnitude than the errors in the x and y positions. This is due to the effect of compensating for the gravity vector using an incorrect attitude estimate because the gyro biases are not compensated for.

Figures 5.7 and 5.8 show the difference between the true and estimated position and attitude estimates after the pre-integrated observations have been corrected for biases. It can be seen when comparing these to the previous figures

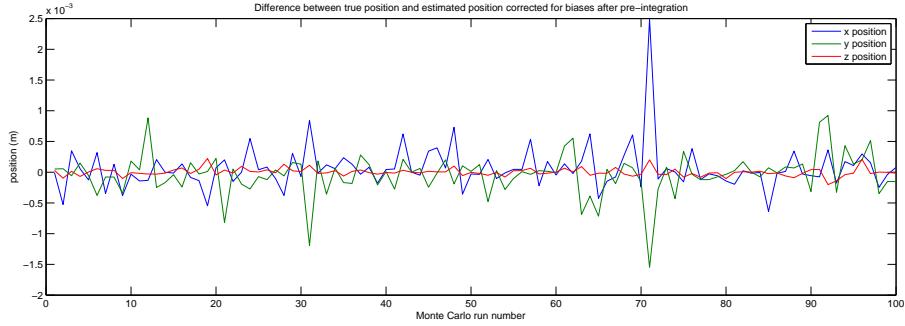


Figure 5.7: Difference in estimated and true position after navigating using pre-integrated inertial delta observations corrected for biases after pre-integration is performed for a period of 100 seconds.

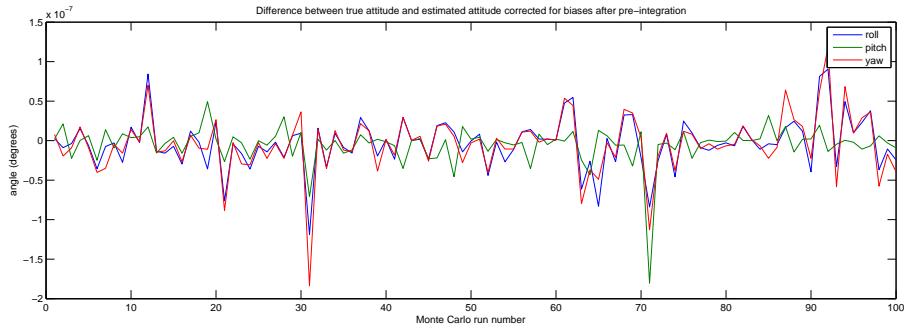


Figure 5.8: Difference in estimated and true attitude after navigating using pre-integrated inertial delta observations corrected for biases after pre-integration is performed for a period of 100 seconds.

that the position difference is reduced by approximately a factor of 200,000 and the attitude difference by an approximate factor of $2e^7$.

From these very large difference factors it can be concluded that while the non-linearity of correcting for biases after inertial pre-integration exists, compared to the error from the bias itself, it is insignificant.

5.2.3 Non-linearity from inertial drift

Even if biases are known with complete certainty there is one remaining source of non-linearity when using pre-integrated inertial delta observations.

This non-linearity comes from the slow accumulation of uncertainty in the

attitude estimate within the inertial delta observation from the integration of the gyro noise. This uncertainty in attitude also couples with the accelerometer noise to produce a slight non-linearity in the acceleration estimate as well.

The predominant effect is in the delta attitude uncertainty, the magnitude of which can be seen from the delta observations uncertainty matrix (the R matrix from algorithm 4.3). It has a similar effect on the accuracy and consistency of the final estimate as a small angle approximation of similar magnitude.

This effect will be small in most cases and will never be worse than the linearisation error accumulated from a non-relinearising filter, such as any state space based or non-delayed state filters, between any two successive poses.

If relinearising is performed when fusing the inertial delta observations then significant improvements over the accumulated linearisations errors of standard integration can be achieved as linearisation errors will only accumulate in the small intervals between poses and be reset at each pose in the chain when fused.

5.3 Comparison to standard inertial integration

5.3.1 Inertial only navigation

To compare the results obtained from using pre-integrated inertial delta observations compared to the standard inertial integration techniques such as those used in [9], Monte-Carlo simulations were performed.

Simulation data was generated with the set-up explained in appendix A. To test the result of inertial integration alone, only inertial observations were used with no other aiding observations. Since standard inertial integration requires initial conditions, they were provided to both the standard and the pre-integrated delta observation cases to make for a fair comparison.

1000 runs were simulated with different IMU sensor noises and random biases for each run over a 13 second period. Table 5.3 shows the mean and variance of the final pose estimate from the 1000 runs along with the estimated variance

Table 5.3: Mean and variance of the results of 1000 Monte Carlo runs for the pre-integrated inertial delta technique

State	mean	$\bar{\sigma}_{\Delta}^2$	estimated variance $\hat{\sigma}_{\Delta}^2$
	$\bar{\mu}_{\Delta}$		
x position (m)	0.11463	0.00717	0.00699
y position (m)	0.10665	0.00691	0.00699
z position (m)	-0.01057	0.00643	0.00699
roll (rad)	0.01166	$2.2146e^{-8}$	$2.1333e^{-8}$
pitch (rad)	-0.00068	$2.0372e^{-8}$	$2.1333e^{-8}$
yaw (rad)	0.09023	$2.1822e^{-8}$	$2.1333e^{-8}$

Table 5.4: Mean and variance of the results of 1000 Monte Carlo runs for the standard inertial integration technique

State	mean	$\bar{\sigma}_{std}^2$	estimated variance $\hat{\sigma}_{std}^2$
	$\bar{\mu}_{std}$		
x position (m)	0.11462	0.00717	0.00699
y position (m)	0.10665	0.00691	0.00696
z position (m)	-0.01057	0.00643	0.00703
roll (rad)	0.01166	$2.2146e^{-8}$	$2.1654e^{-8}$
pitch (rad)	-0.00068	$2.0372e^{-8}$	$2.1333e^{-8}$
yaw (rad)	0.09023	$2.1821e^{-8}$	$2.1654e^{-8}$

for these pose states taken from the diagonal of the filter covariance matrix for the pre-integrated inertial delta observation case. The same results for the case where standard inertial integration was performed is shown in figure 5.4.

Comparison of these two tables shows that the results for the state estimates are almost identical (the difference in the means of the estimates for these two cases can be seen in table 5.5). The true variances of the means is also similar, however there is a slight difference in the estimated variance.

The reason for this difference is due to the use of Euler angles as the attitude states. Since the Euler angle representation has a singularity at $\pm 90^\circ$ pitch, this results in large derivatives calculated around this region. The result of which is a skewing in the Euler rate matrix E_b^n for large pitch angles. This affects

Table 5.5: Analysis of the Monte-Carlo results for the pre-integrated inertial delta technique shown in table 5.3

State	$\hat{\sigma}_\Delta^2$	$\hat{\sigma}_\Delta / \sqrt{n}$	$\bar{\mu}_\Delta - \mu_{true}$	$\bar{\mu}_\Delta - \bar{\mu}_{std}$
x position (m)	0.00699	$2.6439e^{-3}$	$1.9221e^{-3}$	$7.1520e^{-6}$
y position (m)	0.00699	$2.6439e^{-3}$	$3.0857e^{-3}$	$4.0148e^{-7}$
z position (m)	0.00699	$2.6439e^{-3}$	$3.7499e^{-3}$	$1.3192e^{-6}$
roll (rad)	$2.1333e^{-8}$	$4.6188e^{-6}$	$-7.7562e^{-6}$	$9.8368e^{-7}$
pitch (rad)	$2.1333e^{-8}$	$4.6188e^{-6}$	$-2.4868e^{-6}$	$3.5771e^{-7}$
yaw (rad)	$2.1333e^{-8}$	$4.6188e^{-6}$	$-1.8388e^{-6}$	$-1.0964e^{-6}$

the attitude covariance calculation (through the E_t^{t1} term in the G_t matrix in algorithm 4.4) when large pitch angles are present. These large derivatives result in a skewing of the estimated attitude covariance which then flows on to the position and velocity covariance when accelerometer observations are integrated.

The reason why this effect is not present in the pre-integrated inertial delta observation case is that covariance calculations are performed around the zero roll, pitch and yaw attitude, as are the state mean predictions, as can be seen in algorithm 4.3. This is why the estimated variance for each of the attitude and position components are almost identical for the inertial delta case but slightly different for the standard inertial integration case.

Table 5.5 shows an analysis of the error between the true final pose states and the mean of this value for the pre-integrated inertial delta observation case. It can be seen that the error in the mean is within two of the expected standard deviations ($\hat{\sigma}_\Delta / \sqrt{n}$), suggesting that the results of the Monte-Carlo simulations is consistent with the estimated variance.

The last column of table 5.5 shows the difference in the means of the final pose states for the pre-integrated and standard inertial cases. This column shows that this difference is orders of magnitude lower than the difference from the mean for the position states, the small differences are likely due to accumulated numerical rounding error as the order of operations is different for the two cases. This suggests that both methods are obtaining approximately the same result.

5.3.2 Aided inertial SLAM

As well as comparing the performance of pre-integration inertial delta observations to standard inertial navigation techniques, aided navigation with the use of range and bearing observations to landmarks in a SLAM situation was also simulated through the use of stereo camera observations.

The same Monte-Carlo simulation as for section 5.3.1 were conducted only this time stereo camera observations of 100 landmarks randomly placed within 5 metres of the origin at a rate of 6.25Hz were also included. The camera base line was set to 12cm with a field of view and observation uncertainty set to match that of the BumbleBee2 stereo camera described in table 8.2. The IMU simulation was also set up to give similar performance to the HG1900 IMU described in table 8.1.

An EKF implementation was used for these simulations even though it is not the preferred filter structure for application in this thesis. The reasons why it was used is that the standard inertial integration techniques that are used for comparison are traditionally implemented with an EKF [8], also with the high update rate of the IMU (600Hz) placing poses in a graphical smoothing implementation with the standard method would results in a large number of pose nodes (> 7800) making updating the graph and covariance recovery costly.

As the standard technique requires an initial orientation estimate, both filters were initialised at the true orientation. This is sufficient as this simulation was conducted to test the post-initialisation running performance of the two techniques after initial condition estimates have converged.

The results of these simulations for the pre-integrated inertial delta technique and the standard inertial integration technique are shown in tables 5.6 and 5.7 respectively.

The results for aided inertial SLAM with both techniques show similar performance. Both methods produce slightly over confident results, as can be seen by the true variances being larger than the estimated variance, but this is expected from an EKF due to accumulating linearisation errors as the true sensor variances

Table 5.6: Mean and variance of the results of 1000 Monte Carlo runs for the pre-integrated inertial delta technique using range and bearing landmark observations

State	truth	mean	variance	estimated variance
	μ_{true}	$\bar{\mu}_\Delta$	$\bar{\sigma}_\Delta^2$	$\hat{\sigma}_\Delta^2$
x position (m)	0.11655	0.11604	$7.15542e^{-6}$	$4.42798e^{-6}$
y position (m)	0.10973	0.10918	$4.90649e^{-6}$	$2.60041e^{-6}$
z position (m)	-0.00681	-0.00690	$4.43680e^{-6}$	$2.53093e^{-6}$
roll (rad)	0.01165	0.01165	$2.07523e^{-8}$	$1.89906e^{-8}$
pitch (rad)	-0.00068	-0.00068	$2.36031e^{-8}$	$1.87991e^{-8}$
yaw (rad)	0.09023	0.09022	$2.14574e^{-8}$	$1.88455e^{-8}$

Table 5.7: Mean and variance of the results of 1000 Monte Carlo runs for the standard inertial technique using range and bearing landmark observations

State	truth	mean	variance	estimated variance
	μ_{true}	$\bar{\mu}_{std}$	$\bar{\sigma}_{std}^2$	$\hat{\sigma}_{std}^2$
x position (m)	0.11655	0.11623	$6.14861e^{-6}$	$4.43699e^{-6}$
y position (m)	0.10973	0.10922	$4.70222e^{-6}$	$2.60948e^{-6}$
z position (m)	-0.00681	-0.00690	$4.05258e^{-6}$	$2.53346e^{-6}$
roll (rad)	0.01165	0.01165	$2.05766e^{-8}$	$1.91004e^{-8}$
pitch (rad)	-0.00068	-0.00068	$2.29391e^{-8}$	$1.87643e^{-8}$
yaw (rad)	0.09023	0.09023	$2.12838e^{-8}$	$1.90528e^{-8}$

are used and stabilising noise is not added. When compared to the results in tables 5.3 and 5.4 it can be seen that the landmark observations do help to reduce the estimation error, especially the position estimate, which is as expected.

5.3.3 Vision aided inertial initialisation

In order to compare initial condition estimation using pre-integrated inertial delta observations to the batch and large angle methods using over parameterised attitude states, the same simulation set-up as in section 5.1.3 was used. The first 20 poses were simulated and initialisation using the pre-integrated inertial delta technique was compared to a batch initialisation and a large angle initialisation

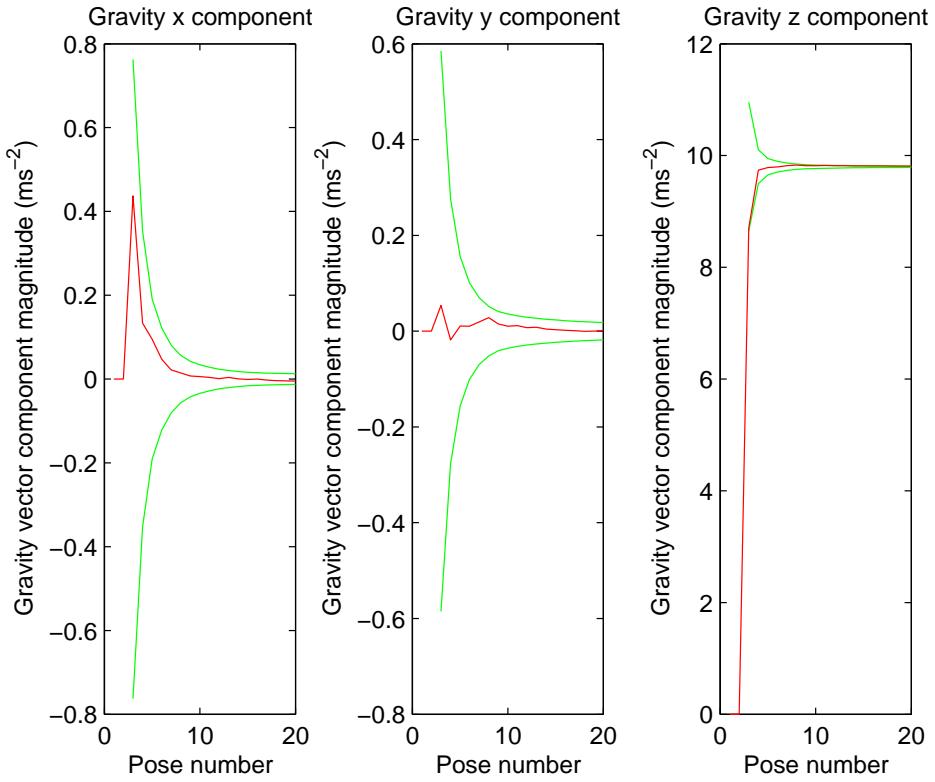


Figure 5.9: Gravity vector estimate convergence using the pre-integrated inertial delta observation technique over the first 20 poses.

which over parameterised the roll and pitch with their sine and cosine values. The yaw value was not over parameterised as it is not observable in this situation and so was fixed at zero for all cases.

As for the inertial delta observation method the attitude is fixed but the gravity vector is unknown, the estimates for the gravity vector and initial velocity over the first 20 poses are shown in figures 5.9 and 5.10 respectively. Initially each of the gravity vector and velocity components are set to zero to indicate a uninformed prior with no indication of the initial orientation with respect to gravity or direction of motion.

It can be seen from these figures that there is a large correction from the initial estimates after the third image is obtained as this is when the initial velocity and gravity vector become observable. After this point the solution settles and the

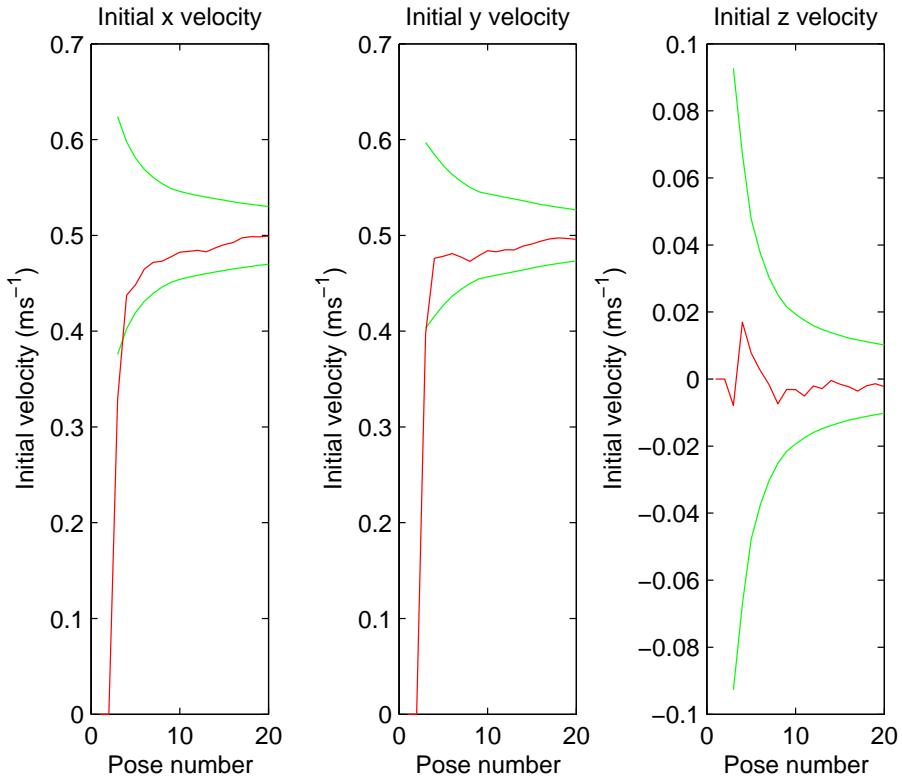


Figure 5.10: Initial velocity estimate convergence using the pre-integrated inertial delta observation technique over the first 20 poses.

uncertainty bounds converge indicating a more accurate estimate.

For the batch initialisation technique the gravity vector is fixed but the initial roll and pitch angles are unknown. For this method a batch solution iterates until it has converged after the first 3 poses are observed and then standard Euler angle estimation is continued from this point. As the batch initialisation requires a starting point for the solution, initial roll and pitch angles of 0, 15, 30, 45 and 60 degrees were tested. The initial velocity prior was set to all zero for each case as was with the inertial delta example.

Figures 5.11 and 5.12 show the convergence of the initial roll and pitch angles estimated for the batch initialisation method for different prior orientations.

It can be seen from these figures that the solution converges to close to the final value after the first 3 poses where the batch initialisation is performed and

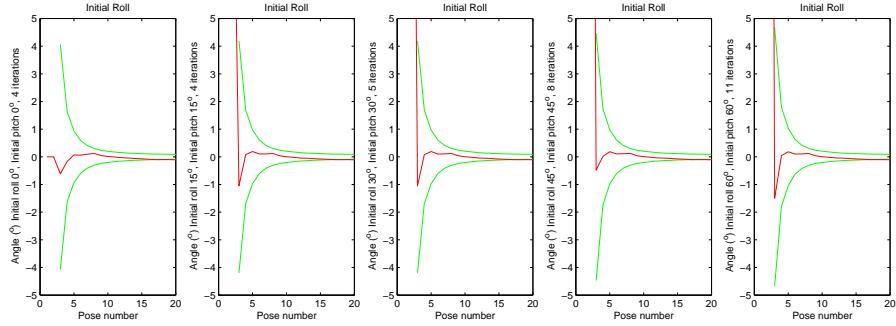


Figure 5.11: Initial roll estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.

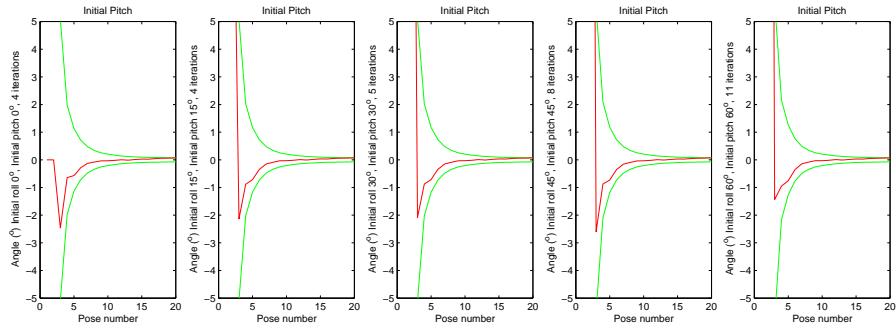


Figure 5.12: Initial pitch estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.

then continues to be slowly refined from that point as more observations become available.

The starting point for the batch solver does not seem to have much effect on the solution obtained which is as expected however the number of iterations required to obtain that solution rises from just 4 for the correct angle initialisation up to 11 for the initialisation with a roll and pitch prior of 60° .

Even when the correct prior orientation is provided, the prior velocity is still all zero and this is why 4 iterations are required even in this case. This can be compared to the inertial delta observation initialisation which is only allowed one iteration for each pose added.

Figures 5.13, 5.14 and 5.15 show the initial velocity estimate convergence for the batch initialisation case. As with figures 5.11 and 5.12 it can be seen that

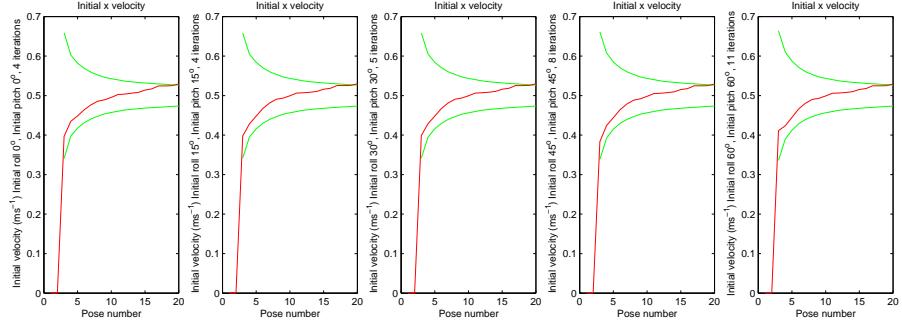


Figure 5.13: Initial x velocity estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.

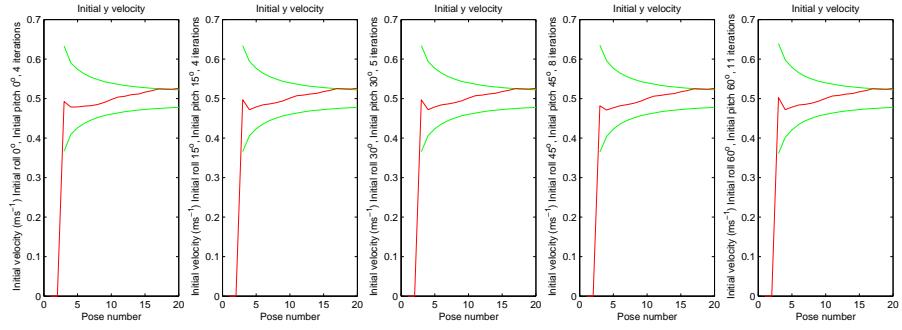


Figure 5.14: Initial y velocity estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.

the initial velocity estimate after the batch initialisation has occurred at pose 3 converges the same value regardless of the prior attitude provided.

Another method for initialising an inertial navigation solution is to over-parameterise the orientation using the sine and cosine of the angle and then set a large prior uncertainty [36]. This method has to be modified slightly for this application as instead of using GPS observations in a globally referenced frame, we only have camera observations taking in the body frame.

To do this modification the initial yaw is fixed at zero as it is unobservable and the roll and pitch are over parameterised. Landmark locations are defined in the body frame of the first pose they are observed from using the inverse depth parameterisation, as is done in all examples in this theses, so that if the initial orientation estimate changes the landmarks will automatically be rotated in the

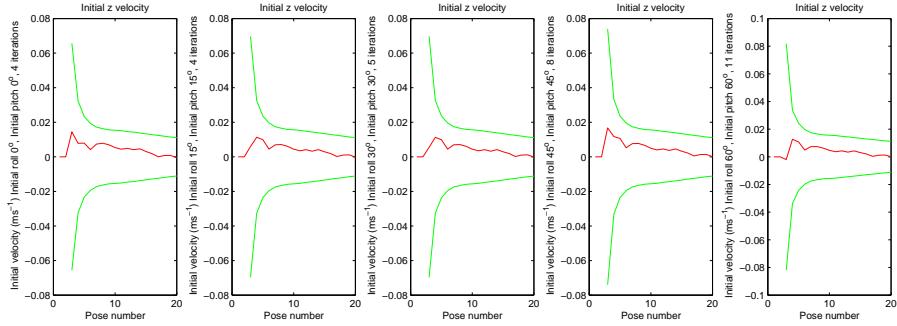


Figure 5.15: Initial z velocity estimate convergence using a batch initialisation technique over the first 20 poses for different initial attitude priors.

navigation frame to stay consistent with their first observation.

A range of different roll and pitch priors were used, the same as was done for the batch initialisation method, and the results for the initial roll and pitch estimates over the first 20 poses can be seen in figures 5.16 and 5.17.

As this is not a batch initialisation technique, only one iteration was allowed per pose as was done for the delta observation initialisation shown in figures 5.9 and 5.10.

It can be seen from figures 5.16 and 5.17 that the large angle initialisation technique appears to work well up to a prior roll and pitch error of about 30° however the solution does become a bit over confident at times as can be seen from the 2σ uncertainty bounds. This would be due to accumulating linearisation errors from the incorrect initial attitude estimates.

For the 45° and 60° prior roll and pitch examples the solution fails to converge. This is due to the magnitude of the linearisation errors being too large for the linear filter to cope with. The reasons why this may be more of a problem in this situation than with the intended application in [36] is most likely due to the kinds of observations provided.

In [36] this technique was created for GPS aided inertial navigation where the GPS observations are provided in the navigation frame. For the SLAM example used in this thesis, the landmark location must be estimated and the observations are provided in a body centred frame. Therefore when an incorrect initial attitude

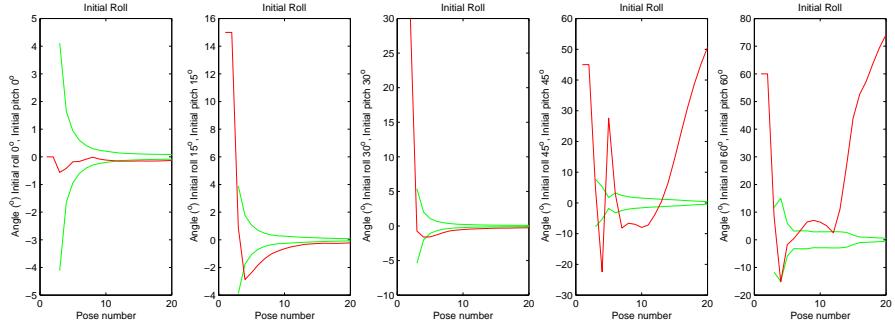


Figure 5.16: Initial roll estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.

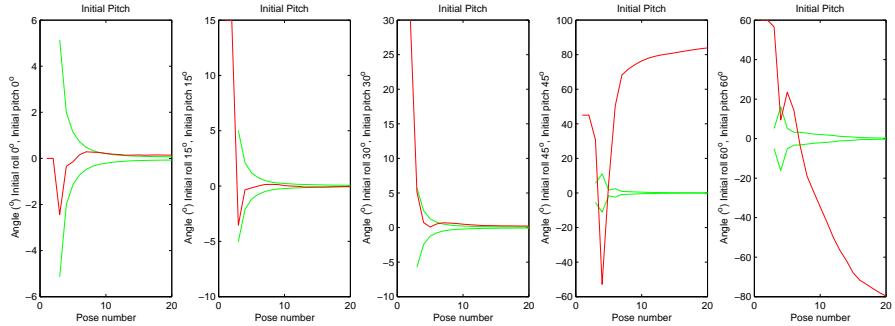


Figure 5.17: Initial pitch estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.

places the estimated landmark location in the wrong place, any further updates need to correct both the landmark location and the vehicle attitude. This extra level of estimation required makes the problem more ill conditioned to start with causing the solution to fail in more extreme situations.

Figures 5.18, 5.19 and 5.20 show the initial velocity estimates corresponding to the initial attitude estimates shown in figures 5.16 and 5.17. As with the initial attitude estimates for this case, the initial velocity estimates converge for prior attitude errors up to about 30° however they can be slightly over confident at times. Convergence of the solution fails for prior attitude errors of 45° and above.

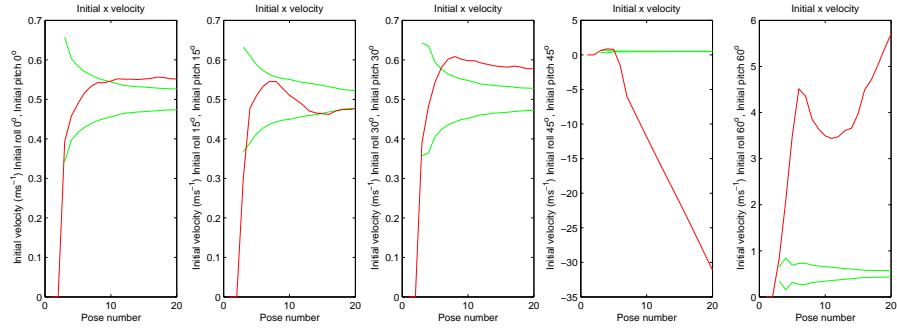


Figure 5.18: Initial x velocity estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.

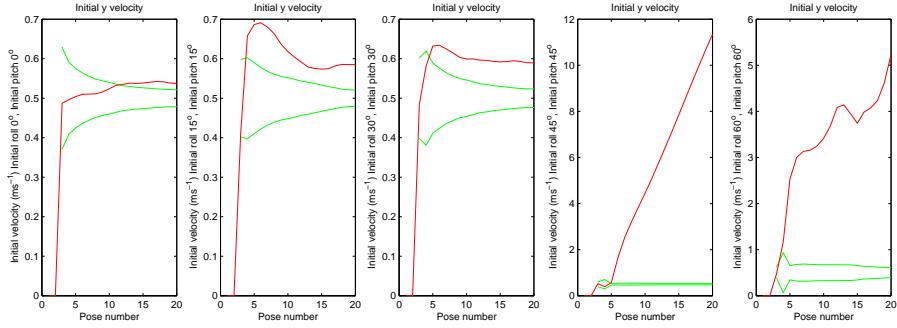


Figure 5.19: Initial y velocity estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.

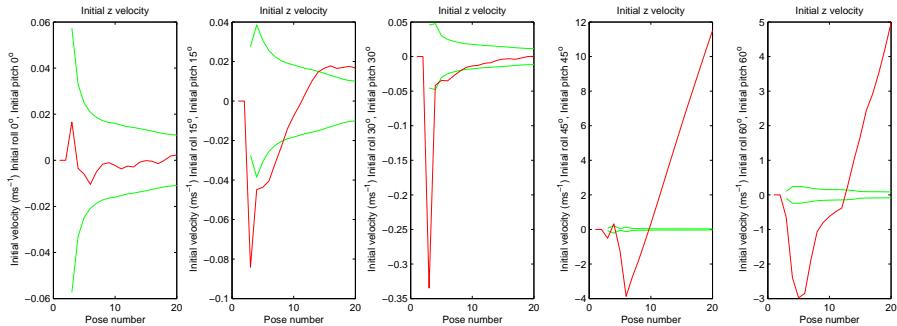


Figure 5.20: Initial z velocity estimate convergence using an over-parameterised roll and pitch initialisation technique over the first 20 poses for different initial attitude priors.

5.4 Chapter summary

This chapter investigated the properties of pre-integrated inertial delta observations developed in chapter 4 through both analysis of the equations and simulation.

The method by which pre-integration allows the initial velocity and gravity vector experienced by the IMU to be recovered in a linear way was explored and Monte-Carlo simulations were run to confirm that it operates as expected.

The validity of correcting for IMU biases after pre-integration if they are not known before and its effect on linearity was also investigated. Further Monte-Carlo tests were run to examine the effect that the non-linearity introduced by the unknown gyro biases has on the inertial navigation solution to confirm that they do not make a significant difference to the final solution. The non-linearity introduced by pre-integrating the IMU sensors noise is also discussed.

Finally a comparison between pre-integrated inertial delta observations and standard inertial integration for SLAM was conducted to confirm that similar results are obtained. Monte-Carlo simulations were performed both with inertial only navigation and with inertial SLAM using range and bearing observations to landmarks.

Chapter 6

Scale estimation

The method for incorporating inertial information into a navigation solution presented in chapter 4 assumes that relative motion observations are available from another sensor which are used to obtain estimates of the initial conditions of the platform. These kind of observations can easily be obtained from a range and bearing sensor, as used in the examples in chapter 5.

Range and bearing sensors, such as stereo camera pairs or laser scanners, are fairly common sensors to use in robotics however, in some applications they are not practical. An example is micro air vehicles where size and weight are even more tightly constrained than in most applications, or for unmanned aerial vehicles where the possible base line of the cameras compared to the distance to visual landmarks due to the flying altitude makes stereo observations virtually monocular.

In these situations where only bearing observations to landmarks are available the use of inertial observations and the estimation of initial conditions is not so straight forward and a slight modification to the estimator needs to be made.

6.1 The scale factor problem

The problem with using bearing only observations is not that the motion of the platform can not be estimated from the observations, but that the scale of the

map produced is not observable [28]. The usual solution to this problem is to fix an arbitrary scale to the map [28, 42]. This solution is fine when only sensors that do not observe scale are used, such as a single camera, but will cause problems when sensors that are sensitive to scale, such as IMUs are introduced.

The problem with aiding inertial observations with a map of unknown or arbitrary scale is that the observations between the camera and the IMU will not agree. Fortunately it can be concluded that if a scale difference between the camera constructed map and the true platform trajectory will cause problems when the inertial observations are fused then the arbitrary scale factor in the visual map must be observable [41].

If this scale factor difference is observable, a way of estimating it can be found and used to correct for the difference.

6.2 Scale estimation

6.2.1 The scale equation

Consider the case where motion is only in one dimension for simplicity in order to determine the requirements for the map scale to become observable from observations from an IMU and a single camera.

Figure 6.1 shows the sample problem where three poses are considered with bearing only landmark observations from each pose and inertial observations between successive poses. The inertial observations used will be in the form of pre-integrated inertial delta observations as described in chapter 4.

In this example there is no rotation so only accelerometer observations along the direction of travel of the platform are used to produce a single Δp^+ and Δv observation for each successive pair of poses. The effect of gravity and biases are not considered in this example.

The poses p_{t1} , p_{t2} and p_{t3} are in a local navigation frame fixed to the first pose (which may or may not be p_{t1}) and are in the arbitrary scale set in the

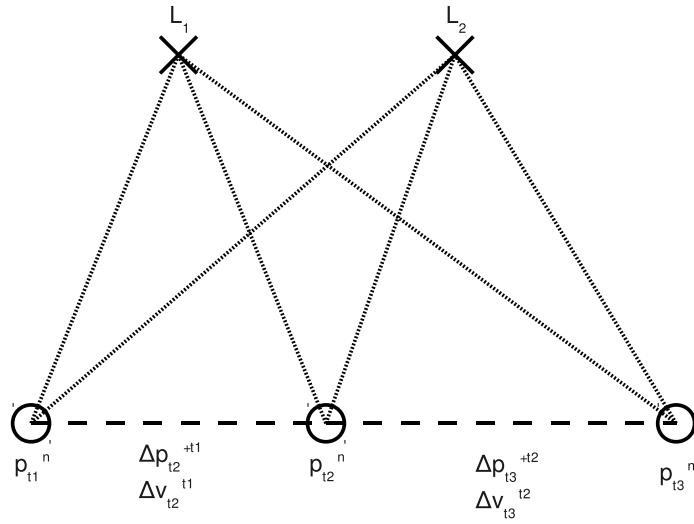


Figure 6.1: Sample problem to investigate map scale observability with bearing only observations and an IMU. In this case the platform moves in one dimension only with accelerometer observations along this dimension used to construct pre-integrated inertial delta observations.

monocular map. These pose estimates are provided as the result of a single camera SLAM or SFM optimisation and do not contain any information from the IMU observations. As the camera observations provide an estimate of the relative motion of the camera only, the origin of this local navigation frame can be arbitrarily set without affecting the final result.

The inertial delta observations are in the body frame of the first pose in the pose pair associated with each observation as described in chapter 4 and are independent of the map scale estimate or map coordinate frame chosen.

This example assumes that the landmark observations are sufficient to provide an estimate of the relative translation of the platform between poses. However no previously known initial conditions for the inertial observations (such as initial velocity) are assumed.

If only the first two poses, p_{t1} and p_{t2} , and their associated observations are considered, the pre-integrated inertial delta observation position update equation (equation 4.9) can be used to assess the scale observability from these observa-

tions. If equation 4.9 is adapted for this simplified problem, by removing the effects of gravity and rotation, and the unknown scale factor is expressed explicitly, as an unknown parameter α , equation 6.1 results.

$$\alpha p_{t2}^n = \alpha p_{t1}^n + (t2 - t1)v_{t1}^n + \Delta p_{t2}^{+t1} \quad (6.1)$$

With p_{t1}^n and p_{t2}^n being provided by the camera observations and Δp_{t2}^{+t1} coming from the pre-integrated inertial observations and times t_1 and t_2 , the only unknown variables in equation 6.1 is the initial velocity v_{t1}^n and the scale parameter α . Therefore with two unknowns and only one equation the scale of the map is not observable from two pose observations unless an estimate of the initial velocity of the platform is known.

If a third pose observation, p_{t3} , is added along with the inertial delta observations from pose p_{t2} to pose p_{t3} , then an additional position update equation, equation 6.2, as well as a velocity update equation, equation 6.3, become available. As long as some of the same visual features are observed from all three poses then the scale parameter, α , in equations 6.1 and 6.2 are the same.

$$\alpha p_{t3}^n = \alpha p_{t2}^n + (t3 - t2)v_{t2}^n + \Delta p_{t3}^{+t2} \quad (6.2)$$

$$v_{t2}^n = v_{t1}^n + \Delta v_{t2}^{t1} \quad (6.3)$$

Substituting equation 6.3 into equation 6.2 results in equation 6.4.

$$\alpha p_{t3}^n = \alpha p_{t2}^n + (t3 - t2) (v_{t1}^n + \Delta v_{t2}^{t1}) + \Delta p_{t3}^{+t2} \quad (6.4)$$

Rearranging equation 6.1 for the velocity term gives equation 6.5.

$$v_{t1}^n = \frac{\alpha p_{t2}^n - \alpha p_{t1}^n - \Delta p_{t2}^{+t1}}{(t2 - t1)} \quad (6.5)$$

Which when substituted into equation 6.4 and rearranged for the scale term, the scale equation, equation 6.6, is obtained.

$$\alpha = \frac{-\left(\frac{t_3-t_2}{t_2-t_1}\right)\Delta p_{t2}^{+t1} + (t_3-t_2)\Delta v_{t2}^{t1} + \Delta p_{t3}^{+t2}}{p_{t3}^n - \left(1 + \frac{t_3-t_2}{t_2-t_1}\right)p_{t2}^n + \left(\frac{t_3-t_2}{t_2-t_1}\right)p_{t1}^n} \quad (6.6)$$

The scale factor, α , can be interpreted as the inertial or true scale divided by the map scale. Therefore a value of α that is greater than 1 indicates that the arbitrary map scale is smaller than the true scale and vice versa. α should always be positive.

6.2.2 Scale observability

Once the scale equation, equation 6.6, has been defined, it can be used to assess the situations in which scale becomes observable.

If for simplicity and without the loss of generality the time between observed poses is assumed to be a constant, dt , then equation 6.6 simplifies and is rearranged to equation 6.7.

$$\alpha = \frac{\Delta p_{t3}^{+t2} - \Delta p_{t2}^{+t1} + \Delta v_{t2}^{t1}dt}{(p_{t3}^n - p_{t2}^n) - (p_{t2}^n - p_{t1}^n)} \quad (6.7)$$

It can be seen from equation 6.7 that the denominator of the scale equation is simply the difference in average velocities of the platform between poses 1 and 2 and poses 2 and 3 as estimated from the landmark observations. The numerator is also the difference in average velocities of the platform between the same poses as estimated from the inertial observations.

It is from this ratio of change in average velocity estimates that the map scale can be estimated. As the scale equation works with changes in estimated average velocities, it can be speculated that without acceleration the scale factor difference between the map and the true trajectory is not observable.

If no acceleration is sensed by the IMU then the Δp_{t3}^{+t2} , Δp_{t1}^{+t1} and Δv_{t2}^{t1} terms will be zero. In this case the numerator of the scale equation will be zero as shown in equation 6.8, resulting in an estimated scale factor of zero.

$$\alpha = \frac{0 - 0 + 0dt}{(p_{t3}^n - p_{t2}^n) - (p_{t2}^n - p_{t1}^n)} = \frac{0}{(p_{t3}^n - p_{t2}^n) - (p_{t2}^n - p_{t1}^n)} = 0 \quad (6.8)$$

Similarly if the distance between the poses as estimated from the visual observations is the same then the denominator of the scale equation will be zero since $p_{t3}^n - p_{t2}^n$ will equal $p_{t2}^n - p_{t1}^n$ resulting in the estimate of α being undefined.

Both of these are extreme cases of when the scale factor is not observable, however when uncertainty in the observations is taken into account varying degrees of scale factor observability can be expected.

6.2.3 Scale uncertainty

In section 6.2.2 it was shown that the scale factor, α , can be observable from just three pose observations. The uncertainty associated with this scale estimate however is dependant on the observability of linear acceleration between the three poses in both the map pose estimates and the inertial observations.

Therefore it would be useful to be able to work out how much acceleration is required for the scale to be known to a desired level of certainty.

If the same simplification of equation 6.6 as made in section 6.2.2 to produce equation 6.7 is made, the derivatives of this equation with respect to the observations and estimates used to calculate it can be found as shown in table 6.1.

Table 6.1: Derivatives of the scale equation 6.7

	Δp_{t3}^{+t2}	Δp_{t2}^{+t1}	Δv_{t2}^{t1}	p_{t3}^n	p_{t2}^n	p_{t1}^n
α	$\frac{1}{\gamma}$	$-\frac{1}{\gamma}$	$\frac{dt}{\gamma}$	$-\frac{\beta}{\gamma^2}$	$\frac{2\beta}{\gamma^2}$	$-\frac{\beta}{\gamma^2}$

Where $\beta = \Delta p_{t3}^{+t2} - \Delta p_{t2}^{+t1} + \Delta v_{t2}^{t1}dt$ and $\gamma = p_{t3}^n - 2p_{t2}^n + p_{t1}^n$. These derivatives of the scale equation can be used along with the associated uncertainty of the observations and estimates used to calculate an estimate of the uncertainty of the scale parameter from this image triplet.

A Jacobian matrix for the scale estimate can be formed, such as in equation 6.9, from the entries in table 6.1. An associated uncertainty matrix for this observation can also be constructed as shown in equation 6.10.

$$H_\alpha = \begin{bmatrix} \frac{d\alpha}{d\Delta p_{t3}^{+t2}} & \frac{d\alpha}{d\Delta p_{t2}^{+t1}} & \frac{d\alpha}{d\Delta v_{t2}^{t1}} & \frac{d\alpha}{dp_{t3}^n} & \frac{d\alpha}{dp_{t2}^n} & \frac{d\alpha}{dp_{t1}^n} \end{bmatrix} \quad (6.9)$$

$$P_\alpha = \begin{bmatrix} P_{\Delta p_{t3}^{+t2} \Delta p_{t3}^{+t2}} & 0 & 0 & 0 & 0 & 0 \\ 0 & P_{\Delta p_{t2}^{+t1} \Delta p_{t2}^{+t1}} & P_{\Delta p_{t2}^{+t1} \Delta v_{t2}^{t1}} & 0 & 0 & 0 \\ 0 & P_{\Delta v_{t2}^{t1} \Delta p_{t2}^{+t1}} & P_{\Delta v_{t2}^{t1} \Delta v_{t2}^{t1}} & 0 & 0 & 0 \\ 0 & 0 & 0 & P_{p_{t3}^n p_{t3}^n} & P_{p_{t3}^n p_{t2}^n} & P_{p_{t3}^n p_{t1}^n} \\ 0 & 0 & 0 & P_{p_{t2}^n p_{t3}^n} & P_{p_{t2}^n p_{t2}^n} & P_{p_{t2}^n p_{t1}^n} \\ 0 & 0 & 0 & P_{p_{t1}^n p_{t3}^n} & P_{p_{t1}^n p_{t2}^n} & P_{p_{t1}^n p_{t1}^n} \end{bmatrix} \quad (6.10)$$

The entries for the top three rows of equation 6.10 are obtained from the inertial delta observation covariance matrix (the R matrix from algorithm 4.3). The entries for the bottom three rows are from the uncertainty estimate of the bearing only trajectory reconstruction.

Finally an estimate of the uncertainty of the scale factor estimate, α , is obtained from equation 6.11.

$$\sigma_\alpha^2 = H_\alpha P_\alpha H'_\alpha \quad (6.11)$$

This estimated scale uncertainty can be used for weighting when combining map scale estimates from different poses as well as to determine the amount of scale information available from a given trajectory.

6.3 Extending beyond 3 poses

The scale equation introduced in section 6.2.1 can be used to determine an estimate of the scale factor difference between the bearing only map and the inertial

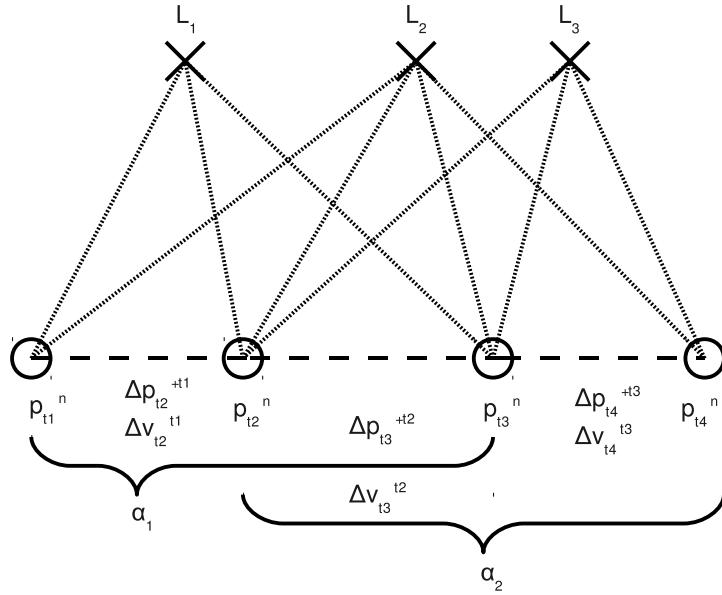


Figure 6.2: Sample problem with 4 poses to show which pose estimates and inertial observations are used to construct the 2 different estimates of the scale factor, α_1 and α_2 . In this case the platform moves in one dimension only with accelerometer observations along this dimension used to construct pre-integrated inertial delta observations.

observations over a triplet of poses. If estimation is to be performed over a larger sequence of poses the correlations between the estimates from different triplets of poses must be accounted for as they would have observations in common.

If a four pose sequence in one dimension is considered, and the simplifications made in equation 6.7 are performed, a second equation for the estimation of α can be made using the last three poses in the sequence. This is equation 6.12

$$\alpha_2 = \frac{\Delta p_{t4}^{+t3} - \Delta p_{t3}^{+t2} + \Delta v_{t3}^{t2} dt}{(p_{t4}^n - p_{t3}^n) - (p_{t3}^n - p_{t2}^n)} \quad (6.12)$$

Figure 6.2 illustrates this problem showing the four poses and landmarks along with the observations made. The poses and observations used to produce the two different estimates of the scale factor, α_1 and α_2 , are shown. Note that poses p_{t2}^n and p_{t3}^n are common to both estimates of α as is the inertial observations Δp_{t3}^{+t2} . This use of common pose estimates and observations between the two scale factor

estimates is what leads to their correlation.

Δv_{t3}^{t2} is not a common observations for the two α estimates as it is not a component of the α_1 estimate as can be seen from equation 6.7.

Note also that all landmarks must be observed at least three times for them to be able to contribute to the scale factor estimate. Also common landmarks must be observed between any two consecutive poses in order for it to be possible to estimate the scale factor over this portion of the trajectory.

An important consequence of this observation is that if there are two consecutive poses within a sequence where no common landmark with an observation track over three or more poses is available then the scale factor estimate can not be carried over from one of these poses to the other. Effectively the scale estimation for the poses after this point has to be performed independently of the previous poses. This has important implication for applications where accurate scale observability is required or where scale may not be observable from every pose.

6.3.1 Scale estimate correlation

Table 6.2 shows the derivatives of the two scale factor estimates, α_1 and α_2 , with respect to the estimated poses and inertial observations. Here again it can be seen that pose estimates p_{t2}^n and p_{t3}^n and the inertial observation Δp_{t3}^{+t2} are common between the scale factor estimates.

Table 6.2: Derivatives of the two scale equations derived from observations from 4 consecutive poses

	Δp_{t4}^{+t3}	Δp_{t3}^{+t2}	Δp_{t2}^{+t1}	Δv_{t3}^{t2}	Δv_{t2}^{t1}	p_{t4}^n	p_{t3}^n	p_{t2}^n	p_{t1}^n
α_1	0	$\frac{1}{\gamma_1}$	$-\frac{1}{\gamma_1}$	0	$\frac{dt_1}{\gamma_1}$	0	$-\frac{\beta_1}{\gamma_1^2}$	$\frac{2\beta_1}{\gamma_1^2}$	$-\frac{\beta_1}{\gamma_1^2}$
α_2	$\frac{1}{\gamma_2}$	$-\frac{1}{\gamma_2}$	0	$\frac{dt_2}{\gamma_2}$	0	$-\frac{\beta_2}{\gamma_2^2}$	$\frac{2\beta_2}{\gamma_2^2}$	$-\frac{\beta_2}{\gamma_2^2}$	0

If a single α value is to be calculated, this can be done by combining both estimates. In order to find the correlation between the estimates, a covariance

matrix containing the states and observations used in the estimates needs to be constructed.

For simplicity two covariance matrices, one for the states and one for the observations, will be formed. This is a similar approach as that used in an EKF prediction step of standard inertial SLAM [9] where the inertial observations are treated like control inputs. This is possible as there are no off diagonal terms between the inertial delta observations used and the position estimates provided from the bearing only observations alone at this point.

Two Jacobian matrices will be formed, one for the derivatives of the scale estimates with respect to the estimated pose states, H_α , and one for the derivatives of the scale estimates with respect to the inertial delta observations, G_α . These matrices can be seen in equations 6.13 and 6.14 respectively.

$$H_\alpha = \begin{bmatrix} 0 & \frac{d\alpha_1}{dp_{t3}^n} & \frac{d\alpha_1}{dp_{t2}^n} & \frac{d\alpha_1}{dp_{t1}^n} \\ \frac{d\alpha_2}{dp_{t4}^n} & \frac{d\alpha_2}{dp_{t3}^n} & \frac{d\alpha_2}{dp_{t2}^n} & 0 \end{bmatrix} \quad (6.13)$$

$$G_\alpha = \begin{bmatrix} 0 & \frac{d\alpha_1}{d\Delta p_{t3}^{+t2}} & \frac{d\alpha_1}{d\Delta p_{t2}^{+t1}} & 0 & \frac{d\alpha_1}{d\Delta v_{t2}^{t1}} \\ \frac{d\alpha_2}{d\Delta p_{t4}^{+t3}} & \frac{d\alpha_2}{d\Delta p_{t3}^{+t2}} & 0 & \frac{d\alpha_2}{d\Delta v_{t3}^{t2}} & 0 \end{bmatrix} \quad (6.14)$$

Along with the Jacobian matrices, two covariance matrices also need to be constructed. One for the pose state estimates made from the bearing only observations, P_α , shown in equation 6.15, note that this equation is fully dense. And one for the inertial delta observations covariances, Q_α , shown in equation 6.16. This matrix is considerably more sparse.

$$P_\alpha = \begin{bmatrix} P_{p_{t4}^n p_{t4}^n} & P_{p_{t4}^n p_{t3}^n} & P_{p_{t4}^n p_{t2}^n} & P_{p_{t4}^n p_{t1}^n} \\ P_{p_{t3}^n p_{t4}^n} & P_{p_{t3}^n p_{t3}^n} & P_{p_{t3}^n p_{t2}^n} & P_{p_{t3}^n p_{t1}^n} \\ P_{p_{t2}^n p_{t4}^n} & P_{p_{t2}^n p_{t3}^n} & P_{p_{t2}^n p_{t2}^n} & P_{p_{t2}^n p_{t1}^n} \\ P_{p_{t1}^n p_{t4}^n} & P_{p_{t1}^n p_{t3}^n} & P_{p_{t1}^n p_{t2}^n} & P_{p_{t1}^n p_{t1}^n} \end{bmatrix} \quad (6.15)$$

$$Q_\alpha = \begin{bmatrix} P_{\Delta p_{t4}^{+t3} \Delta p_{t4}^{+t3}} & 0 & 0 & 0 & 0 \\ 0 & P_{\Delta p_{t3}^{+t2} \Delta p_{t3}^{+t2}} & 0 & P_{\Delta p_{t3}^{+t2} \Delta v_{t3}^{t2}} & 0 \\ 0 & 0 & P_{\Delta p_{t2}^{+t1} \Delta p_{t2}^{+t1}} & 0 & P_{\Delta p_{t2}^{+t1} \Delta v_{t2}^{t1}} \\ 0 & P_{\Delta v_{t3}^{t2} \Delta p_{t3}^{+t2}} & 0 & P_{\Delta v_{t3}^{t2} \Delta v_{t3}^{t2}} & 0 \\ 0 & 0 & P_{\Delta v_{t2}^{t1} \Delta p_{t2}^{+t1}} & 0 & P_{\Delta v_{t2}^{t1} \Delta v_{t2}^{t1}} \end{bmatrix} \quad (6.16)$$

Once these four matrices are formed the covariance matrix between the two estimates of the scale factor α can be found using equation 6.17. This will be a dense 2 by 2 matrix.

$$\Sigma_\alpha^2 = H_\alpha P_\alpha H'_\alpha + G_\alpha Q_\alpha G'_\alpha \quad (6.17)$$

This method can be extended to any number of poses so that scale factor estimates can be calculated over arbitrarily long observation tracks.

6.4 One dimensional scale observability analysis

To asses the observability of the scale equation developed in section 6.2 a one dimensional simulation was performed. The simulation set-up is as described in appendix A but the motion was constrained to the y direction only. The trajectory used was $\sin(\text{time}/2) + \cos(\text{time}/2)$.

SLAM was performed using bearing only observations to generate the estimated map and trajectory, the inertial observations were not used in this estimate. The distance between the first two poses was arbitrarily set to 1 metre to fix the scale of the visual map. Figure 6.3 shows both the simulated position and the estimated position with the arbitrarily set scale. The visual map scale is approximately 13 time larger than the true scale as the first two positions are simulated as 0.0767m apart.

Once the bearing only SLAM estimate was calculated, the scale factor, α ,

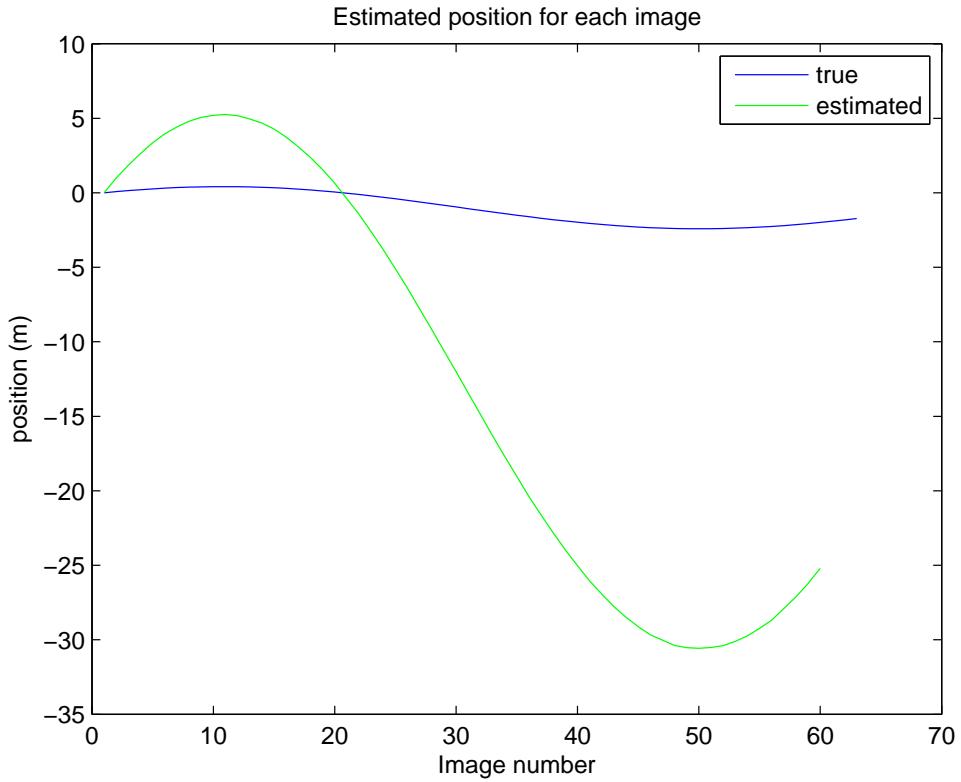


Figure 6.3: True simulated position and estimated position using bearing only SLAM and an arbitrary distance on 1 metre between the first two poses. Inertial observations are not used in the SLAM estimate.

was calculated for each three pose triplet in the trajectory by equation 6.7 using the estimated positions and the accelerometer observations. For comparison a true scale factor was also calculated using the true simulated positions and the bearing only SLAM estimated positions by equation 6.18.

$$\alpha_{true} = \frac{(p_{t3true}^n - p_{t2true}^n) - (p_{t2true}^n - p_{t1true}^n)}{(p_{t3}^n - p_{t2}^n) - (p_{t2}^n - p_{t1}^n)} \quad (6.18)$$

These can both be seen in figure 6.4. Notice how the true scale factor is fairly stable around the true value of 0.0767 while the estimated scale factor fluctuates wildly. This is due to the scale only being weakly observable from any three chosen poses. This fluctuating level of observability is captured in the scale factor uncertainty estimate calculated from equation 6.11 which is also shown in

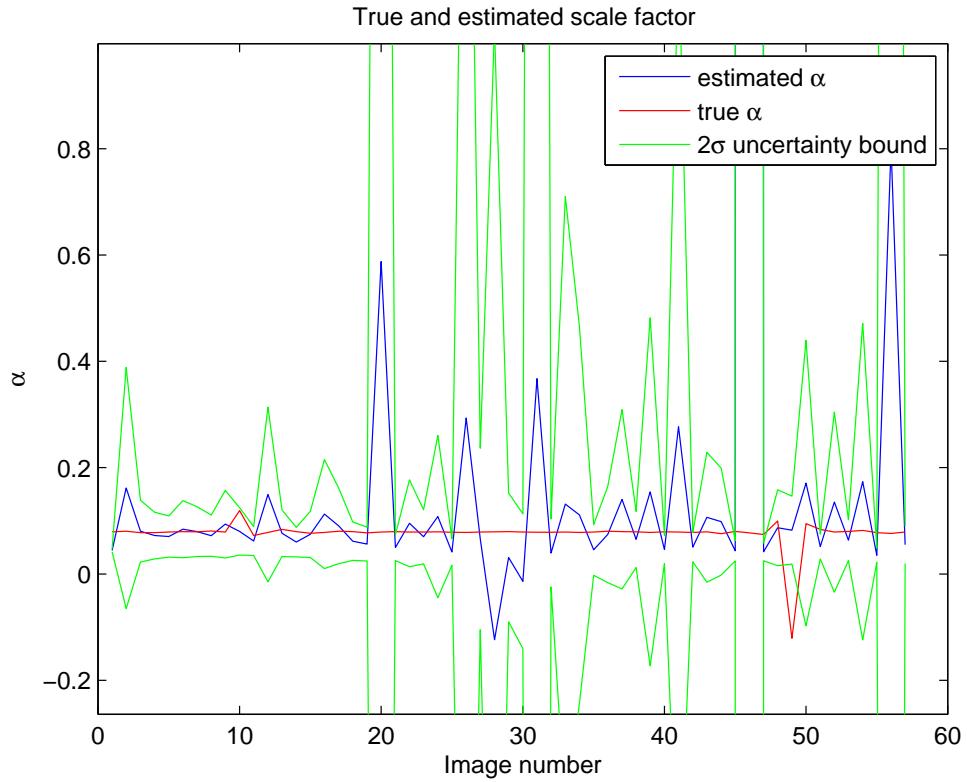
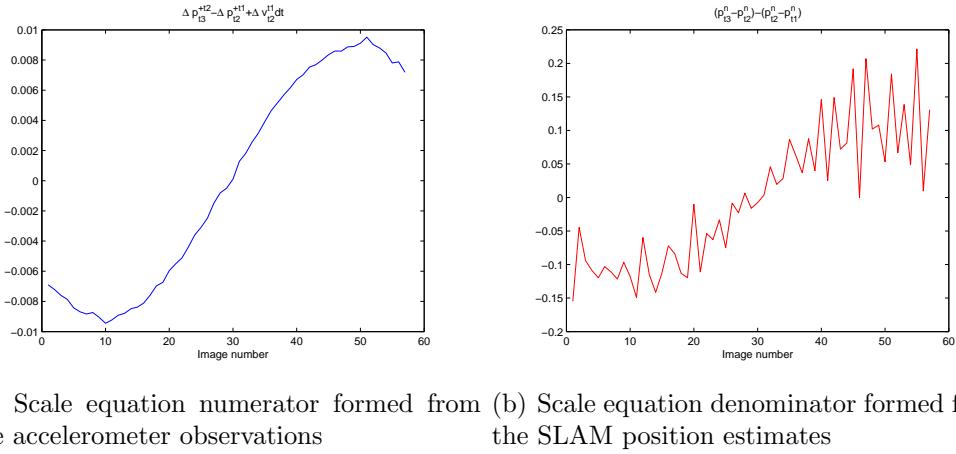


Figure 6.4: Comparison of the estimated scale factor from equation 6.7 to the true scale factor from equation 6.18 for the simulated trajectory shown in figure 6.3.

figure 6.4. These scale uncertainty estimates are centred around the estimated scale factor, not the true scale factor. Negative estimates for the scale factor occur when the acceleration observation from the IMU is in the opposite direction to that predicted from the visual map estimates.

The two components used to calculate the estimated scale factor, the numerator formed from the accelerometer observations and the denominator formed from the bearing only SLAM position estimates, are shown in figure 6.5. These plots give a clue as to how the accuracy of the scale estimate can fluctuate so much within the trajectory.

As the scale estimate is basically a ratio of the change in average velocity over three consecutive poses, if there is little or no change in relative velocity



(a) Scale equation numerator formed from the accelerometer observations
(b) Scale equation denominator formed from the SLAM position estimates

Figure 6.5: Plots of the numerator and denominator components of the scale equation shown in figure 6.4

over this period, the scale can not be estimated accurately. While the numerator (figure 6.5(a)), formed from the accelerometer observations is fairly smooth even with the sensor noise, it is obvious that the main source of the scale estimate uncertainty is the noisy denominator of equation (figure 6.5(b)) formed from the bearing only SLAM pose estimates.

The large scale uncertainty spike that can be seen around image 46 in figure 6.4 can be explained by the denominator of the scale equation dropping to almost zero at this point as can be seen in figure 6.5(b).

6.5 Extending to the 6DoF case

The scale equation developed in section 6.2 is for a one dimensional example, if it is to be applied to a full 6 DoF system it must be adapted.

The simplest way to achieve this is to use the gyro observation with the bearing only observations to estimate the visual map. Gyro observations can be used in the map as they provide relative rotation estimates between the poses without the need for full inertial integration and without requiring the map scale.

Using gyro observations in a visual map also constrains the rotation/translation ambiguity that can occur with narrow angle (approximately affine) cameras [13]

making relative pose estimates, which are needed for estimating scale, more accurate. This also increases the accuracy of the scale estimate as can be seen from section 6.2.3.

Once the orientation of the poses are known to a high level of accuracy from the gyro observations, this virtually reduces the problem to three, one dimensional, scale estimates.

If the scale equation, equation 6.7, for the one dimensional problem is calculated for each of the three translational dimensions, three scale estimates, one for each dimension are obtained, as shown in equation 6.19. As gravity is present in the 6DoF case, the inertial delta observations also need to have the effect of the gravity vector removed, the $\frac{1}{2}g^n\Delta t^2$ terms for the two Δp^+ components cancel out, leaving just the $g^n\Delta t$ term for the Δv component.

$$\begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = \begin{bmatrix} \frac{\Delta p_{t3x}^{+t2} - \Delta p_{t2x}^{+t1} + \Delta v_{t2x}^{t1} dt - g_x^n \Delta t}{(p_{t3x}^n - p_{t2x}^n) - (p_{t2x}^n - p_{t1x}^n)} \\ \frac{\Delta p_{t3y}^{+t2} - \Delta p_{t2y}^{+t1} + \Delta v_{t2y}^{t1} dt - g_y^n \Delta t}{(p_{t3y}^n - p_{t2y}^n) - (p_{t2y}^n - p_{t1y}^n)} \\ \frac{\Delta p_{t3z}^{+t2} - \Delta p_{t2z}^{+t1} + \Delta v_{t2z}^{t1} dt - g_z^n \Delta t}{(p_{t3z}^n - p_{t2z}^n) - (p_{t2z}^n - p_{t1z}^n)} \end{bmatrix} \quad (6.19)$$

The x , y and z subscripts on the position estimate and inertial delta terms denotes that it is the component of the position or inertial delta observation in the x , y or z direction of the map navigation frame respectively. These components are simply recovered for the position estimates as they are already in this frame, however the inertial delta observations are not in this frame as they are calculated in the body frame of the last pose.

The inertial delta observations need to be rotated from the body frame into the map navigation frame using the C_{b1}^n and C_{b2}^n rotation matrices that are formed by the attitude estimated for their respective poses in the vision/gyro generated map. This rotation applied to equation 6.19 can be seen in equation 6.20.

$$\alpha = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \end{bmatrix} = \frac{C_{b2}^n \Delta p_{t3}^{+t2} - C_{b1}^n \Delta p_{t2}^{+t1} + C_{b1}^n \Delta v_{t2}^{t1} dt - g^n \Delta t}{(p_{t3}^n - p_{t2}^n) - (p_{t2}^n - p_{t1}^n)} \quad (6.20)$$

This gives three independent scale estimates, one for each dimension. As long as the correlations between the orientation estimate of the inertial trajectory and the map are maintained, through the inertial delta attitude observation covariances a consistent solution can be found.

An added advantage of keeping track of these correlations is that the scale does not have to be observable in all three directions as landmark observations tie the dimensions together. Scale observability in only one direction is sufficient, therefore linear acceleration is only needed in one direction.

If the scale uncertainty is to be calculated as well, the covariance matrices for the inertial delta observations also need to be rotated into the navigation frame. This needs to be done before the uncertainty matrix for the scale estimate observations from equation 6.10 can be calculated.

This rotation of the observation covariance can be done as shown in equation 6.21 from [53]. P_{Δ}^b is the original delta observation covariance, C_b^n is the rotation matrix between the current body frame and the navigation frame and P_{Δ}^n is the delta observation covariance in the navigation frame to be used in equation 6.10.

$$P_{\Delta}^n = C_b^n P_{\Delta}^b C_b^{nT} \quad (6.21)$$

6.6 Scale parameterisation

When implementing such a system with a bearing only sensor and inertial observations there is a choice of the possible parameterisation used to resolve this scale ambiguity.

The explicit method explained in section 6.2.1 can be used where the bearing only observations are used to construct a map of arbitrarily fixed scale then accelerometer observations can be used with these estimates to determine the true scale of the map. The scale factor between these two estimations, α , is then estimated explicitly.

When this method is used the gyro observations can also be combined into

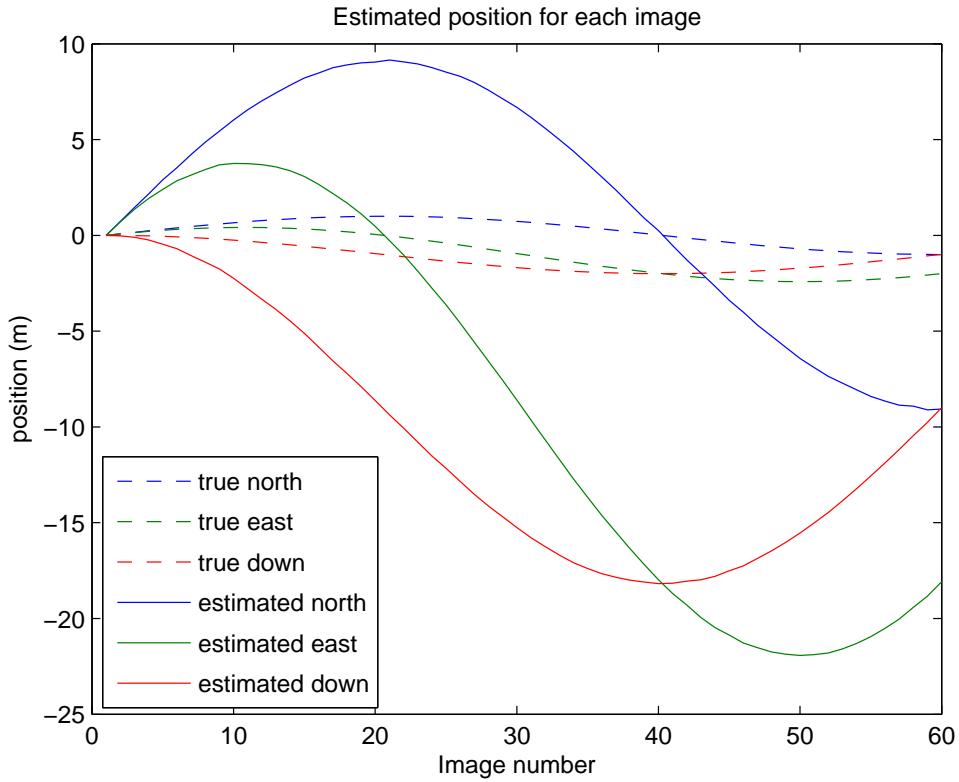


Figure 6.6: True simulated position and estimated position using bearing only SLAM and an arbitrary distance on 1 metre between the first two poses for the 6DoF case. Gyro observations are used in the bearing only SLAM estimate but accelerometer observations are not.

the map to aid in its construction as scale information is not required for these observations to be used. This is useful as the more accurate orientation information obtained from the gyros leads to more accurate position estimates and therefore scale estimates for the map.

6.6.1 Three dimensional explicit scale estimation

To test the explicit scale estimation for the 6DoF case, simulations were run using the set-up explained in appendix A with the full 6 degrees of freedom trajectory. Bearing only landmark observations, as well as gyro observations were used to construct a SLAM map and trajectory estimate without the accelerometer

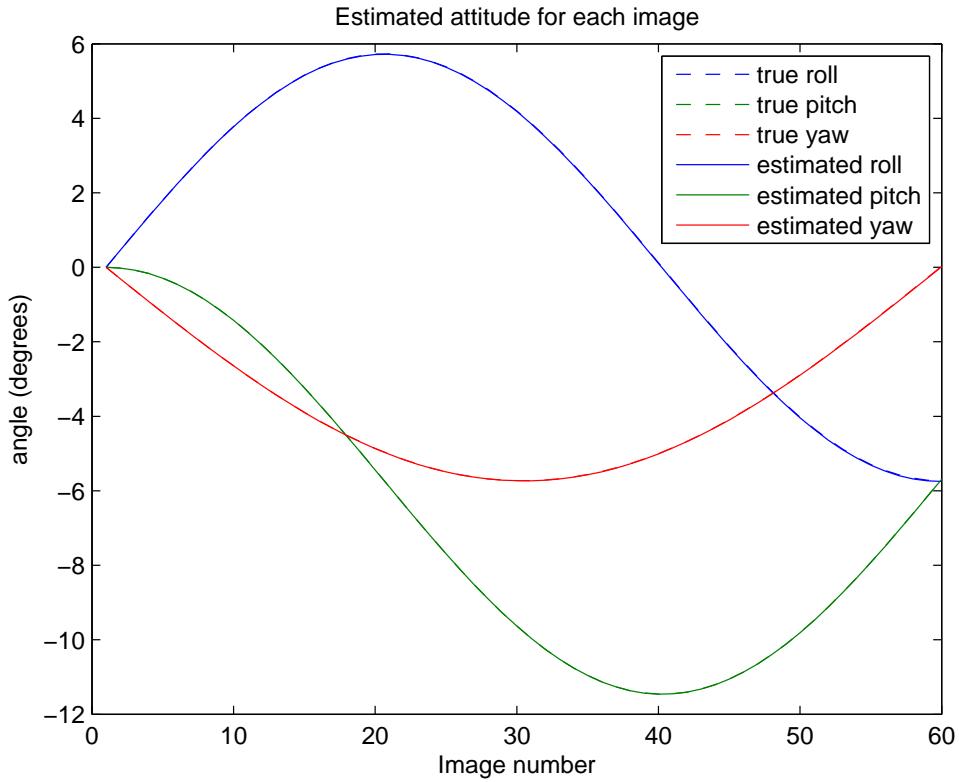


Figure 6.7: True simulated attitude and estimated attitude using bearing only SLAM and an arbitrary distance between the first two poses for the 6DoF case. Gyro observations are used in the bearing only SLAM estimate, leading to the high accuracy of the estimate shown, but accelerometer observations are not.

observations. An arbitrary scale constraint of fixing the distance between the first two poses to be 1 metre was also used.

Figure 6.6 shows the true and estimated trajectory used for this simulated example. It can be seen that the estimated scale is approximately 9 times the true scale as the first two poses are actually 0.1103 metre apart instead of 1 metre as arbitrarily set for the map. This implies an expected estimated α value of approximately 0.11.

The attitude estimates for this trajectory are estimated very accurately due to the inclusion of the gyro observations in the bearing only SLAM estimate. This can be seen from figure 6.7.

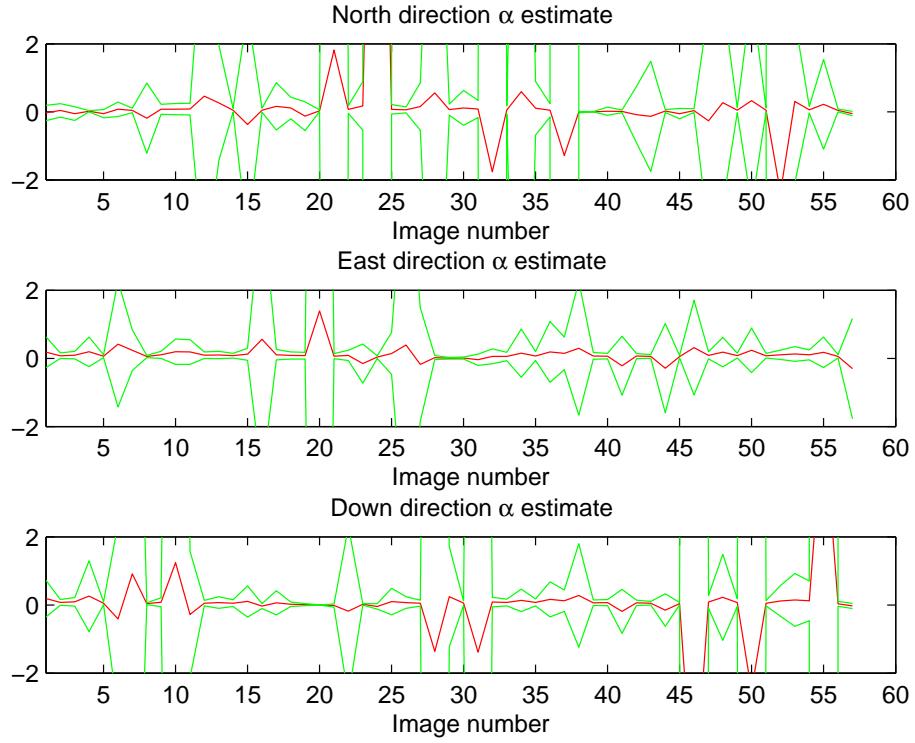


Figure 6.8: Estimated scale factor (α) for each direction in the 6DoF simulation with the estimated 2σ uncertainty bounds.

Figure 6.8 shows the estimated α values for each of the 3 translational directions from the trajectory shown in figures 6.6 and 6.7. These values were calculated using equation 6.20. It can be seen from this figure that the scale estimate, which is expected to be around 0.11, still fluctuates wildly and suffers from low observability during much of the trajectory.

6.6.2 Implicit scale parameterisation

Another option is to attempt to fuse both the inertial and bearing only observations at the true scale with the scale factor expressed implicitly within the landmark map. If the scale factor of the map is expressed explicitly in the distances to the landmarks as in [41], additional constraints need to be imposed to keep

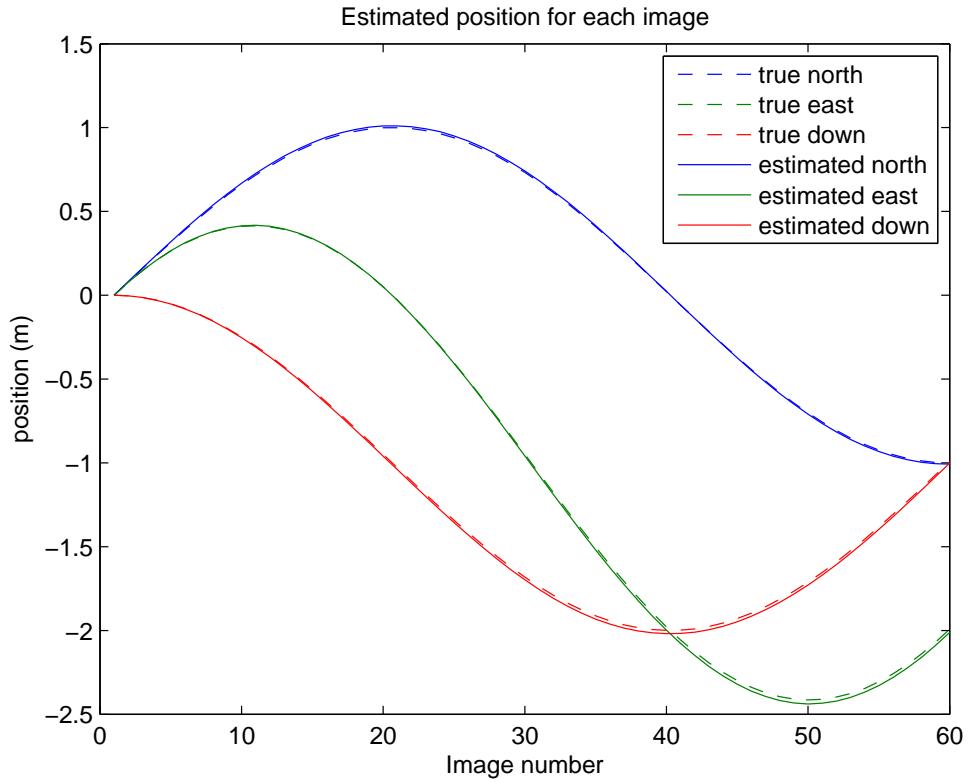


Figure 6.9: True simulated position and estimated position using bearing only SLAM and inertial observations via the implicit scale estimation method for the 6DoF case.

the problem well conditioned until scale is observable. This can be done through a constraint on distance the between the estimated poses in the trajectory or on the range to landmarks.

To test this method, the same simulation data as used in section 6.6.1 was used to generate a visual-inertial SLAM map. As the map scale is not initially known, a weak prior in the distance between two poses in the map is usually required in order to allow an estimate to be obtained until the scale of the map becomes observable. This prior was added at the beginning of the estimation but later removed so that the estimated map scale was completely a product of the visual and inertial observations.

The resultant estimated trajectory can be seen in figure 6.9. The scale of the

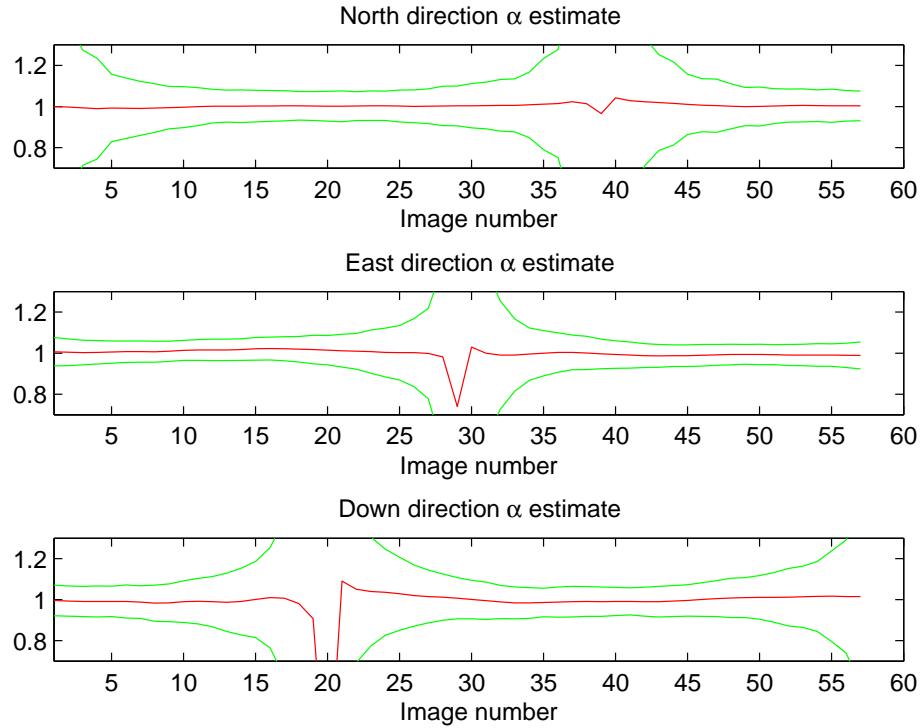


Figure 6.10: Estimated scale factor α for the three directions with associated 2σ uncertainty bounds for the 6DoF trajectory with implicit scale estimation. Notice how the true scale of 1 is within the uncertainty bounds for the whole trajectory.

estimated trajectory is almost exactly the same as the true scale as can be seen from this figure.

To assess the estimated scale, the scale factor in each direction over the trajectory was calculated along with their associated uncertainties using equation 6.20, these are shown in figure 6.10.

Explicit representation of scale leads to much faster convergence of the map as the scale of the map is not continually changing as it becomes observable. However, in comparison to the result shown in figure 6.8, the result in figure 6.10 is much smoother showing the stabilising effect on the map scale that the accelerometer observations give. This is a result of the much more accurate relative acceleration estimates obtained.

The implicit method also has the advantage of constraining the scale drift over long trajectories. Using the explicit method, a constant scale factor for the whole map is assumed which is not reasonable for long trajectories. Because of this in practice the implicit method is the method of choice for implementations where scale observability is obtained in this way.

Another thing to notice from figure 6.10 is that the regions where the accuracy of the scale estimate from three consecutive poses is at its lowest is around pose 1 and 40 for the north direction, pose 30 for the east direction and pose 20 and 60 for the down direction. It can be seen from figure 6.9 that these poses correspond to the points in the trajectory when the acceleration in the respective directions is small. This result is as expected from the analysis of the scale equation performed in section 6.2.2.

6.6.3 Implicit scale analysis

In order to validate the map scale estimated using the implicit scale parameterisation method, 100 Monte-Carlo runs were performed over the first 20 poses of the trajectory in the same way as for the results shown in figure 6.9. To assess the estimation of scale, the estimates for the extreme points in the north, east and down direction for each run were compared to the true value and their mean and variance calculated. These results can be seen in table 6.3.

Table 6.3: Extreme values in the north, east and down components from 100 Monte-Carlo simulations of the first 20 poses for implicit scale estimation from the trajectory shown in figure 6.9

Direction	μ_{true}	$\bar{\mu}$	$\sigma_{\bar{\mu}}$	$\bar{\mu} - \mu_{true}$	$\sigma_{\bar{\mu}}/\sqrt{n}$	$1 - (\mu_{true}/\bar{\mu})$
north (m)	0.99871	1.00729	0.13420	0.00858	0.01342	0.00852
east (m)	0.41406	0.41858	0.05882	0.00452	0.00585	0.01079
down (m)	-0.94923	-0.95826	0.13704	-0.00903	0.01370	0.00942

The third column of table 6.3 shows that there is a relatively large standard deviation in the estimated scale, on the order of about 13%. This is expected due

the the weak observability of scale and the short trajectory of only 20 poses used for these simulations. However, the last column shows that the mean difference in estimated scale is on the order of only 1%. This along with the analysis of the variance of the mean of the runs from column four and five suggest that the estimates are unbiased and consistent.

6.7 Chapter summary

This chapter investigated how the inertial observations can be used to make the scale of a map produced by bearing only SLAM observable.

The equations linking the poses in an estimated trajectory from both bearing only landmark observations and inertial observations were examined to determine where the scale of the map appears in their interaction. Rearranging of these equations was performed to obtain a scale equation which shows that the bearing only map scale can be observable from just three consecutive poses and gives a clue to the situations under which it can be estimated.

Furthermore a way of determining the accuracy of the estimated scale from the scale equation was also developed along with how to calculate the correlation of the scale equation over different poses. A way to extend this scale estimate over more than three poses by keeping track of the correlation between consecutive estimates was also discussed.

The relative advantage of explicit parameterisation of the scale of the map as well as an implicit parameterisation in the distances in the map its self are discussed and simulations are provided for illustration.

A discussion of how to implement this scale estimation on 6DoF system and the possible ways to parameterise scale was presented. Simulations were performed to evaluate the performance of the scale estimation in both the one dimensional and the full six degree of freedom cases.

Chapter 7

Practical Techniques

In the previous chapters the theory of pre-integrated inertial delta observations was developed and analysed, however there are some practical considerations that must be made in order to be able to implement them as part of a real system.

These include tools to analyse the estimates obtained to assess if the filter is operating as expected and methods to find and match visual observations including various ways of detecting and removing outliers. Techniques to initialise new pose estimates both for a starting point in the solution and to obtain initial linearisations are needed as well as graph optimisations for both speed and stability.

7.1 Observation edge energy

The graph edge energy for an observation is shown in equation 7.1. E_i is the edge energy for observation i , z_i is the value of the observation and x are the estimated states. Edge energy is a good measure of how well the estimated states agree with the observations made.

$$E_i(z_i|x) = -\ln(p(z_i|x)) \quad (7.1)$$

The edge energy is used in [22] and [23] with the idea that minimising the

sum of all the edge energies in the graph results in the maximum a posteriori (MAP) solution. A similar concept is used in [37] with a weighted square cost function (equation 7.2) that is closely related to the edge energy and is viewed as a constraint.

$$v_i^T R_i^{-1} v_i \quad (7.2)$$

R_i is the covariance of the observation noise and v_i is the innovation for the observation shown in equation 7.3 where $h_i(x)$ is the observation function.

$$v_i = z_i - h_i(x) \quad (7.3)$$

The weighted square cost function is related to the edge energy since for the assumption of a Gaussian distribution the edge energy can be expressed as shown in equation 7.4.

$$E_i(z_i|x) = \frac{1}{2} v_i^T R_i^{-1} v_i \quad (7.4)$$

The relation between the edge energy and the normalised innovation can be seen from equation 7.2 and can be easily calculated for multi-dimensional observations as the observation covariance matrix R_i takes care of any correlation between the observation dimensions.

7.1.1 Pre-integrated inertial delta observation edge energy

The implicit handling of observation correlation when calculating observation edge energies is useful for pre-integrated inertial observations as the three components are highly correlated due to their reliance on shared inertial observations. The inertial delta observation covariance matrix calculated in algorithm 4.3 shows this correlation and can be used in the edge energy calculations.

Figure 7.1 shows how the various estimated vehicle states are connected

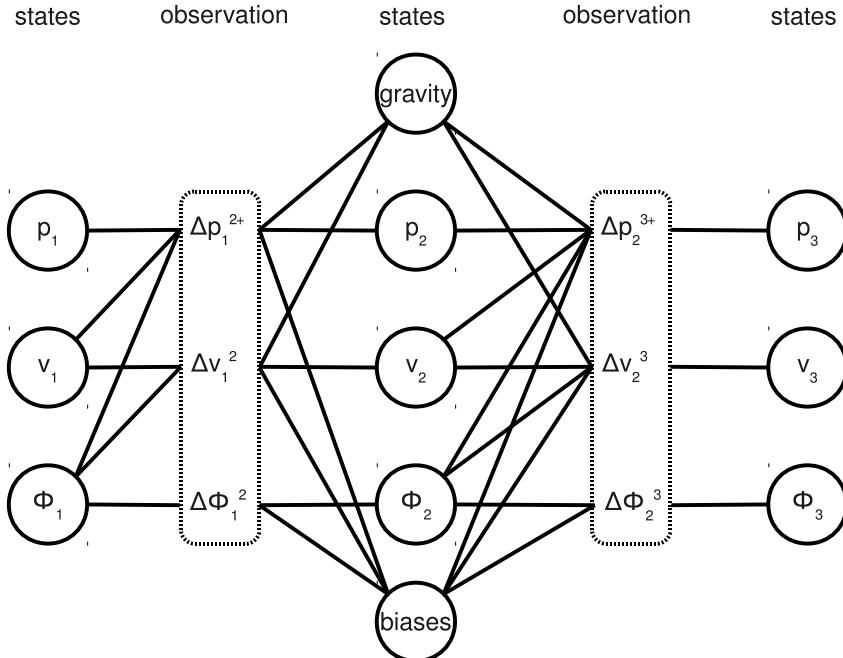


Figure 7.1: Graphical structure of pre-integrated inertial delta observations showing the graph edges connecting the observation nodes with the estimated vehicle state nodes. The gravity and bias states are common to all poses.

through pre-integrated inertial delta observations. Note how the observations only connect the states of one pose to the next pose allowing its edge energy to be calculated for each $[\Delta p^+, \Delta v, \Delta \phi]^T$ triplet independently. This is possible since the edge energy of the observations are conditionally independent given the estimated state values as can be seen in equation 7.1.

$$\frac{1}{2} \left(\begin{bmatrix} \Delta p^+ \\ \Delta v \\ \Delta \phi \end{bmatrix} - \begin{bmatrix} h_{\Delta p^+}(x) \\ h_{\Delta v}(x) \\ h_{\Delta \phi}(x) \end{bmatrix} \right)^T R^{-1} \left(\begin{bmatrix} \Delta p^+ \\ \Delta v \\ \Delta \phi \end{bmatrix} - \begin{bmatrix} h_{\Delta p^+}(x) \\ h_{\Delta v}(x) \\ h_{\Delta \phi}(x) \end{bmatrix} \right) \quad (7.5)$$

Equation 7.5 shows how the observation energy for a pre-integrated inertial

delta observation is calculated. The $h_{\Delta p^+}(x)$, $h_{\Delta v}(x)$ and $h_{\Delta \phi}(x)$ predicted observation values are obtained from equations 4.12, 4.13 and 4.14 respectively. The joint observation covariance matrix, R , is from algorithm 4.3.

The edge energy for observations is not only useful for determining what graph relaxations are required to obtain the correct state means for estimates but can also be used to assess data association and evaluate the result of a step correction in an iterative solution which is of interest to this application where non-linear observation functions and constraints are used.

7.1.2 Observation edge energy distribution

One thing to note when inspecting the observation edge energies of a solution is that even though they can be thought of as a kind of normalised innovation, their distribution may not necessarily resemble normally distributed white noise of the appropriate variance as expected. There are two reasons why this may be the case.

One reason is that the observations may be multi-dimensional but the observation energy calculation returns a single scalar value. This value is the magnitude of the multi-dimensional innovation for that observation. When the observation energy is plotted the graph may appear to show correlation in the innovations but this is not necessarily the case if the full dimensional innovation is examined.

The distribution of edge energies may also appear to have much lower variance than expected. This is particularly common on the gyro observations when only a few features are observed with the cameras. This does not mean that the gyro observations are more accurate than expected, but simply that the orientation estimate produced is almost entirely obtained from the gyro observations.

This is particularly common in the considered SLAM applications as for even low cost gyros the uncertainty in the orientation change over the period between images will be very small. If only a few visual observations are made most of the visual information will go into constraining the position of the camera. Due to the possibly low angular accuracy of the visual observations (maybe 0.1°)

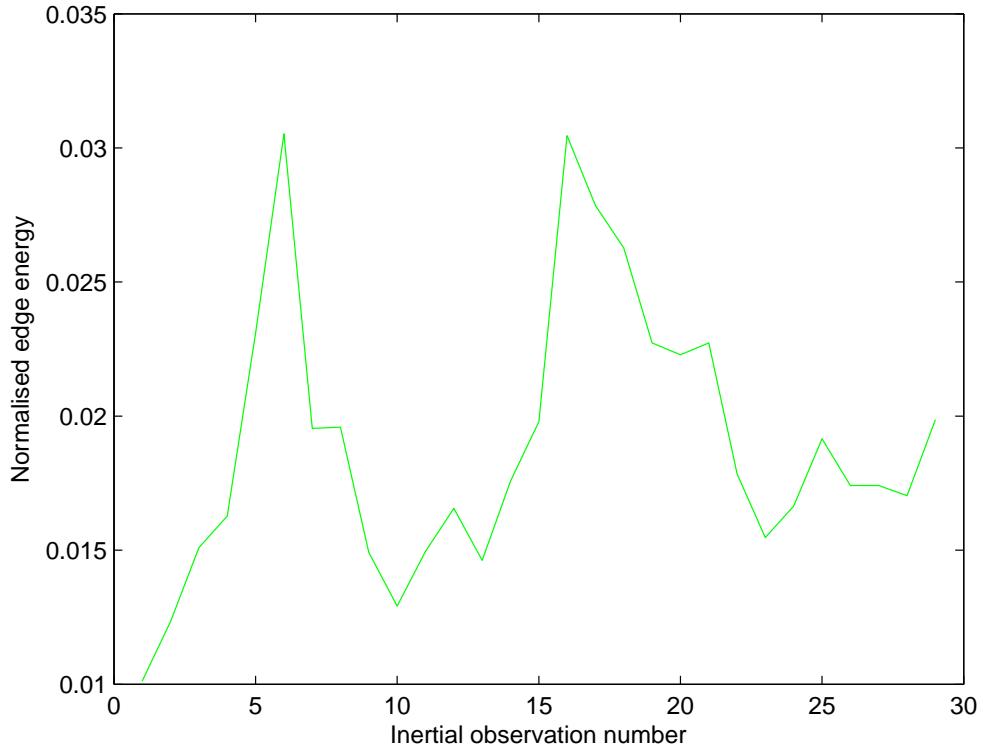


Figure 7.2: Normalised observation edge energy for the delta attitude observations using the estimated poses. Note how the energies are very small, this is because the rotation information provided by the gyro observations is much more accurate than that provided by the visual observations so the estimated attitude is mostly derived from the gyro observations.

the SLAM solution for the vehicle attitude will pretty much be reliant on the gyro observations. This should be reflected in the estimated uncertainty of the relative orientation between poses, but as the observation energy calculations are conditioned on the estimated state means, as shown in equation 7.1, it is not reflected here.

This can be seen in figure 7.2 where the observation edge energy for the delta attitude observations is shown after a simulated 30 pose run using inertial and visual observations. The edge energies for the delta attitude observations are so small as the angular resolution of the cameras is very low compared to the accuracy of the gyro observations that the estimated attitude is almost completely

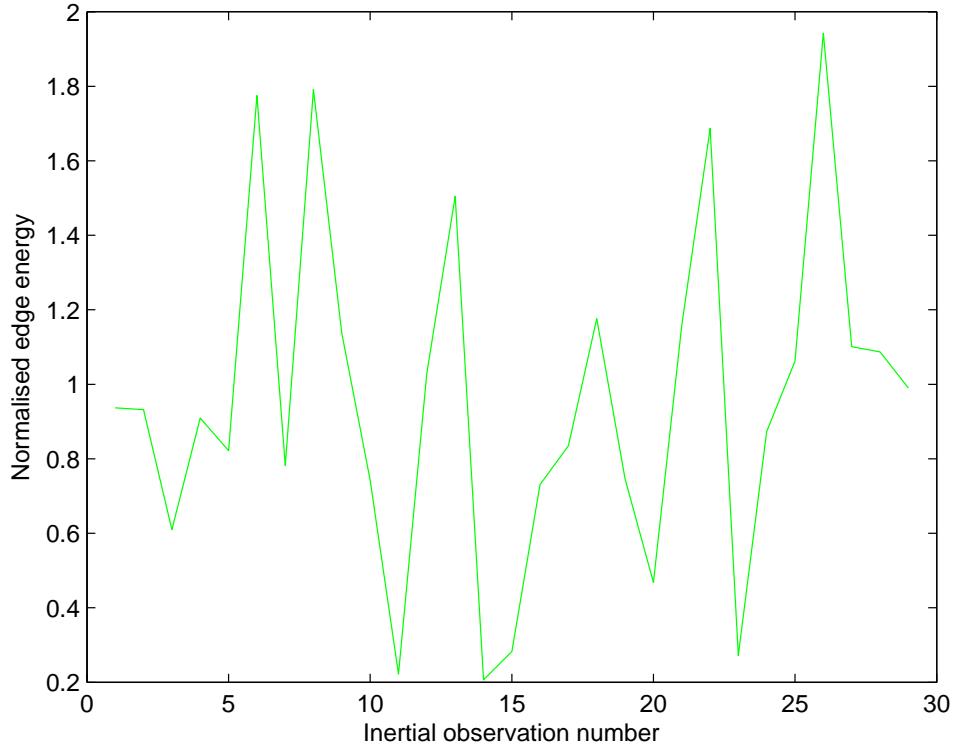


Figure 7.3: Normalised observation edge energy for the delta attitude observations using the true poses. Note how this distribution is much more consistent with what would be expected for innovations resembling zero mean white noise than what is shown in figure 7.2.

derived from the gyro observations.

To obtain the true inertial observation innovations, the observation energy calculations need to be conditioned on the true state means instead of the estimated means. This is done in figure 7.3 which uses the same delta attitude observations as in figure 7.2, but the true pose means are used for the energy calculations instead of the estimated means. This distribution is much more consistent with what would be expected for innovations resembling zero mean white noise.

7.2 Feature extraction and matching

There are many different feature extracting algorithms available to find visual features in an image suitable for tracking. One of the oldest and simplest techniques is Harris corners [52]. Recently a number of new feature descriptors such as SIFT [39], SURF [5] and CenSur [1] have been developed however most of them still rely on using extrema in the second derivative to locate the features just as Harris corners do. Also these descriptors are usually computationally intensive to calculate and have fairly low discriminative power.

These newer descriptors are good for matching a whole image with a large number of features to another whole image with a similarly large number of extracted features even over large viewpoint changes. However, one to one matching or tracking of features using these descriptors is usually not reliable as their descriptors are only weakly discriminative.

For SLAM problems the images between which features are to be tracked are usually taken from locations that are close together and therefore the change from one image to the next is relatively small. In these situations optical flow algorithms, such as Lucas-Kanade (LK) [40], work very well and are fast to compute in its pyramidal form. Therefore the main tracking algorithm used in this thesis is the optical flow tracking of Harris corners using the LK algorithm.

Descriptor matching such as that used with SIFT and SURF features is still useful for SLAM for loop closure detection where large difference in pose may be observed and optical flow will not work. These descriptors can also be used for loop closure detection, for example using FabMap [12].

Any visual feature matching technique is going to produce a percentage of incorrect matches. These outliers can have a detrimental effect on filter estimates as the large innovations will have a disproportionate effect on the estimated means. Fortunately there are a number of methods to detect and remove outliers in the visual data association.

7.2.1 Using pose estimates for data association

One method is to use the filter estimates themselves to predict observations and test data association, however there are a number of issues with this method.

Firstly, using the filter estimate for data association requires the covariance of the estimates to be recovered which can be computationally intensive if information from filters are used. Secondly, if the estimates from the filter are wrong then using this technique practically guarantees that all data associations from this point forward will be incorrect therefore making it impossible for the filter to recover from the error. Finally, when the filter starts and the initial conditions of the platform are unknown or for newly initialised features which are not well localised yet, the uncertainties for the estimated positions of features in the new frame will be so large that they will not be able to effectively associate the observations.

Therefore it would be useful if another source of information that is not affected by the, possibly inaccurate, filter estimate could be used to help in data association.

7.2.2 Fundamental matrix outlier rejection

One way to prune incorrectly associated visual feature matches is by using the fundamental matrix. The fundamental matrix is a way of representing the geometry between the view points of two projective cameras. It is independent of the structure of the scene being observed but can be calculated purely from matched feature observations between the two views.

The fundamental matrix for an image pair can be used to detect feature matches that do not lie on an epipolar line through the fundamental matrix constraint shown in equation 7.6.

$$x' F x = 0 \quad (7.6)$$

x and x' are the image locations of a pair of matched features in the first

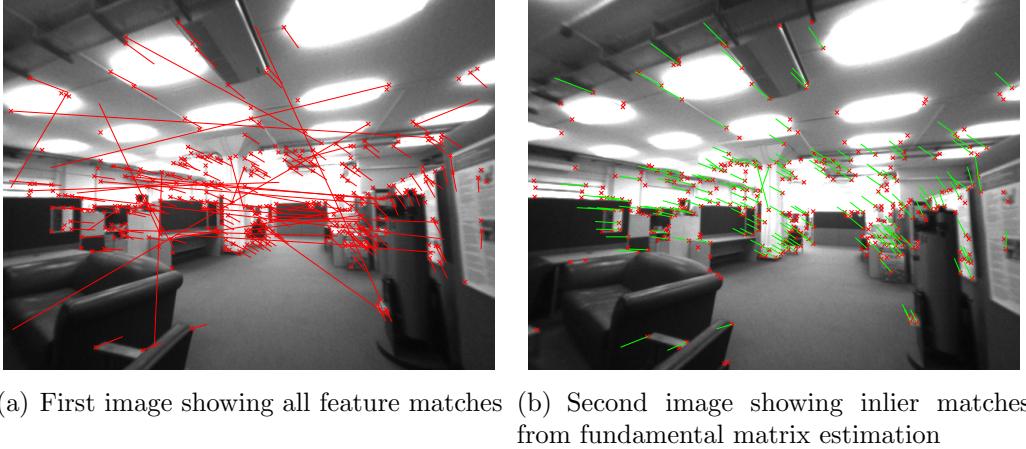


Figure 7.4: Two consecutive images showing matched features and pruning of outliers using RANSAC fundamental matrix estimation.

and second image respectively defined in homogeneous coordinates and F is the fundamental matrix.

One of the most popular methods for estimating the fundamental matrix is the 8-point algorithm [29]. This method is popular as it is simple to implement, fast to calculate and can be used as part of a RANSAC [21] routine to cope with uncertain data association. These benefits come from the linearity of the algorithm but it can still give results comparable to the non-linear methods given proper normalisation of the image points before processing [27].

Figure 7.4 shows an example of the rejection of outlier matches using the fundamental matrix calculated using RANSAC on the matched points. Figure 7.4(a) shows the first image with all the matches obtained using normalised cross correlation on extracted Harris corners, figure 7.4(b) is the second image with only the inlier matches remaining. Notice that even though the majority of outliers are removed, there are still two remaining incorrect matches across the white pillar in the centre of the image.

One problem with relying on fundamental matrix calculations for outlier rejection is that for many of the datasets used there are not always sufficient observations in an image for the fundamental matrix to be calculated robustly (at least 8 inliers are required for outlier rejection to be possible) and therefore the

pruning of incorrectly associated features can not be performed. The estimated poses of the cameras from the SLAM filter could be used to calculate the fundamental matrix but this could be problematic due to the same reasons explained in section 7.2.1.

7.2.3 Incorporating gyro observations into fundamental matrix estimation

If inertial observations are available, the gyro observations can provide the relative rotations of the camera between two adjacent images to a fairly high degree of accuracy. The accuracy of the relative rotation calculated from the gyro observations between two consecutive images can commonly be on the order of 0.001° which is insignificant when compared to the angular resolution of the cameras used which is commonly on the order of 0.1° .

The benefit of using gyro observations when estimating motion from camera observations to provide a more accurate solution and resolve any possible translation/rotation ambiguity [13] is not a new idea [49]. However in the past gyro observations have been used in a larger SLAM or SFM solution which requires recovery of all the feature locations as well as vehicle states. This is not helpful for visual match outlier rejection and will not work in the rotation only case where there is little or no translation as the recovery of the landmark locations will not be possible.

If instead these known rotations are used to constrain at least some of the degrees of freedom of the fundamental matrix, it can be estimated with fewer tracked visual features and does not require translation as landmark location recovery is not required.

This is possible since the fundamental matrix can be constructed as shown in equation 7.7 from [28]. The two K matrices are the camera intrinsic calibration matrices and the R matrix is the rotation between the two camera view points, which can be obtained from the gyro observations. This leaves just the epipole

e' to be estimated which has three dimensions but only two degrees of freedom. This is a reduction from the eight dimension, seven degree of freedom problem for normal fundamental matrix estimation.

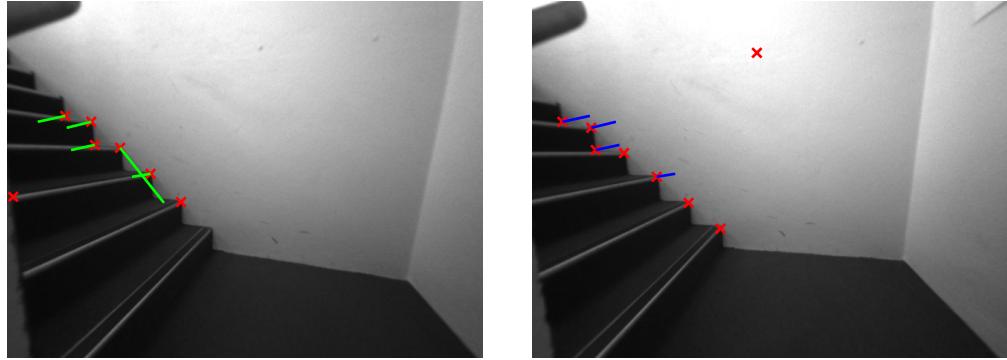
$$F = [e']_{\times} K' R^{-1} K^{-1} \quad (7.7)$$

Therefore, less matches are required if observations from gyros are used to constrain the fundamental matrix calculated from matched image observations. It is shown in appendix B that as few as three feature matches are required for the Fundamental matrix to be recovered in a linear way. As few as two matches are needed if a non-linear technique is used as the epipole location only has two degrees of freedom. These methods are derived from a modification of those presented in chapter 11 of [28].

This method assumes that the rotation estimate between two consecutive images from the gyro observations is perfectly known. However, in practice this is not usually a problem since estimated rotations over the short period between frames is known to an extremely high degree of certainty if gyro observations are used. For example, if the gyros from the Honeywell HG1900 are used, as in the set-up presented in chapter 8, even if the gyro biases are uncertain the standard deviation for each of the roll, pitch and yaw axes were on the order of 0.001° between consecutive images that were taken at 6.25Hz. This is insignificant when compared to the 0.2° angular resolution of the cameras used.

An example of the usefulness of the reduced requirement on the number of inliers is shown in figure 7.5. In this image sequence, only five matched features are obtained of which only four are inliers. Under normal fundamental matrix estimation using only the visual features the fundamental matrix between these two images could not even be calculated let alone used for outlier rejection. Since only three inliers are required when gyro observations are used this method successfully identifies the outlier in these matches.

In fact, using gyro observations will also result in much more accurate recovery of the camera projection matrices as well due to the more accurate estimate



(a) First image showing all feature matches (b) Second image showing inlier matches
from fundamental matrix estimation using gyro observations

Figure 7.5: Two consecutive images showing matched features and pruning of outliers using RANSAC fundamental matrix estimation with gyro observations.

of the rotation between the images. In many situations recovery of the projective matrices suffer from a large ambiguity between translation and rotation of the cameras, especially for narrow field of view cameras. The use of rotation information from the gyros resolves this ambiguity.

7.2.4 Using observation edge energies to remove outliers

The fundamental matrix test removes the vast majority of outliers from the tracked visual features, however some mis-associated points will always remain. This is especially the case when only a few observations are available in an image as at least eight correctly tracked features are required to be able to reject outliers even when the non-linear seven point algorithm is used.

As well as using gyro observations to help with data association, it is possible to use the accelerometer information aided with observations from other frames if further pruning of mis-associated observations is conducted within the estimation of the filter. Observations edge energies for the visual observations can be calculated, in a similar way as was described for the inertial observations in section 7.1, to help detect incorrect data association.

After observations have been added to the graph and a graph relaxation has

been performed to obtain updated mean estimates for the states, the edge energy for the visual observations can be calculated and compared to the observation uncertainty to determine if they are correctly associated. A large normalised edge energy ($> 2\sigma$) would suggest that the data association for the observation is incorrect.

7.3 Pose initialisation options

For visual SLAM the vision equations are non-linear therefore an initial estimate of the new pose and landmark locations are required to use as a linearisation point and a starting point for the information smoothing or Kalman filter solutions. Accurate initial estimates are important as even with relinearisation, initialising a new pose without an accurate initial estimate can cause convergence to be slow or failure of convergence altogether.

For initialisation of new features the best option is usually to obtain an initial range through triangulation either from stereo observations or from monocular observations from the first two poses. However, for initialisation of a new pose estimate there are a number of options to choose from and each has its own advantages and disadvantages.

7.3.1 Prediction from previous filter estimates

The standard method for pose initialisation used in SLAM is by predicting the current pose from the previous pose of the vehicle using either a process model [14] or inertial observations [9, 33].

The most common process model used for SLAM is a constant velocity model. Constant velocity models simply take the previous position and velocity estimate of the vehicle and assume that it continues to travel at that velocity with no rotation during the period of prediction. An example of a constant velocity model prediction is shown in equation 7.8.

$$\begin{bmatrix} p_{t+1}^n \\ v_{t+1}^n \\ \phi_{t+1}^n \end{bmatrix} = \begin{bmatrix} p_t^n + v_t^n \Delta t \\ v_t^n \\ \phi_t^n \end{bmatrix} \quad (7.8)$$

Constant velocity process models are generally only valid for high observation rates, for example 30Hz video, and low vehicle dynamics as the assumption of constant velocity and no rotation during the prediction period is normally only approximated in these situations. This method also requires an accurate prior estimate of velocity, which is usually only obtained after a few poses have already been estimated. Most implementations get around this by requiring the vehicle to be stationary, or approximately stationary, when the filter is started.

However, inertial observation based prediction models, such as the one presented in equation 4.1, can handle acceleration and rotation during the prediction period as this is observed by the IMU. The major problem with inertial prediction is that the initial velocity, gravity vector, and therefore attitude and IMU biases must already be known to a degree of certainty. These estimates can be obtained after a number of poses have been estimated, or after an initialisation routine, but use of inertial prediction models from the first pose without initialisation is not possible.

7.3.2 Initialisation of a new pose from image observations

The pose initialisation options presented in section 7.3.1 rely on the estimates generated by the filter. These are fine when accurate filter estimates are available, but when they are not, or when it is desired to isolate possible errors in the filter, as described in section 7.2, pose initialisation methods purely using observations are available.

Recovery from fundamental matrix estimation

The simplest method for initialising a new pose from visual observations alone is by recovering the camera poses from the fundamental matrix between the two

images. The use of the fundamental matrix for outlier rejection in data association was presented in 7.2.2, this same fundamental matrix can be used to extract the relative camera positions and orientations as presented in [28].

One problem with the fundamental matrix is that scale estimates are not available, it can provide the relative rotation and translation direction of the camera, but not the distance of that translation. If scale information is required it can be obtained from calculating a trifocal tensor between three consecutive frames to carry the scale information from one pair of poses to the next.

Fundamental matrices may also not be well conditioned which means that the extracted poses may not be very accurate. This is especially common for narrow field of view cameras where there may be an ambiguity between sideways translation and yaw that can occur when features are far away from the camera, as well as forward motion of the camera.

The estimated poses from the fundamental matrix can be refined by using bundle adjustment on the poses and features, a previous pose can also be included to resolve the scale ambiguity if desired. However ill conditioning can still be a problem depending on the number and distribution of visual observations available.

Including gyro observations into visual pose initialisation

If inertial observations are available then these can be beneficial to this problem. Gyro observations can be used with the fundamental matrix calculation in order to improve conditioning and resolve any possible translation/rotation ambiguity. This is possible as gyro integration can be performed without knowing the initial conditions, providing an accurate relative orientation estimate.

The method for incorporating gyro observations into fundamental matrix calculations is introduced in section 7.2.3, and derivations behind the equations are presented in appendix B.

This method can be used for both the stereo and monocular case. As well as resolving the translation/rotation ambiguity problem, the use of gyro observations

in the fundamental matrix calculation also reduces the problem to the estimation of three variables with two degrees of freedom (three degrees of freedom in the stereo case) so less feature correspondences are required.

Stereo pose alignment techniques

If stereo vision is used then the new pose can be estimated accurately purely from the stereo observations provided there are sufficient tracked features close to the camera. One possible technique for doing this is the stereo visual odometry method presented in [30].

The advantage of the stereo technique is that it provides the distance of translation as well as the direction unlike the fundamental matrix method, however it can only be used with stereo images.

7.4 Optimising information smoothers for long trajectories

When a dataset for a long trajectory is being processed, solving for the joint distribution for the whole dataset including all poses and landmarks is incredibly computationally expensive.

When the solver is traversing along the trajectory the newer poses and landmarks are usually the ones who's estimates are changing significantly, with older landmark and pose estimates remaining almost the same. There are a number of ways to take advantage of this fact when designing a solver for the SLAM problem that is more computationally efficient as in most cases only the newer poses and features can be updated to get an accurate approximation of the joint solution.

There are a number of techniques that involve marginalising out the older poses, such as in traditional EKF SLAM [16], or various forms for conditioning on older poses at stages during the solve [20, 23].

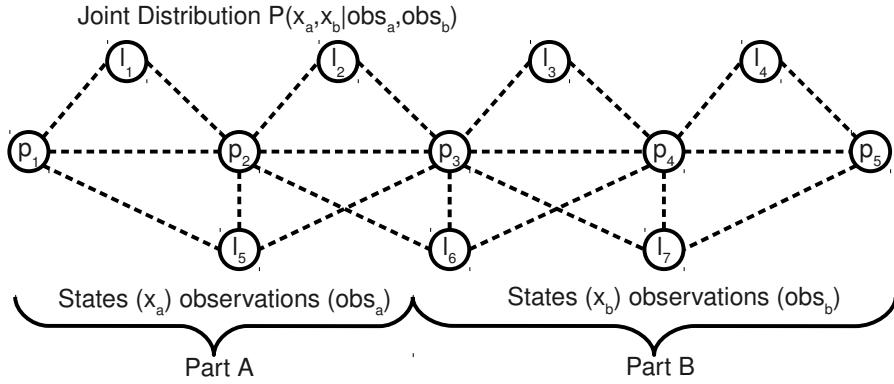


Figure 7.6: Joint probability distribution of a trajectory and map. The states in part A and B of the map are estimated together using observations from both part A and B of the map.

Figure 7.6 shows an example graphical SLAM problem with a map and trajectory that is split into 2 parts, part A and part B. These are the states that will be solved for if the joint solution is to be computed.

If only the states in part B of the map are expected to change as new observations are added then the whole joint distribution does not need to be recalculated. There are a number of ways to change the problem to allow this to be done more quickly.

7.4.1 Marginalisation

If the states in part A of the map are marginalised out, they, and their associated observations, do not need to be considered by the solver any more. An example of marginalisation of the graph shown in figure 7.6 is shown in figure 7.7.

Marginalisation is fast for state space implementations, such as the EKF, but computationally intensive for graphical or information based solvers, but it is by no means intractable.

The advantage of marginalisation is that it does not fix the means of the previ-

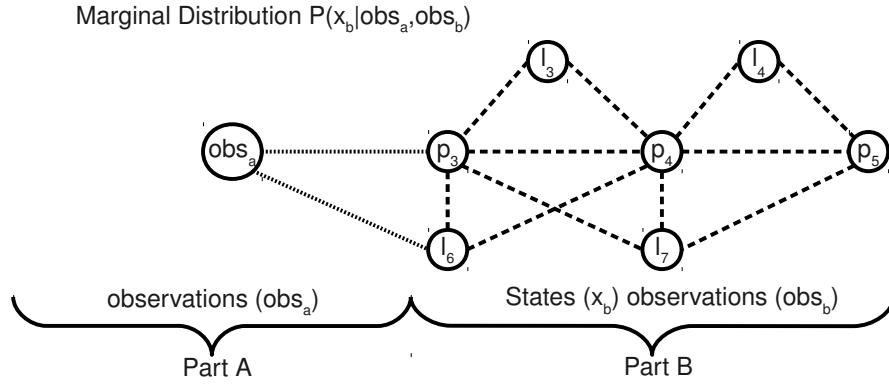


Figure 7.7: Marginal probability distribution of a trajectory and map. The states in part B of the map are estimated using the observations from part B as well as the estimated states from part A with their associated uncertainties.

ous states that have been marginalised out. Therefore biasing or over confidence in the remaining states is not a problem as long as the linearisations of the past observations still remain valid.

However a problem with this approach is that previous observations can not be relinearised in the future, but this should not be too much of a problem if the states are very well known. Also if errors in the solution from previous poses are extreme, such as from poor data association or observability problems, these errors will remain in the solution and could cause a problem for the solver. This is especially important in the cases where the solution fails and diverges as it will not be possible to recover from such an event.

7.4.2 Conditioning

Another option is to condition the states in part B of the map on the current estimated means of the states in part A. An example of conditioning of the graph shown in figure 7.6 is shown in figure 7.8.

This is computationally very easy to do with graphical or information based

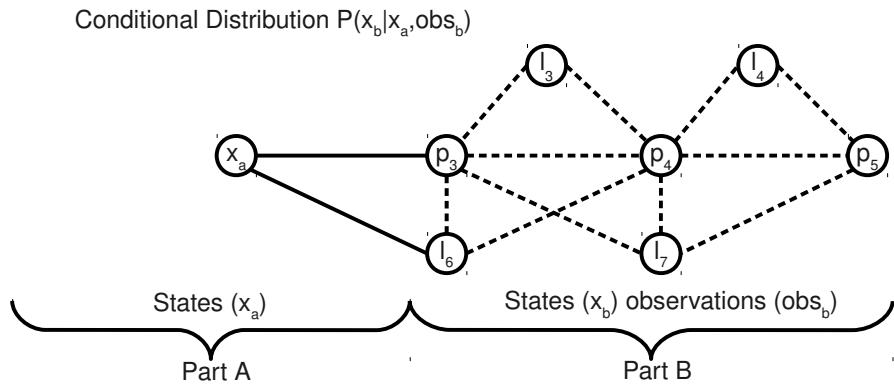


Figure 7.8: Conditional probability distribution of a trajectory and map. The states in part B of the map are estimated using the observations from part B assuming the estimated states in part A are known with complete certainty.

solvers and can work well when past poses are known to a high degree of accuracy. However, problems can arise when past poses are not well known as this can introduce accumulating errors quickly into the solution, especially where the gravity vector and velocity need to be known accurately for the inclusion of inertial observations.

Also past errors or failures in the filter will still remain in future solutions which would make it impossible to recover from problems such as solution divergence or unobservability. Loop closure is also problematic as the loop trajectory could not be adjusted past the conditioned poses when adjustment after loop closure is attempted.

All these problems are caused by conditioning resulting in the filter being overconfident in the accuracy of its solution.

7.4.3 Sliding window forced independence

A third option that is particularly well suited for the pre-integrated inertial/vision applications considered in this thesis is to simply throw away the observations

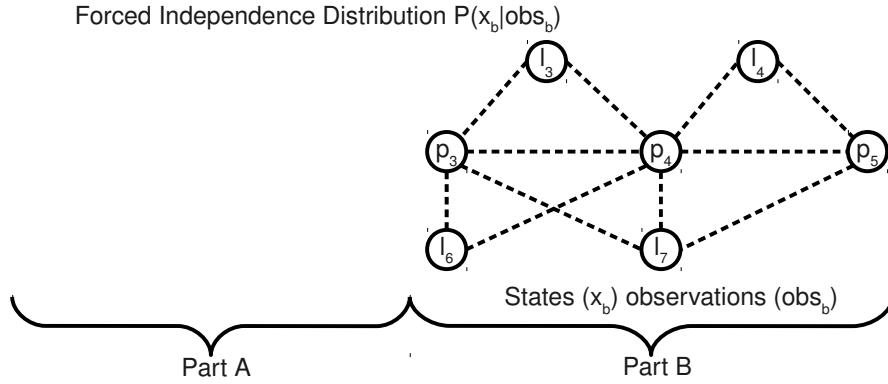


Figure 7.9: Forced independence probability distribution of a trajectory and map. The states in part B of the map are estimated using the observations from part B completely independent of the observations or estimates from part A.

and estimates associated with old poses and landmarks. This way a solution for the recent poses and landmarks will be found that will be completely independent from past observations and estimates.

This form of forced independence for the solution has a number of advantages for not only computational load but also robustness and flexibility of the solution. An example of the result of forcing part B of the graph shown in figure 7.6 to be independent from part A is shown in figure 7.9.

Because all past observations, both inertial and visual, are thrown away, the gravity vector and initial velocity estimates are formed purely from the observations made within the poses currently being solved for. This independence also means that poses can be solved for at any point in a trajectory without having to solve right from the beginning of the dataset every time which makes recovery of portions of interest in the dataset fast and efficient.

Robustness is also introduced as data association errors, portions of unobservability or, in extreme cases, divergence of the solution do not affect later portions of the trajectory once they are outside the solver window. This allows automatic

recovery from errors if and when they occur.

As the solutions are independent from previous portions of the trajectory, loop closure can also be performed without the need to update all the pose and landmark estimates back along the entire loop. This updating can be very computationally intensive and often leads to marginal local improvements in accuracy of the map.

One downside of forced independence is that the final solution may not be as accurate as the joint solution as far fewer observations are involved in arriving at an estimate and the removal of past poses will lead to a dead-reckoning type drift in the position. This should usually not be too much of a problem as the observations made close to a state are the ones that have the most influence on its solution and these are still retained.

7.5 Chapter summary

This chapter discussed some of the practical implementation issues that need to be considered when using pre-integrated inertial delta observations and visual observations in a real world SLAM solution.

The concept of normalised observation edge energy with respect to the pre-integrated inertial delta observations and how to calculate it was presented. The relation between observation edge energy and normalised innovation was also discussed with an investigation of the distribution of edge energies in the results from a SLAM implementation and why they may differ from an expected white noise innovation sequence.

A fast method for feature extraction, matching and outlier rejection was discussed using Harris corners, Lucas-Kanade optical flow and the fundamental matrix. A method for further rejection of mis-associated visual observations by using the observation edge energy during estimation in the filter, and the possibility of incorporating gyro observations in the initial visual data association step outside of the filter was also presented. This method reduces the number matched

features required for outlier rejection as well as providing more accurate camera pose estimates from the fundamental matrix.

The need for an initial estimate of a new pose for determining a linearisation point for visual observations and a starting point for the SLAM filter was explained. A number of possible ways of obtaining this initial estimate were presented along with the advantages and disadvantages associated with each.

Finally, the details of the solving method used for graph relaxation are discussed before possible ways of optimising the solving of the trajectory estimation. Traditional graph marginalisation and conditioning operations are explained and then an alternative method of forcing independence between sections of the map and the possible advantages of it are described.

Chapter 8

Experiments

In order to test the performance of pre-integrated inertial delta observations in real world situations datasets were collected on a human mounted sensor suite walking in and around buildings. This set-up was used to obtain the type of data that would be expected from a fire fighter or similar rescue worker using a system to keep track of their movements.

8.1 Experimental set-up

8.1.1 Sensors

The sensor suite contains a Honeywell HG1900 IMU which provides inertial observations at 600Hz and a Point Grey Research Bumblebee2 stereo camera unit which has been fitted with 2.1mm wide angle lenses. The wide angle lenses were used to make it possible to obtain more useful observations in confined areas such as hallways and staircases where there are usually few distinct visual features and provide increased parallax for landmark observations when moving forward. Stereo images were recorded at a frame rate of 6.25Hz.

Figure 8.1 shows a photo of the sensors used. Tables 8.1 and 8.2 show the specifications for the IMU and cameras respectively. This unit was hand held and carried around the building for the datasets collected to simulate the kinds



Figure 8.1: Sensor suite used to obtain the walking datasets. The Point Grey Research Bumblebee2 stereo camera can be seen on the top of the unit with the Honeywell HG1900 IMU mounted just below it.

Table 8.1: HG1900 IMU measured specifications

Sampling rate	600 Hz
Accelerometer noise	$0.0775 \text{ ms}^{-2}(1\sigma)$
Gyroscope noise	$0.001 \text{ rads}^{-1}(1\sigma)$
Accelerometer bias stability	$\pm 0.003 \text{ ms}^{-2}$
Gyroscope bias stability	$\pm 6.0e^{-5} \text{ rads}^{-1}$

of observations that would be made by a human mounted system.

8.1.2 SLAM implementation

The SLAM algorithm chosen for these experiments is a graphical implementation of an information smoother using sliding window forced independence as described in section 7.4.3. The graphical form information smoother was chosen to allow

Table 8.2: Point Grey Research Bumblebee2 with 2.1mm lenses specifications

Sampling rate	6.25 Hz
Focal length	2.1 mm
Resolution	640 × 480 pixels
Field of view	97° × 80°
Angular resolution	0.2° (at centre)
Stereo baseline	12 cm

relinearisation of the visual observations to be performed.

A sliding window size of 30 poses was chosen as it provided a computationally tractable number of observations to process while still providing accurate results for the datasets used. Once the limit of 30 poses has been reached in the filter, every time a new pose is added, the oldest pose and any observations associated with it are removed. The second oldest pose is then conditioned on its last estimated position and attitude to provide the new body reference frame. This also allows the removed poses to be plotted on the same axis as the new poses so that the full trajectory can be constructed. The gravity vector estimate is not conditioned on the removed pose so this is only a redefinition of the reference frame, not a conditioning on the past estimates.

Features were extracted using Harris corners and matched with Lucas-Kanade optical flow before outlier rejection using RANSAC to calculate the fundamental matrix between consecutive images. This method was chosen as it is fast and simple while still providing a sufficient number of well localised tracked features. Once a feature is lost by the optical flow tracker it is not searched for again so the average number of consecutive poses that features are tracked for is about five to seven images in the datasets used. Further data association rejection is also preformed by using the visual observation edge energy test as explained in section 7.2.4.

The inverse depth representation [42] for visual features is used with the feature parameterised in the body frame of the first pose it is observed from. One

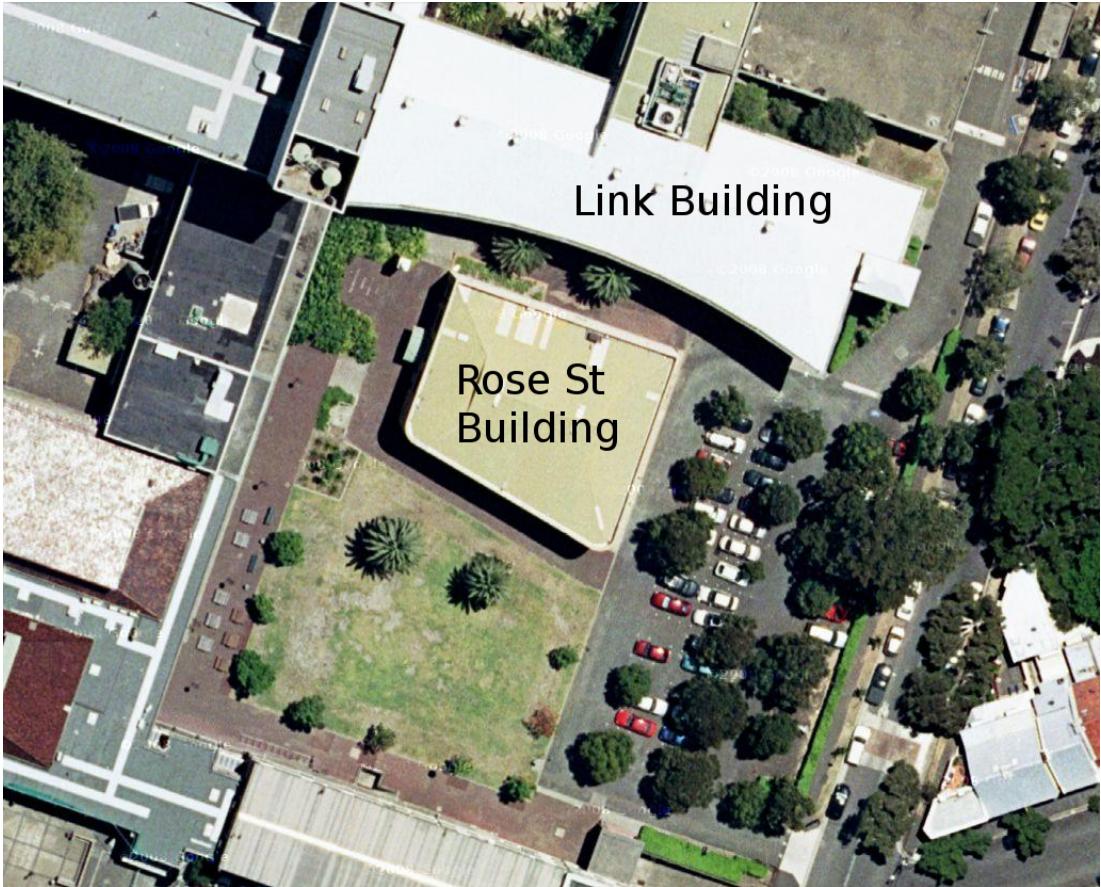


Figure 8.2: Satellite image from Google Maps showing the buildings in the area that the datasets were collected. The building in the middle with the green roof is the Rose St. building and the white building just north of it is the Link building.

difference from the implementation in [42] is that no arbitrary range prior is used for the features, they are either initialised with just the two observations from the stereo pair or, in the monocular case, from the observations from the first two observed poses. Without a prior on the range the biasing effect of this prior on the map and trajectory is removed [41].

Loop closure is not attempted and as soon as a feature is not tracked from one frame to the next it is not acquired again but still remains in the map. If this feature is found again by the feature extractor it is initialised as a separate feature. This method is used as it is fast to implement as extracted features do not have to be compared to all features in the map for matching, just the features



Figure 8.3: Example image of the environment that the datasets were taken in. This image is taken from the dataset shown in figure 8.26 and is of the downstairs area inside the Rose St. building.

tracked by optical flow from the previous frame. Also the problem that can occur if a new feature is incorrectly matched to a feature already in the map causing incorrect loop closure is removed.

Detection of loop closure and re-alignment of the map as a result is a possible extension as discussed in section 9.2.2.

8.1.3 Experimental location

The datasets were collected in and around the Rose St. Building and the Link building at the University of Sydney. Figure 8.2 shows a satellite image of the area in which the following datasets were collected.

Figures 8.3 is an image taken from one of the datasets showing the downstairs



Figure 8.4: Example image of the environment that the datasets were taken in. This image is taken from the dataset shown in figure 8.26 and is of the outside area between the Link building and the Rose St. building.

area inside the Rose St. building in which much of the data was collected. Figure 8.4 shows the outside area between the Link building and the Rose St. building.

These two images give an impression of the type and quality of the images that are used in these datasets as well as the environment in which they were taken.

8.2 Initial testing

In order to test sensor calibration and if the SLAM implementation using pre-integrated inertial observations is performing as expected, a simple dataset moving between two marked locations going both forward and backwards was col-

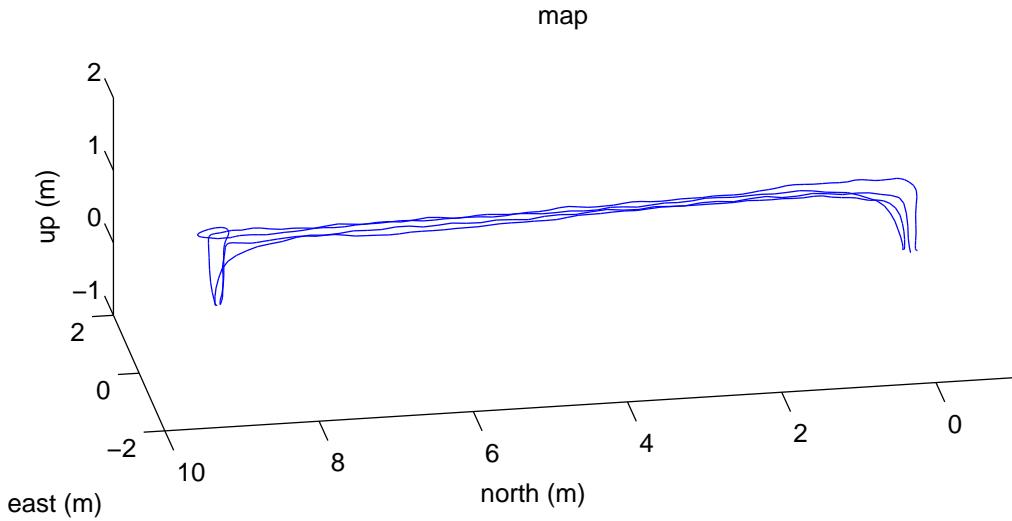


Figure 8.5: Estimated trajectory of the sensor platform when moving back and forth between two marked locations on the ground that are 9m apart.

lected. Two point were marked on the ground 9 metres apart and the sensor suite was carried between them first going forwards, and then backwards. Each time one of the two marked locations was reached, the sensor suite was placed on the ground at the marked location to an accuracy of a few centimetres.

The resultant estimated trajectory for this dataset can be seen in figure 8.5. It can be seen in this figure how the unit starts on the ground and is carried back and forth between the points, being placed back on the marks on the ground at each end.

Figure 8.6 shows the estimated position of the sensor unit in its north, east and down components taken in the body frame of the first pose. The north component of the position shows how the unit was carried from the starting position to the second point and back again twice. The first time was going forwards, and the second time backwards. This can be verified if the yaw component of the attitude during these periods is inspected in figure 8.7.

The estimated pose states during the 5 different stationary periods in this

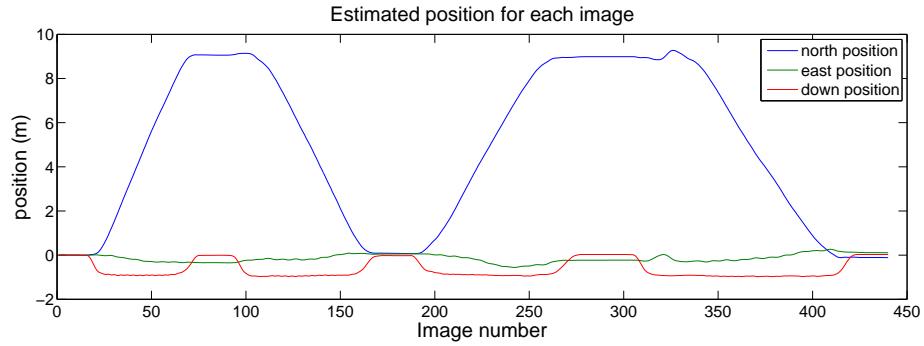


Figure 8.6: Components of the estimated position for the trajectory shown in figure 8.5. The north, east and down components are in the body frame of the first pose.

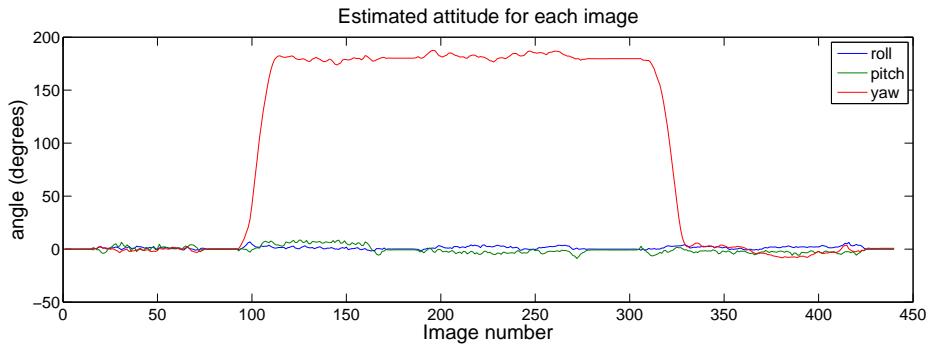


Figure 8.7: Components of the estimated attitude for the trajectory shown in figure 8.5. The roll, pitch and yaw components are in the body frame of the first pose.

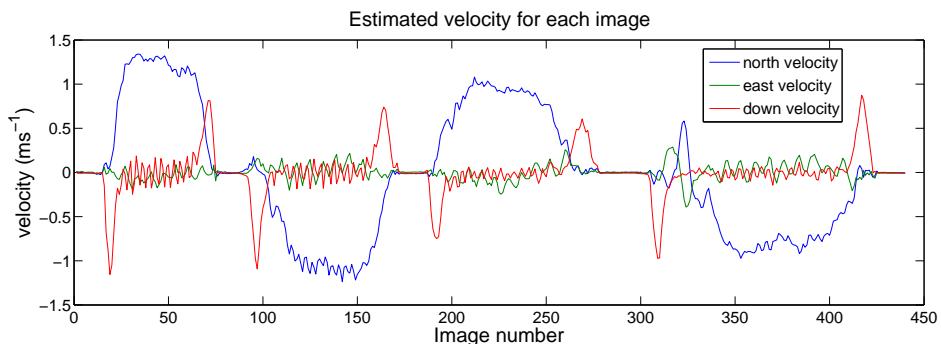


Figure 8.8: Components of the estimated velocity for the trajectory shown in figure 8.5. The north, east and down components are in the body frame of the first pose.

Table 8.3: Estimated position, velocity and attitude of the sensor platform during the stationary periods at the marked locations in the dataset shown in figure 8.5

Pose number	1	80	186	290	430
North position (m)	0	9.0653	0.0698	8.9882	-0.1074
East position (m)	0	-0.3422	0.0578	-0.2286	0.1142
Down position (m)	0	-0.0090	-0.0239	0.0284	0.0293
Roll ($^{\circ}$)	0	-0.1432	0.0802	-0.0115	0.0401
Pitch ($^{\circ}$)	0	0.0000	-0.6646	-0.7792	0.0516
Yaw ($^{\circ}$)	0	0.3209	-179.7655	179.6796	0.6761
North velocity (ms^{-1})	0.0046	-0.0090	0.0001	-0.0002	-0.0009
East velocity (ms^{-1})	0.0047	0.0094	-0.0080	0.0040	-0.0015
Down velocity (ms^{-1})	0.0033	-0.0040	0.0086	-0.0081	-0.0080

dataset are shown in table 8.3. The first pose is the body frame that is used as the navigation frame and that is why it has a perfectly zero position and attitude. The two points are 9 meters apart and it can be seen from pose 80 and 290 that the estimated location is within a few centimetres of this value. The solution does appear to drift slightly though with the location of the sensor platform at the final pose (pose 430) being estimated as 16 cm from the original starting position.

The estimated velocities for this trajectory appear to be very accurate with each of the components of the velocity at each stationary location being estimated as less than 1 cms^{-1} . Each of the attitude components are also estimated to less than 1° of the true value at each stationary position. This is within the accuracy of placement of the unit and is expected due to the high quality of the gyros in the IMU used.

For comparison to the visual/inertial solution, an inertial only estimate of the trajectory was also generated. The IMU was initialised using visual observations in a batch initialisation while stationary to obtain the initial conditions and biases of the unit. After this initialisation, only the inertial observations were used to obtain a navigation solution without the visual observations. The result can be seen in figure 8.9.

It can be seen from this figure that the inertial only observation appears to

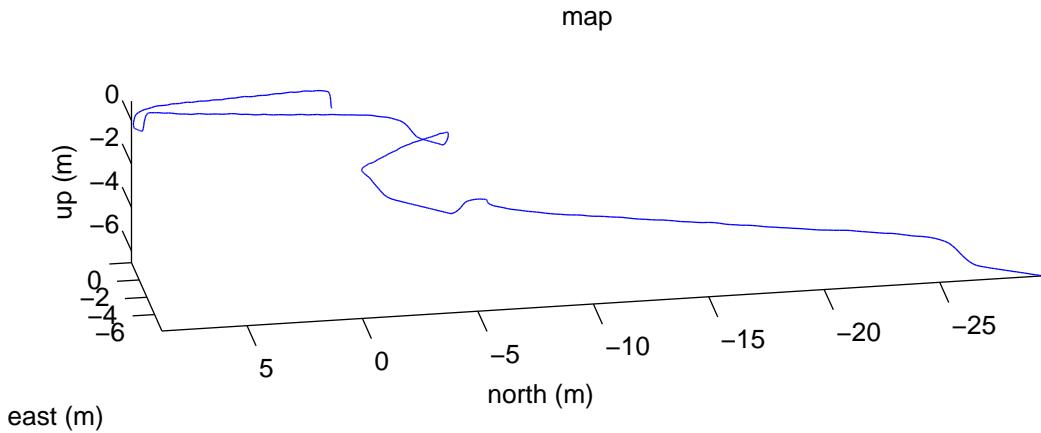


Figure 8.9: Estimated trajectory of the sensor platform when moving back and forth between two marked locations on the ground that are 9m apart using only inertial observations for comparison to figure 8.5. The initial conditions for the IMU were estimated while stationary but no visual observations were used to aid the inertial after it has started moving.

track the position of the unit during the first portion of the trajectory but soon after that it diverges from the solution and drifts away at an increasing rate. This is as expected for the IMU used as it is not a navigation quality IMU and from this the benefit of the visual observations to constrain the estimated velocity can be seen.

8.3 Second test

After the initial testing confirmed the system was working properly, a more sophisticated test was performed to assess its accuracy and drift characteristics.

Three locations were marked in a triangular pattern for this test as well as a section with lines at 30cm intervals. The trajectory went between the three marked locations a total of 3 times with a detour where the unit was placed on the ground and moved back and forth along the 30cm intervals on the last

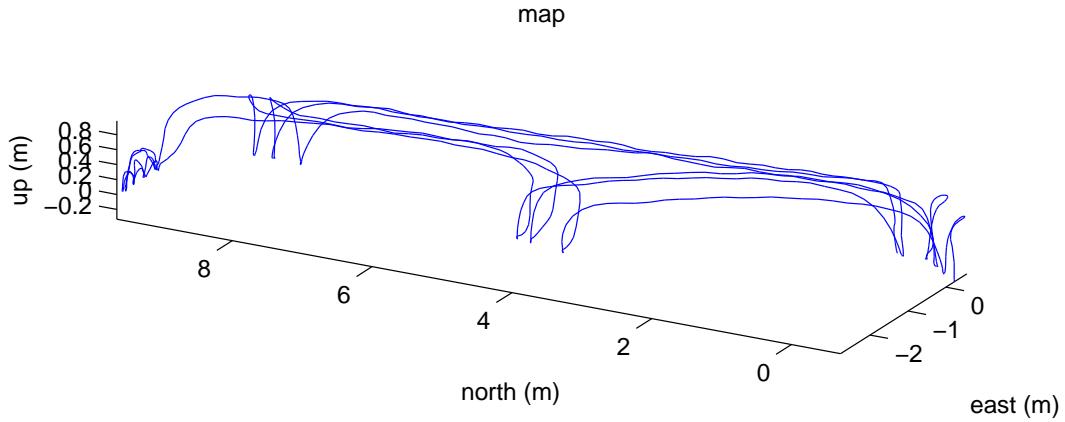


Figure 8.10: Estimated trajectory of the sensor platform when moving between the three marked locations in a triangular pattern and the section of 30cm intervals. The section with the 30cm intervals can be seen on the extreme left side of the figure.

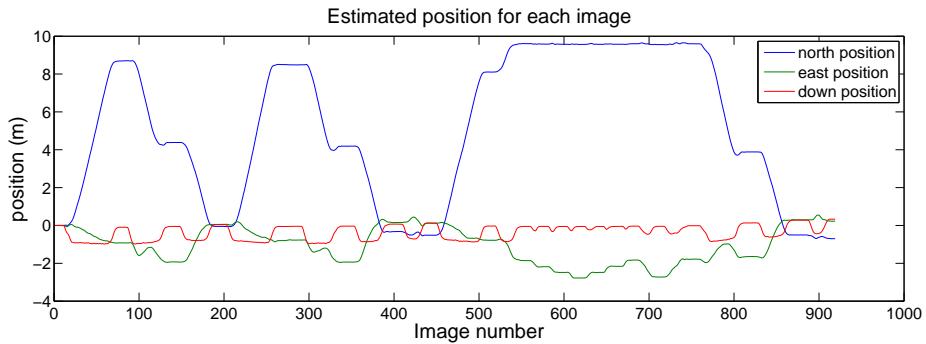


Figure 8.11: Components of the estimated position for the trajectory shown in figure 8.10. The north, east and down components are in the body frame of the first pose.

circuit. This dataset was collected over 150 seconds and covered a total distance of approximately 83 metres.

The plot of the estimated trajectory can be seen in figure 8.10. The components of the position, attitude and velocity estimates for this trajectory can be seen in figures 8.11, 8.12 and 8.13 respectively.

In order to analyse the accuracy and drift characteristics of the system the estimates of the position of the sensor suite when it was places on the marked

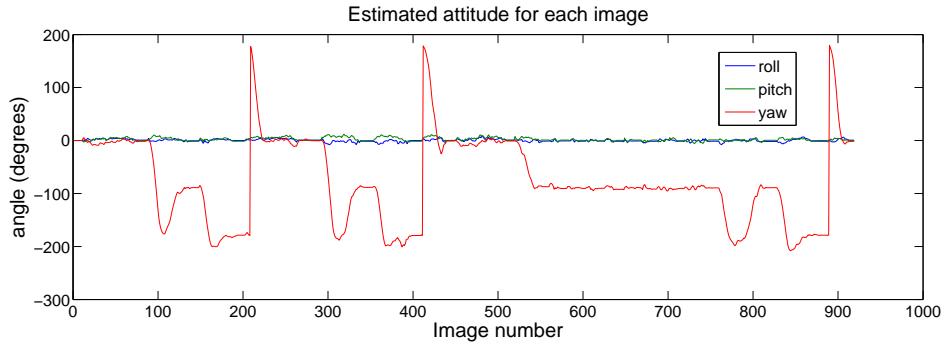


Figure 8.12: Components of the estimated attitude for the trajectory shown in figure 8.10. The roll, pitch and yaw components are in the body frame of the first pose.

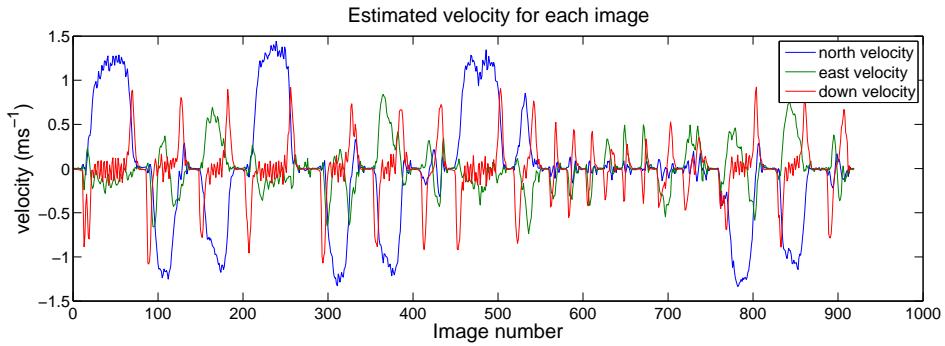


Figure 8.13: Components of the estimated velocity for the trajectory shown in figure 8.10. The north, east and down components are in the body frame of the first pose.

locations are shown in table 8.4 for the initial location. This is the position that the unit was in when the run was started and is therefore the origin used for the body frame based coordinate system.

It can be seen from table 8.4 that there is drift in the solution, especially the north component between the second and third time the unit was placed at these locations. This drift is similar to dead reckoning drift as no loop closure or lost feature reacquisition is used.

Even though dead reckoning drift is present in the position estimates, the instantaneous velocity estimates do not drift. Every time the unit is placed on the mark the components of the velocity estimates are on the order of $1\text{cm}\text{s}^{-1}$ or

Table 8.4: Estimated position, velocity and attitude of the sensor platform during the stationary periods at the first marked location in the dataset shown in figure 8.10

Pose number	1	195	400	875
North position (m)	0	-0.0514	-0.3262	-0.5017
East position (m)	0	0.0552	0.1519	0.2961
Down position (m)	0	0.0451	0.0698	0.2817
Roll ($^\circ$)	0	-0.0401	0.0115	-0.4297
Pitch ($^\circ$)	0	-1.2490	-1.3579	-1.6387
Yaw ($^\circ$)	0	-178.6368	-179.1754	-177.8117
North velocity (ms^{-1})	-0.0016	-0.0009	0.0084	0.0017
East velocity (ms^{-1})	0.0025	-0.0006	0.0039	-0.0051
Down velocity (ms^{-1})	-0.0092	-0.0071	-0.0102	-0.0108

Table 8.5: Estimated position, velocity and attitude of the sensor platform during the stationary periods at the second marked location in the dataset shown in figure 8.10

Pose number	82	280	515
North position (m)	8.6965	8.4849	8.1041
East position (m)	-0.9216	-0.7738	-0.7851
Down position (m)	-0.0957	-0.0478	-0.0304
Roll ($^\circ$)	0.7850	0.7448	0.7105
Pitch ($^\circ$)	0.0344	0.5901	0.6646
Yaw ($^\circ$)	-1.5011	-0.2979	-0.9167
North velocity (ms^{-1})	0.0127	-0.0104	-0.0001
East velocity (ms^{-1})	-0.0105	0.0010	-0.0013
Down velocity (ms^{-1})	-0.0108	-0.0081	-0.0063

less. This is a similar result as seen in the previous test.

Tables 8.5 and 8.6 show similar results for the second and third locations in this dataset. Similar position drift can be seen in these poses on the order of $0.5m$ in the north components. All of the north component drifts are in the southerly direction suggesting that any one particular circuit of the trajectory is more consistent than the final position error as they all drift in the same direction.

Table 8.6: Estimated position, velocity and attitude of the sensor platform during the stationary periods at the third marked location in the dataset shown in figure 8.10

Pose number	140	345	820
North position (m)	4.3828	4.1888	3.8841
East position (m)	-1.9260	-1.9368	-1.6434
Down position (m)	-0.0515	-0.0371	0.1344
Roll ($^{\circ}$)	-1.3063	-1.3694	-1.7590
Pitch ($^{\circ}$)	-0.1089	-0.2406	0.4183
Yaw ($^{\circ}$)	-88.8371	-88.4647	-89.0090
North velocity (ms^{-1})	0.0031	-0.0027	0.0017
East velocity (ms^{-1})	0.0021	0.0001	0.0031
Down velocity (ms^{-1})	-0.0101	-0.0070	-0.0088

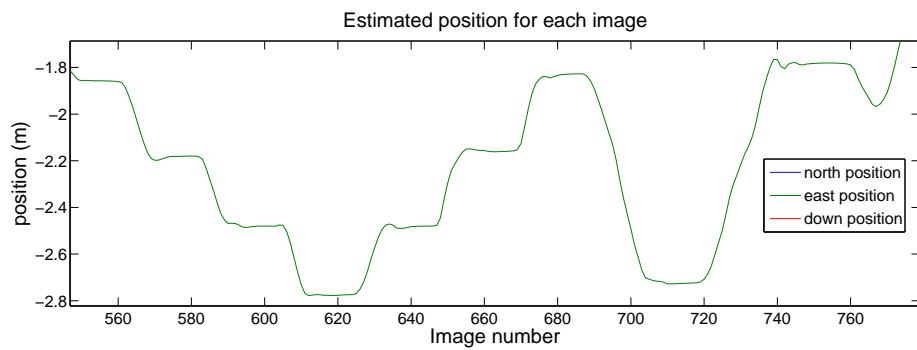


Figure 8.14: Close up showing the east component of the position during the incremental movements between the 30cm intervals from the trajectory shown in figure 8.10.

This supports the idea that the trajectory should be locally consistent even if global consistency is not maintained as identified in section 3.4.

Similar accuracies for the velocity estimates when stationary at the second and third position as they were at the first position can also be seen.

A close up of the east component of the estimated position during the period where the unit was moved between the 30cm intervals is shown in figure 8.14. This section was done to test the local accuracy of the estimate.

Table 8.7 shows the estimated east position component when stationary at

Table 8.7: Estimated east position of the sensor platform during the stationary periods for the 30cm intervals in the dataset shown in figure 8.10. The estimated and true offset from the first location in this section is also shown

Pose number	555	577	600	618	644	664	684	712	757
East position (m)	-1.86	-2.18	-2.48	-2.78	-2.48	-2.16	-1.83	-2.73	-1.78
Estimated offset (m)	0.00	-0.32	-0.62	-0.92	-0.62	-0.30	0.03	-0.87	0.08
True offset (m)	0.00	-0.30	-0.60	-0.90	-0.60	-0.30	0.00	-0.90	0.00

these locations as well as the estimated and true offset from the first pose in this period. The estimated local error is within 3cm except for the last pose.

8.4 First run

After the tests presented, full trajectory datasets were taken walking through and around buildings to simulate the intended application.

The first dataset was taken over the two levels of the Rose St. building starting downstairs, going up the front staircase, along the top floor and back down the back staircase to approximately the starting point. This dataset was taken over a period of 115 seconds and covered a distance of approximately 120 metres.

The estimated trajectory for this dataset is shown in figures 8.16 and 8.17 with an approximate overlay of the trajectory on a satellite image shown in figure 8.15. Unlike the test datasets presented previously, in this example the sensor unit it moving both at the beginning and the end of the trajectory to demonstrate its ability to self initialise from any initial conditions. The initial and final position are approximatly the same however the orientation and velocities are different as the unit was moving in opposite directions at the beginning and end.

The front stair case can be seen as the loop in the lower right corner of figure 8.16 as well as on the left side of figure 8.17. The estimated position of the platform when each image was taken is shown in figure 8.18, the north east and down position are relative to the platform pose in the first image and are not

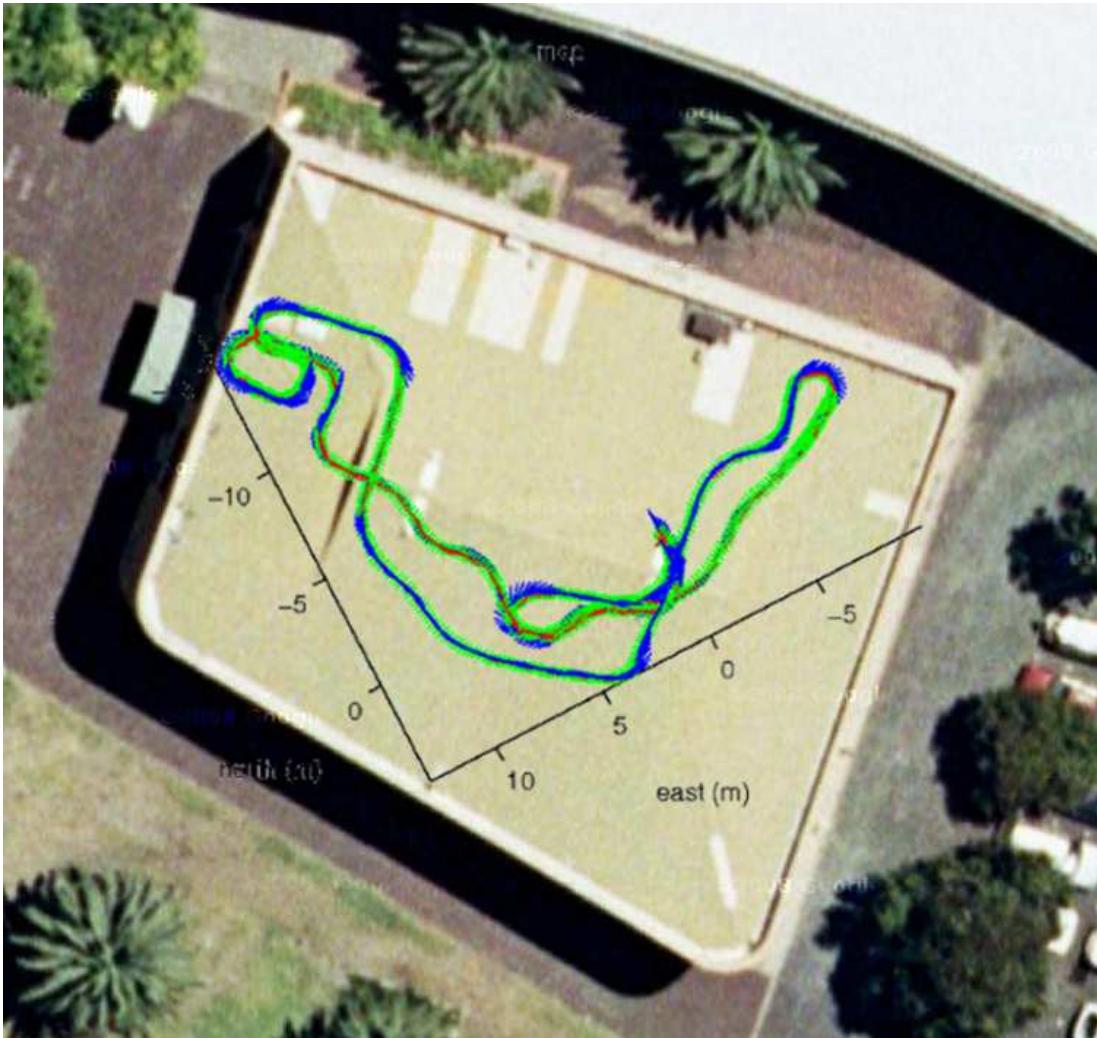


Figure 8.15: Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building overlaid onto a satellite image of the area. The trajectory does not line up exactly partly due to the viewpoint of the image not being taken from directly above.

aligned to a global frame. The velocity and attitude are shown in figures 8.19 and 8.20 respectively. Notice from figure 8.19 how the individual steps taken during the collection of the dataset can clearly be seen in the down velocity estimates.

As with the other tests, no loop closure is performed in these examples and features are only observed as long as a track is maintained in the images and lost features are not required.

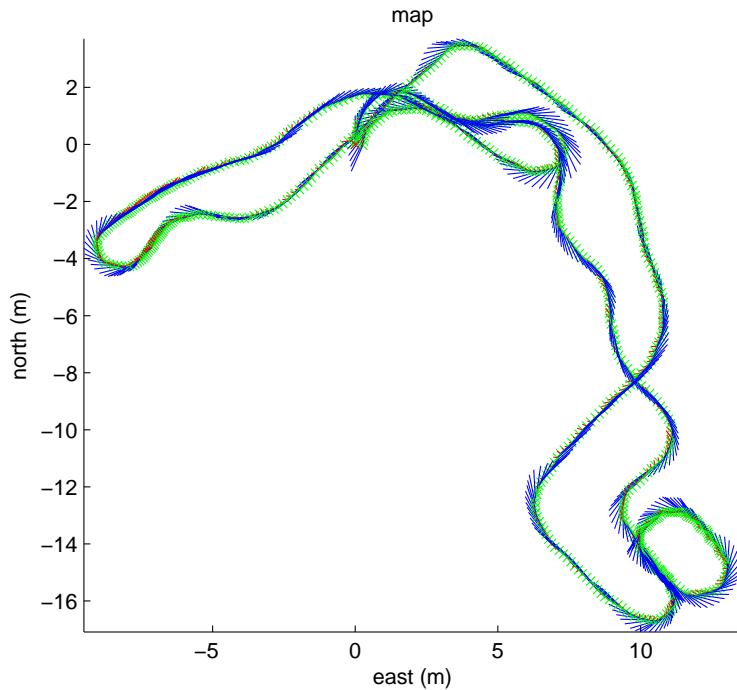


Figure 8.16: Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building. The red cross at $(0,0)$ is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The trajectory consists of 720 image pairs taken over 115 seconds and is approximately 120 metres in length.

The visual observations used in this dataset were extracted Harris corners tracked with Lucas-Kanade optical flow and then a RANSAC implementation to determine the fundamental matrix was used to reject outliers as is described in chapter 7. This is all done before any state estimation is performed. While the filter is running the visual observation edge energies are monitored and this is used as a further means of outlier detection.

Table 8.8 shows the estimated pose states for the first and last pose in this trajectory. These poses are in approximately the same location (to about 10cm or so accuracy) but it can be seen from table 8.8 that the estimated position is off by about 82cm, mostly due to the error in the down direction as can also be seen in figure 8.17. The estimated velocities for both poses are non-zero as

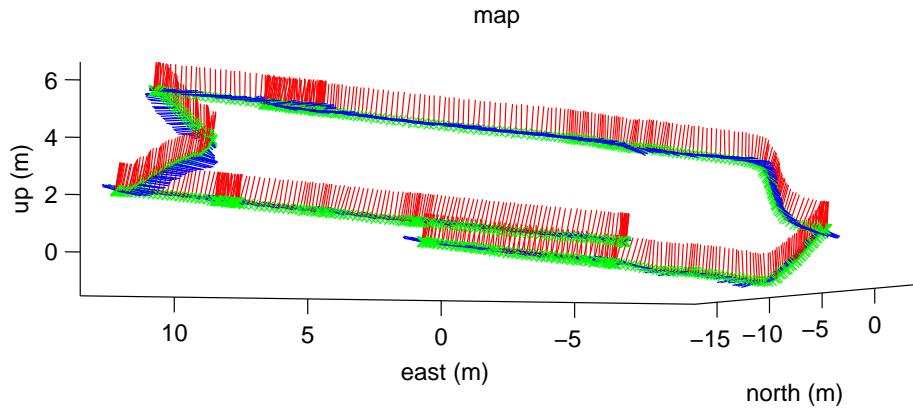


Figure 8.17: Side view of the estimated trajectory of the first dataset showing the two levels of the Rose St. building. The red cross at $(0,0)$ is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The back set of stairs can be clearly seen on the right side of the image as well as the misalignment of the two portions of the trajectory on the bottom level.

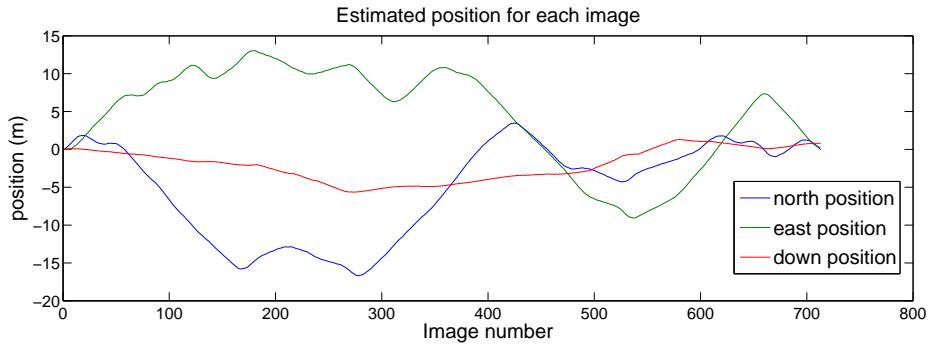


Figure 8.18: Estimated position of the platform for each image in the first Rose St. building dataset. The north, east and down components are in the body frame of the first pose.

this trajectory started and finished while still walking to demonstrate that the platform does not need to be stationary when the filter starts and the initial velocity and attitude do not need to be known.

To compare the drift in attitude over the trajectory the estimated gravity vector for the first and last pose are presented in table 8.9. The gravity vector is reported in the body frame of the first pose in both cases and, due to the forced

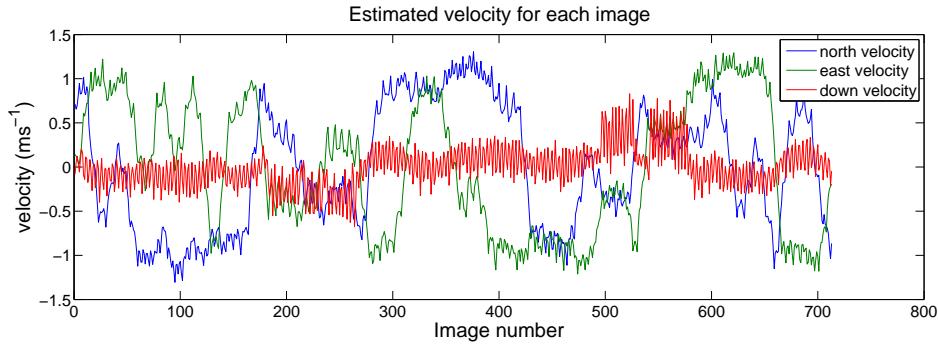


Figure 8.19: Estimated velocity of the platform for each image in the first Rose St. building dataset. The north, east and down components are in the body frame of the first pose. Notice how the down component of velocity clearly shows the individual steps taken even though they can not be seen in the position estimates from figure 8.18.

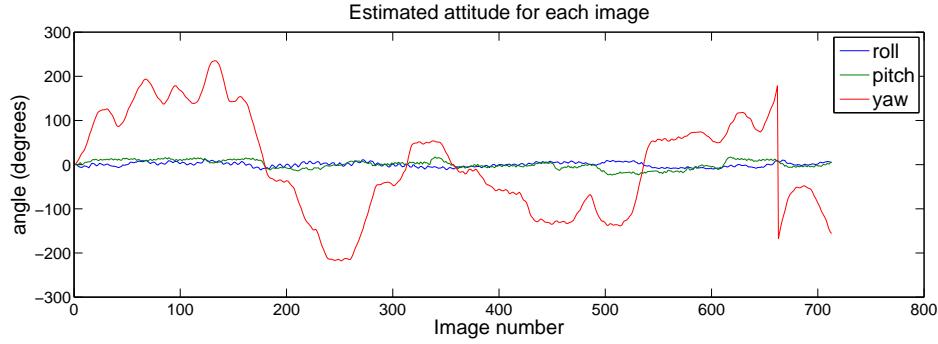


Figure 8.20: Estimated attitude of the platform for each image in the first Rose St. building dataset. The roll, pitch and yaw components are in the body frame of the first pose.

independence filter used, are completely independent as they do not have any common observations in their estimates. The difference in the magnitude of the estimated gravity at these two poses is less than 0.0064% and the angle between them is only 0.45° .

Figure 8.21 shows the result for the same dataset where fundamental matrix outlier rejection is performed but the observation edge energy test is not. There are obvious errors in this estimate as can be seen by the jumping around of the estimated position of the platform in the front stair case shown in the bottom right of the figure. Even though this result is wrong it is important for a number

Table 8.8: First and last pose estimates for the trajectory in the first Rose St. building dataset

Pose number	1	713
North position (m)	0	-0.02909
East position (m)	0	0.26414
Down position (m)	0	0.78309
Roll (o)	0	5.57339
Pitch (o)	0	5.80399
Yaw (o)	0	-156.105
North velocity (ms^{-1})	0.70932	-0.87437
East velocity (ms^{-1})	0.14298	-0.19654
Down velocity (ms^{-1})	-0.08262	-0.05031

Table 8.9: Estimated gravity vector and magnitude for the first and last pose in the first Rose St. building dataset

Pose number	1	713
North component (ms^{-2})	-0.638351	-0.682180
East component (ms^{-2})	0.871024	0.934738
Down component (ms^{-2})	9.742408	9.734159
Magnitude(ms^{-2})	9.802075	9.802701

of reasons.

Firstly it shows how the inertial observations can be used to help with visual observation outlier rejection. Figure 8.22 shows the estimated instantaneous velocity for the result in figure 8.21. It can be seen that there are large spikes in the estimated velocity around images 176 and 236. These correspond to the poses in the staircase with the incorrect visual data association. This is how the inertial observations identify these features as incorrect as these velocity spikes were not observed by the IMU.

Figure 8.23 shows a plot of the number of features tracked between frames and gives a clue as to how these outliers were missed by the fundamental matrix test. The green line in figure 8.23 shows the number of feature matches in each

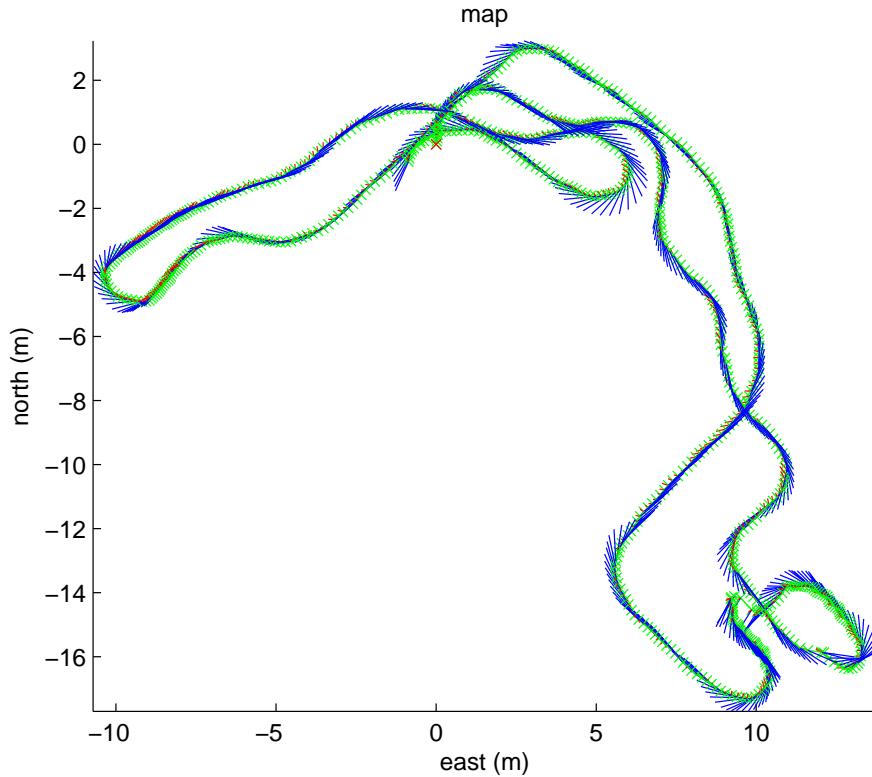


Figure 8.21: Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building without using edge energy outlier detection. The red cross at $(0,0)$ is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. Notice the errors in the estimated pose within the front staircase in the bottom right of the figure.

frame that survived both the fundamental matrix and then edge energy tests, these are considered to be the correctly associated observations. The red line that can be seen for some of the images is the number of observations before the edge energy test has been performed and indicates where outliers have not been caught by the fundamental matrix test alone.

The horizontal blue line in figure 8.23 is at the 8 features per image mark, this is a critical number as it shows the number of images below which a fundamental matrix test is unable to detect outliers in the visual observations. As a fundamental matrix has 7 degrees of freedom, a valid fundamental matrix can be

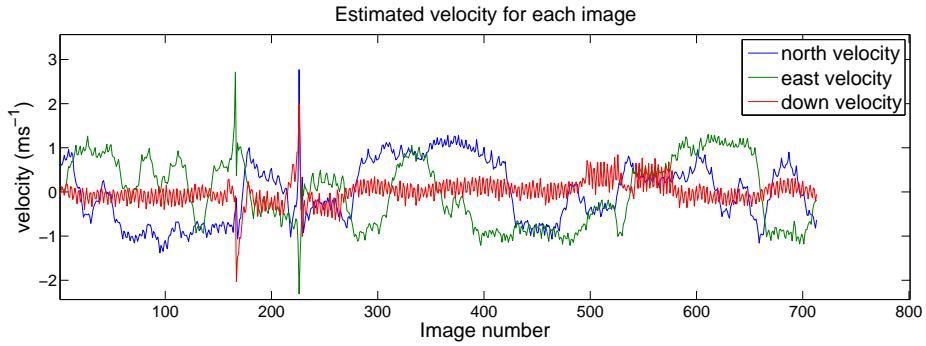


Figure 8.22: Estimated velocity of the platform for each image in the first Rose St. building dataset. The north, east and down components are in the body frame of the first pose. Notice the spikes in estimated velocity around the 176th and 236th pose caused by incorrect data association within the front staircase.

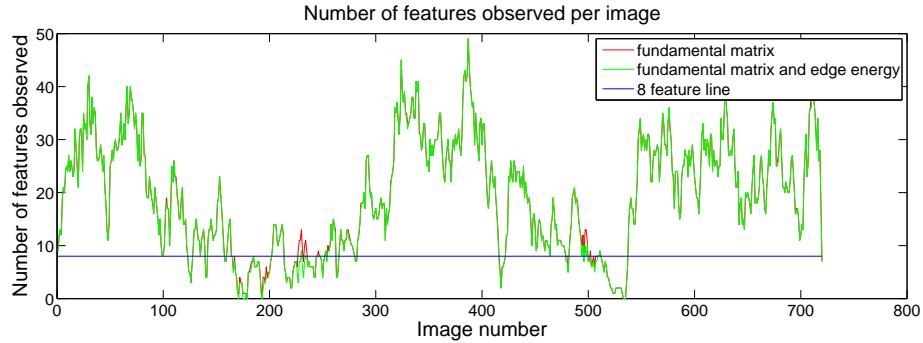


Figure 8.23: Number of image features observed for each image in the first Rose St. building dataset. The red line is after the fundamental matrix constraint has been applied to remove outliers and the green line is after observation edge energies have been used as well. The blue line is at 8 features per image which is the minimum number of correctly associated features required for the fundamental matrix technique to work. Notice how the number of observed features drops around the 200th and 550th image corresponding to the front and back staircases respectively where there are very few visual features to observe. This is where fundamental matrix pruning of outliers performs worst.

formed for any selection of 7 random points. If an 8th point is added, which is an outlier, a fundamental matrix formed by RANSAC still can not identify which point is the outlier as no matter which 7 are used to calculate the matrix, those 7 will be counted as inliers and the remaining point as the outlier. Therefore at least 8 inliers must exist for the RANSAC fundamental matrix test to work, even

if the 7 point algorithm is used.

It can be seen from figure 8.23 that the majority of images with outliers detected from the edge energy test occur where the number of available visual features is near or below this line.

Another important point to note is that for some of these images, the number of available inliers is less than 3, so even if outliers did not exist the filter would not be able to constrain the location of the camera from visual observations alone. This shows that for this dataset, the inertial observations are not only important for rejecting the outliers in the visual features, but also constraining the estimated pose of the platform during periods when insufficient visual observations are available.

Figure 8.24 shows a sample image from the front staircase and it can easily be seen why so few features were tracked in this section of the dataset. The pure white wall of the staircase takes up the majority of the image with no available features to extract. The corners of the stairs which are the only points that can be extracted from the image all look identical and therefore are very difficult to reliably associate from frame to frame.

A third important observation that can be made from the result in figure 8.21 is that even though the filter estimate diverges in the front staircase, it quickly recovers and relative estimates of the trajectory after this point are not affected by this error. This can be seen from the comparison of the later part of the trajectory between figures 8.16 and 8.21. This recovery is automatic and did not require intervention by the user.

The reason why the filter could recover was due to the ability of the pre-integrated inertial observations to be used without initial conditions and the use of forced independence in the information smoother. Because the initial conditions for the inertial observations can be recovered in a linear way, as was described in section 4.2, incorrect initial conditions can be recovered from in a linear way by the filter and the forced independence means once the filter window has moved past the poses with this association error, the error no longer affects the solution.



Figure 8.24: Sample image inside the front staircase from the first dataset. There are very few available visual features from this image except for corners of the stairs which are indistinguishable from each other and therefore can not be reliably tracked from frame to frame.

8.5 Second run

A second long dataset was taken starting in the same location as the first dataset on the bottom floor of the Rose St. building. The trajectory goes up the back staircase of the Rose St. building, across the bridge between the Rose St. building and the Link building, along the third floor of the Link building and down two stories of the back staircase of the Link building, then along the ground outside through the car park and into the back door of the Rose St. building back to the starting point. This dataset was taken over 180 seconds and covered approximately 180 metres.

The estimated trajectory for this dataset is shown in figures 8.26 and 8.27 with

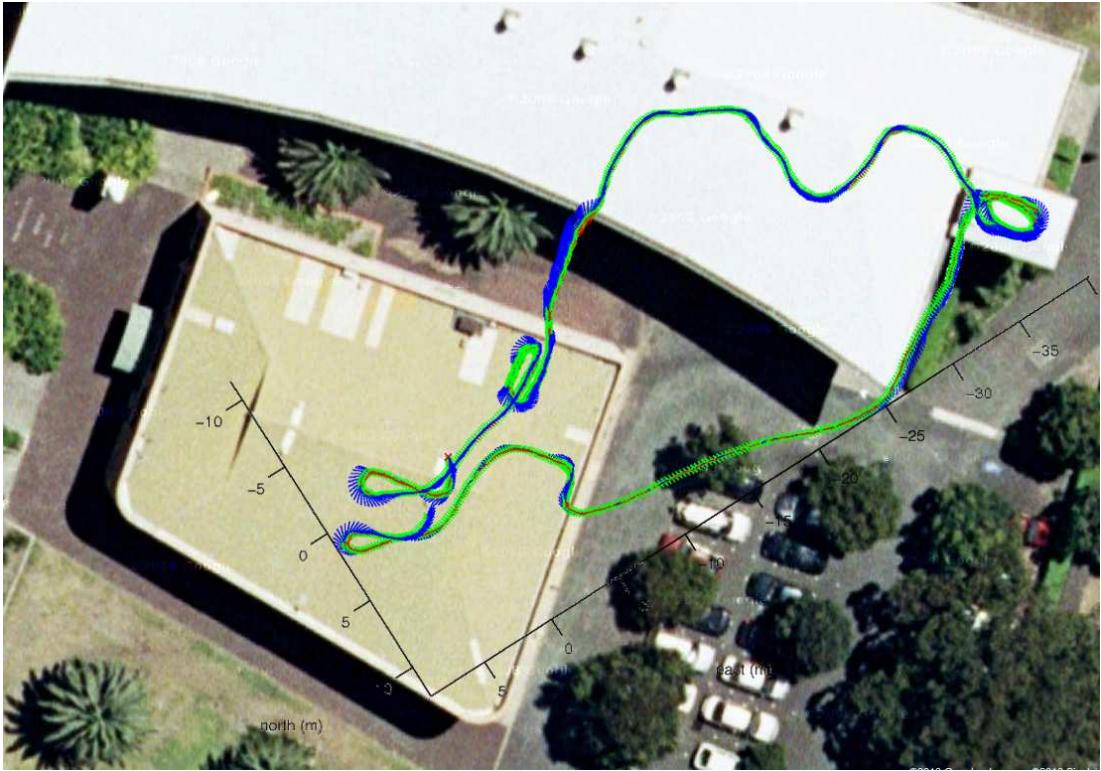


Figure 8.25: Top view of the estimated trajectory of the second dataset between the Rose St. building and the Link building overlaid onto a satellite image of the area. The trajectory does not line up exactly partly due to the viewpoint of the image not being taken from directly above and the drift in the solution towards the end of the trajectory. The bridge between the two buildings is not present in this photo.

an overlay on a satellite image shown in figure 8.25 for comparison.

The estimated trajectory for this run gives an idea of the accuracy of the implementation. Table 8.10 shows the estimated states for the first and last pose in this trajectory which were in approximately the same place. As with the previous dataset, the platform is moving both at the beginning and the end of the trajectory as can be seen from the velocity estimates. The final estimated position error is approximately 2.48 metres. This error is mainly in the north and east components, but the down component estimate is almost within the accuracy of the placement of the unit at the end. This drift is expected as no loop closure is performed.

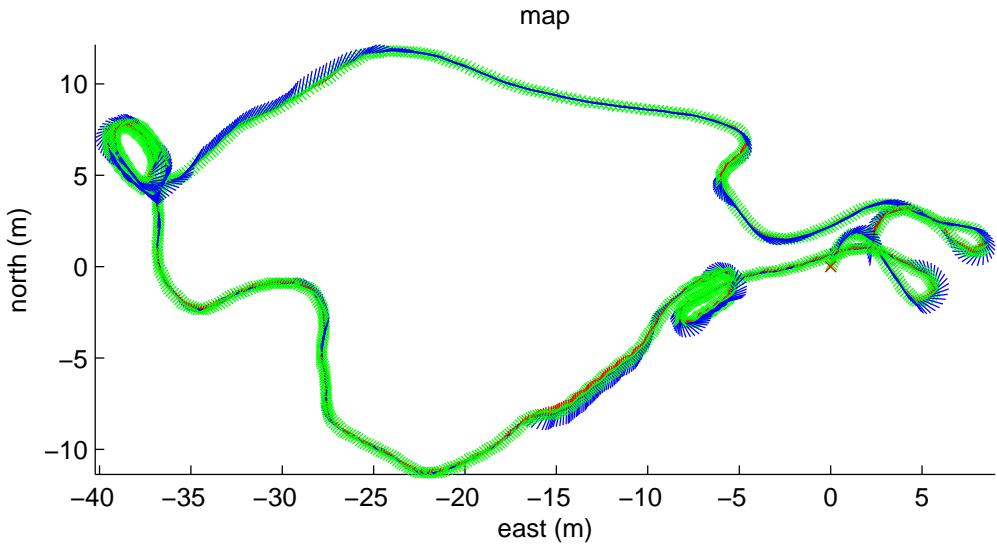


Figure 8.26: Top view of the estimated trajectory of the second dataset between the Rose St. building and the Link building. The red cross at $(0,0)$ is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The trajectory consists of 1130 image pairs taken over 180 seconds and is approximately 180 metres in length.

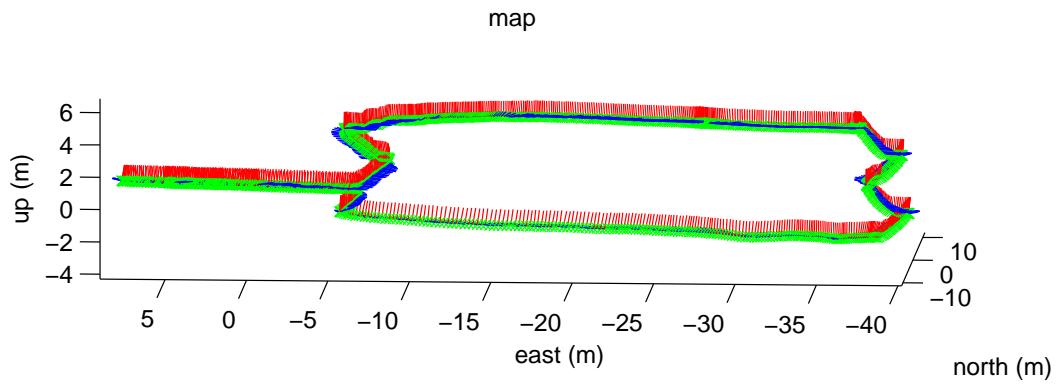


Figure 8.27: Side view of the estimated trajectory of the second dataset showing the two levels of the Rose St. building and the staircase down the back of the Link building. The red cross at $(0,0)$ is the starting point. The red lines show the up direction at each image pose and the blue lines are the forward direction at these poses. The Link building stair case can be clearly seen on the right side of the figure.

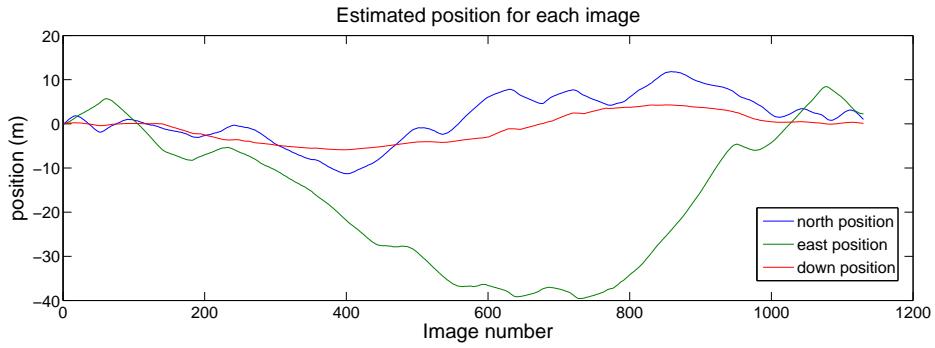


Figure 8.28: Estimated position of the platform for each image in the second dataset. The north, east and down components are in the body frame of the first pose.

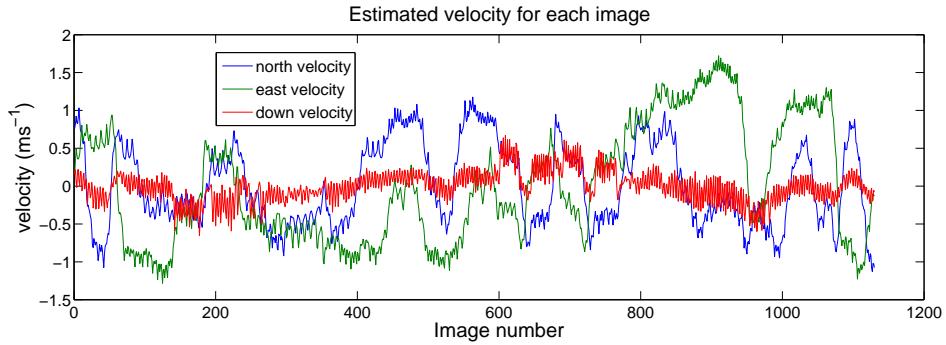


Figure 8.29: Estimated velocity of the platform for each image in the second dataset. The north, east and down components are in the body frame of the first pose. Notice how the down component of velocity clearly shows the individual steps taken even though they can not be seen in the position estimates from figure 8.28.

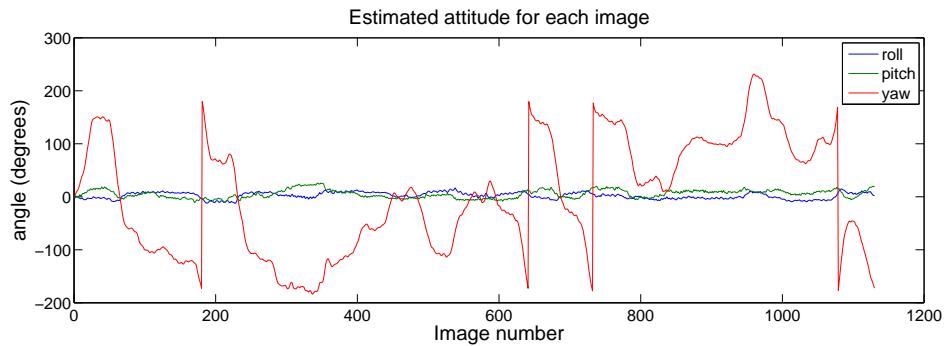


Figure 8.30: Estimated attitude of the platform for each image in the second dataset. The roll, pitch and yaw components are in the body frame of the first pose.

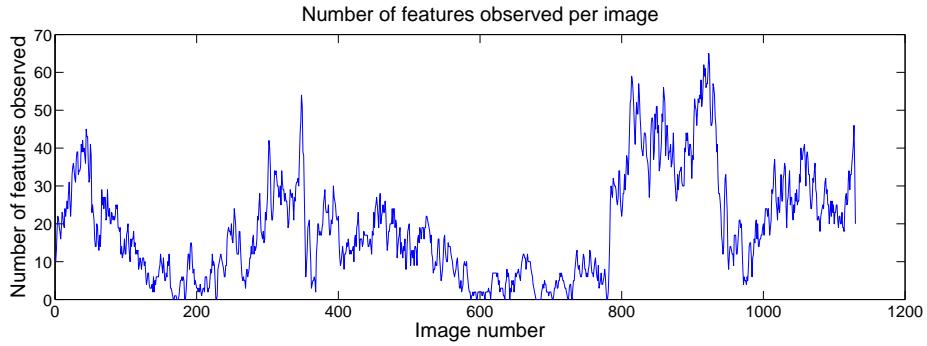


Figure 8.31: Number of image features observed for each image in the second dataset.

Table 8.10: First and last pose estimates for the trajectory in the second dataset through the Rose St. building and the Link building

Pose number	1	1130
North position (m)	0	0.99440
East position (m)	0	2.27144
Down position (m)	0	0.11145
Roll ($^\circ$)	0	1.99135
Pitch ($^\circ$)	0	20.5475
Yaw ($^\circ$)	0	-172.399
North velocity (ms^{-1})	0.76594	-1.07630
East velocity (ms^{-1})	0.15865	-0.11597
Down velocity (ms^{-1})	0.15966	-0.07324

Table 8.11 shows a comparison of the gravity vector estimates for the first and last pose in this trajectory. As with the previous dataset these estimates are completely independent as well. It can be seen that the difference in the magnitude of the gravity vector estimates is within 0.023% and the angle between them is only 0.76° .

An interesting portion of this dataset is in the staircase of the link building when the unit is taken down two levels in a row. It can be seen from figure 8.31 that during this period (approximately from pose 600 to 800) that the average number of observations made per frame is relatively low, as has also been seen in the front staircases in the previous trajectory in section 8.4. This relatively long

Table 8.11: Estimated gravity vector and magnitude for the first and last pose in the second dataset through the Rose St. building and the Link building

Pose number	1	1130
North component (ms^{-2})	-1.393374	-1.487129
East component (ms^{-2})	0.201320	0.289583
Down component (ms^{-2})	9.699877	9.685959
Magnitude(ms^{-2})	9.801512	9.803735

period of time (32 seconds) without sufficient visual observations to constrain the pose of the unit alone shows the benefit of using an IMU for this kind of application. It also demonstrates the performance of the filter during prolonged periods with few observations.

8.6 Monocular scale estimation

In order to test scale estimation using a single camera and inertial observations the dataset used in section 8.4 was processed using inertial observations and images from the left camera only. The resultant estimated trajectory for the monocular case is shown in figure 8.32. The stereo result previously shown in figure 8.16 will be used for scale comparison.

As no prior initial conditions are known the filter in this example was initialised by using the fundamental matrix calculated from the first two images using gyro observations. This is the method explained in section 7.3.2 and derived in appendix B.

A weak initial distance prior between the first two poses of 1 metre with a standard deviation of 1 metre was used to stabilise the estimate until sufficient observations were made to estimate the scale. This prior was removed after the first 20 poses were added into the filter.

After the first few poses once the estimate of the initial conditions of the platform had stabilised, new poses were instead initialised by propagating the inertial

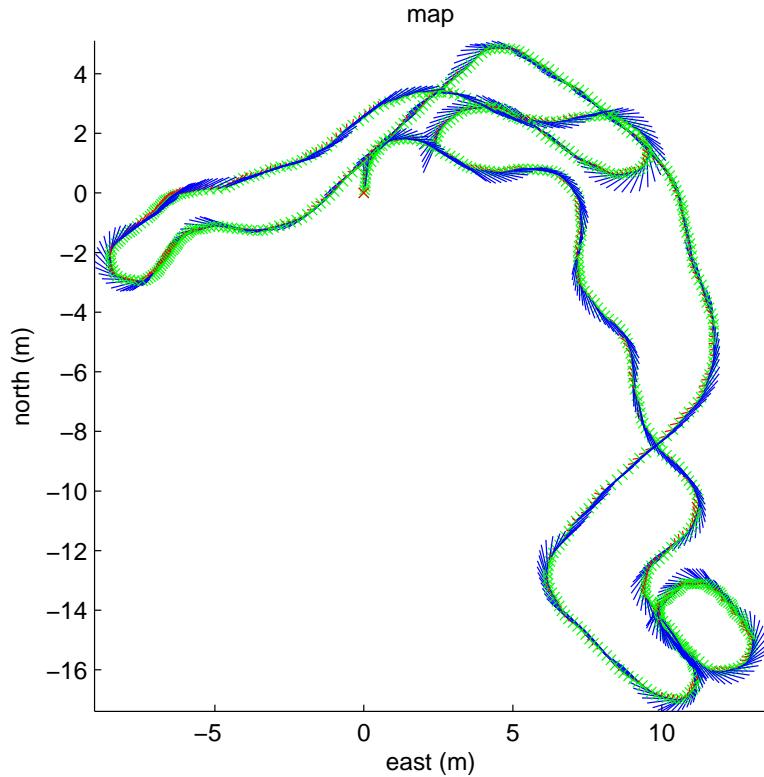


Figure 8.32: Top view of the estimated trajectory of the first dataset over the two levels of the Rose St. building processed with only inertial and monocular observations to assess scale observability. This is the same dataset as used in section 8.4 and figure 8.16 for comparison.

prediction model using the last estimated pose and the inertial observations as explained in section 7.3.1. This method was chosen as it implicitly incorporates the estimated scale information and gives an estimate of the distance between the current and new pose which is not available from the fundamental matrix calculation alone. This is hoped to provide a more accurate starting point for the graphical solver and therefore lead to faster convergence of the solution.

New features were initialised after their second observation using the pose mean estimates of the two poses they were first observed from and triangulation from the visual observations of the feature.

The same 30 image sliding forced independence window was used in this ex-

Table 8.12: First and last pose estimates for the trajectory in the first Rose St. building dataset using monocular observations

Pose number	1	713
North position (m)	0	1.58994
East position (m)	0	2.39830
Down position (m)	0	0.72042
Roll (o)	0	5.57606
Pitch (o)	0	5.77881
Yaw (o)	0	-156.070
North velocity (ms^{-1})	0.73614	-0.87982
East velocity (ms^{-1})	0.15458	-0.20162
Down velocity (ms^{-1})	-0.08452	-0.04595

periment as in the previous results in this chapter. Due to this sliding window the scale estimate at any pose is derived purely from observations made in the local area. Therefore scale estimates made for portions of the trajectory more than 30 poses apart can be considered to be independent.

Table 8.12 shows the estimated pose states for the first and last pose in this trajectory. This can be compared to the results in table 8.8 for the stereo case. It can be seen from this table that the estimated final position has a larger error in the monocular case of 2.97m compared to just 0.82m for the stereo case.

This is expected due to the lower number of observations used and the need to maintain an accurate estimate of scale in the monocular case. This error can be seen mostly in the north and east components with the down component being almost the same. The final attitude estimates for both cases are very similar which is also as expected since the majority of the attitude accuracy comes from the gyro observations and the attitude estimate is independent from the scale problem.

Table 8.13 shows the gravity vector and magnitude estimates for the first and last pose in this trajectory for the monocular case which can be compared to the stereo results in table 8.9. The gravity vector estimate is not as accurate in this case due to the lower number of visual observations used and the weaker

Table 8.13: Estimated gravity vector and magnitude for the first and last pose in the first Rose St. building dataset using monocular observations

Pose number	1	713
North component (ms^{-2})	-0.651686	-0.682310
East component (ms^{-2})	0.866544	0.937213
Down component (ms^{-2})	9.743319	9.736759
Magnitude(ms^{-2})	9.803461	9.805529

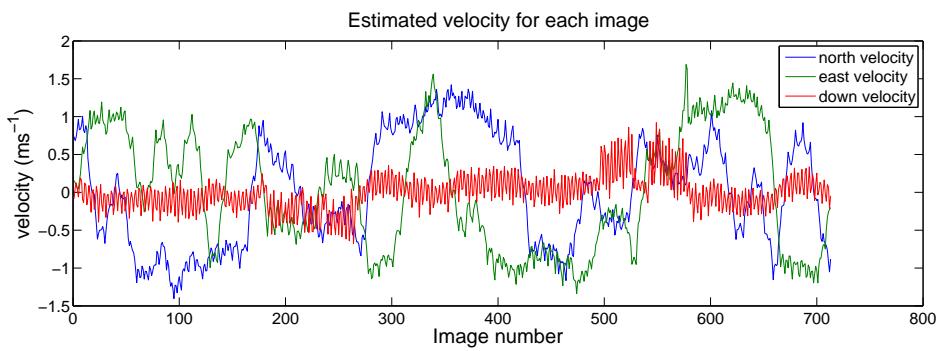


Figure 8.33: Estimated velocity for the monocular result shown in figure 8.32. The estimated velocity for the stereo case on the same dataset can be seen in figure 8.19.

constraint on the scale. However the final gravity vector estimates for the stereo and monocular case are within 0.013° of each other and their magnitudes are within 0.029%.

Figure 8.33 shows the estimated velocity for each pose in this trajectory for the monocular case. When compared to figure 8.19 this gives a good first indication of the performance of the scale estimation as the velocity estimate is directly influenced by the local scale estimate. From these figures it can be seen that the velocity estimates in the two cases roughly track each other with the exception of differences in the east velocity estimate around pose 340 and 575 and in the down component around pose 550. These can be interpreted as inaccuracies in the scale estimate in these regions.

This inaccuracy in the scale estimate near poses 340, 550 and 575 can also be seen in the estimated distance between consecutive poses shown in figure 8.34.

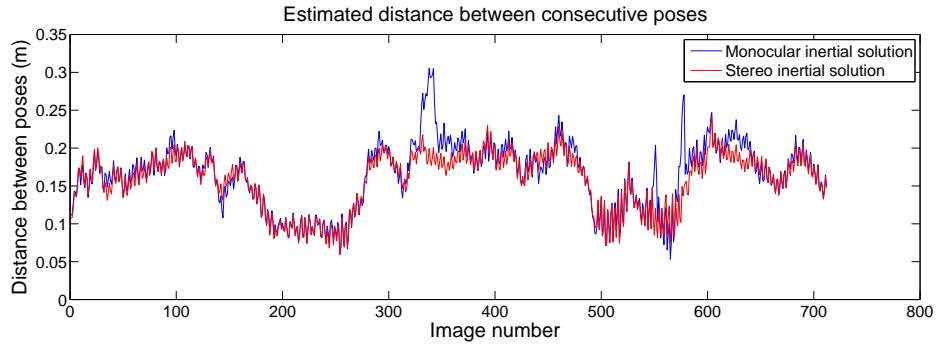


Figure 8.34: Estimated distance between consecutive poses for the monocular and stereo result from the first dataset over the two levels of the Rose St. building.

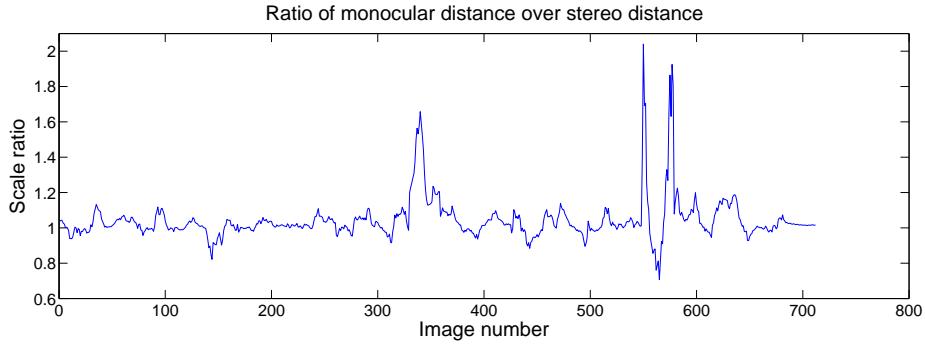


Figure 8.35: Ratio of the estimated distance between consecutive poses for the monocular and stereo result from the first dataset over the two levels of the Rose St. building.

When the ratio of these distances between poses for the two results is taken an estimate of the scale factor over the trajectory for the monocular result can be found. This result is shown in figure 8.35

The mean scale value for this dataset, taken from the ratio of estimated pose distances, is 1.0425 with a variance of 0.0144 including the regions of poor scale observability around poses 340, 550 and 575. This level of scale accuracy was possible using 30 images taken at 6.25Hz which represents just 4.8 seconds worth of observations.

This dataset used a forward looking camera which is not ideal when performing monocular SLAM as very little parallax is experienced for many features. However, the accelerations due to the high dynamic nature of a human mounted

system does add to the scale observability in this example as can be seen in the sinusoidal nature of the down velocity component in figure 8.33.

One drawback of estimating map scale in this way is that because the scale may be only weakly observable, convergence of the solution when a new pose is added may be slow. In this example when a new pose and image observations from that pose was added an average of 20 iterations of a damped indirect gradient descent solve were required, with relinearising of the observations between each iteration, in order to achieve convergence within 1%. In the stereo case an average of just 2 iterations were required.

This slow convergence rate is due to the weak observability of the scale causing the solution for the whole map to oscillate as it converges if the predicted new pose and landmark features are not close to the final solution. It may be possible to reduce this problem if a different parameterisation or solving method for the monocular scale estimation is used.

8.7 Chapter summary

This chapter presented and analysed the results of a real-world implementation using pre-integrated inertial delta observations with visual SLAM.

The sensor suite used and the implemented form of the SLAM filter was presented in section 8.1. This was followed by a description of the environment in which the datasets were taken.

Two test trajectories were analysed first in sections 8.2 and 8.3 where the sensor unit was placed on known marked locations a number of times throughout the trajectories. This was done to assess the accuracy of the solutions generated and to ensure the system was performing as expected.

These tests confirmed the local metric accuracy of the filter, in particular the accuracy of the estimated velocity. Gradual dead reckoning drift was also observed which was expected as loop closure was not performed.

After the two initial tests, two longer datasets that more closely resemble

the kinds of trajectories the system would experience in applications of interest were collected. Analysis of these datasets highlighted the importance of using edge energies for identifying mis-associated visual features as suggested in section 7.2.4, as well as the robustness of using forced independence in the filter to help with automatic recovery from errors such as mis-association.

The accuracy of the implementation over longer trajectories was also seen, especially during periods where few visual observations are available. This confirmed the benefit of using an IMU to aid visual SLAM as during these periods the location of the camera could not be constrained from visual observations alone.

Finally, one of the longer datasets was used in a monocular inertial SLAM implementation with no initial conditions to assess the performance of scale estimation on real data. Comparisons of the produced map scale were made to the stereo inertial result for the same dataset to prove the accuracy of the estimated scale.

Chapter 9

Conclusions

9.1 Summary of contributions

This thesis has developed and analysed a new technique for processing and incorporating inertial observations when used with other body frame sensors such as cameras. Practical ways to implement a SLAM system using this new technique were also explained and results presented.

This chapter summarises the contributions from the work in this thesis. Possible future directions for research are also discussed.

9.1.1 Development of pre-integrated inertial delta observations

Chapter 4 presented the development of the pre-integrated inertial delta observation as a new way to pre-process and use inertial observations. This method allows the inertial observations to be integrated before initial conditions of the platform are applied.

The result not only allows inertial observations to be used before the velocity and attitude of the platform are known, removing any initialisation requirements for the inertial navigation system, but it also allows these initial conditions to be recovered quickly in a linear way. This is due to the choice of a body frame centred

navigation coordinate frame that greatly reduces, if not completely removes, many of the sources of non-linearity in the inertial integration equations.

Since a number of inertial observations are pre-integrated into a single inertial delta observation, this method also reduces the number of prediction steps required in a filter and the number of inertial poses that need to be retained in a delayed state smoothing implementation.

The role of the new Δp^+ component of the delta observations was explained, specifically its ability to link the average velocity experienced by visual (or similar body frame sensor) observations to the instantaneous velocity experienced by the inertial observations.

9.1.2 Analysis of pre-integrated inertial delta observations

Chapter 5 investigated the equations of the pre-integrated inertial delta observations developed in chapter 4 to analyse how this method improves linearity and removes initial condition requirements for inertial navigation.

Monte-Carlo simulations confirmed that pre-integrated inertial observations perform as expected for the estimation of the initial conditions of the platform. The validity of correcting for inertial sensor biases after pre-integration instead of during was also tested in simulation and confirmed to be accurate.

A comparison of results achieved with pre-integrated inertial delta observations to results using standard inertial integration techniques was also made.

9.1.3 Visual map scale estimation using inertial observations

Chapter 6 showed how accelerometer observations can be used to make the scale of a bearing only map observable. The equation behind scale estimation in these situations was derived along with the estimate of its uncertainty.

The extension of this scale equation over multiple poses and to 6 DoF systems was also discussed. Results of Monte Carlo simulations of an example implemen-

tation demonstrating the scaling of the map from using inertial observations was also presented.

9.1.4 Inclusion of gyro information in fundamental matrix calculations

In section 7.2.3 the use of gyro observations to reduce the number of image matches required to estimate the fundamental matrix was introduced. The derivation of how to use this information to recover the fundamental matrix linearly using just three matched features is shown in appendix B.

This method has advantages for outlier rejection when only a small number of tracked features are present as well as giving more accurate estimated camera positions by resolving the possible ambiguity between rotation and translation that can occur when fundamental matrices are estimated from narrow field of view cameras.

9.1.5 Sliding window forced independence

Section 7.4.3 presented another choice for reducing the computational load when solving graphical implementations of SLAM besides the traditional marginalisation and conditioning of the graph.

The forced independence method isolates the currently considered poses and features by ignoring all past estimates and observations. This method is possible with using pre-integrated inertial delta observations as the initial conditions for the navigation solution can be estimated linearly from the observations within the current section of the map alone.

This technique is conservative but has the advantage of being able to isolate errors in linearisation and data association to local areas of the map allowing the filter to automatically recover from errors as demonstrated in section 8.4.

9.2 Future work

There are a number of areas identified for future work in this thesis, these are discussed further in this section.

9.2.1 Yaw estimation using pre-integrated inertial delta observations

Body frame sensor observations, such as from cameras or laser scanners, alone do not provide the ability to fix the local first pose based reference frame to any form of absolute reference frame. More specifically the absolute yaw of the local frame in a globally fixed frame can not be determined.

The lack of observability of absolute yaw is actually an advantage when used with pre-integrated inertial delta observations when no initial condition estimates are available. This is because it allows the initial attitude to be fixed to an arbitrary value removing the major source of non-linearity from the inertial integration equations.

As the gravity vector is estimated, in a linear way, as part of the inertial states this can be used to derive the initial roll and pitch of the platform relative to the global inertial frame, however an absolute yaw estimate can not be observed from body frame relative and inertial observations alone.

This is not a problem when body frame sensors are the only other sensors used to aid the inertial estimates but when a sensor that is sensitive to absolute yaw estimates, such as GPS or a magnetometer, is used this becomes a problem as absolute yaw can not be recovered in a linear way.

The problem is that for body frame velocity observations, such as those provided from visual observations, a certain, platform consistent, parameterisation must be used to deal with large heading uncertainties. However, for navigation frame velocity observations, such as those provided from GPS, another, geographic consistent, parameterisation must be used [51].

One way of dealing with this problem would be to over parameterise the

yaw estimate into $\cos(\text{yaw})$ and $\sin(\text{yaw})$ in a way similar to that used in [51] or [36]. These methods claim to be linear as the rotation equations are linear combinations of the $\cos(\text{yaw})$ and $\sin(\text{yaw})$ terms, and they do go a long way to making the observation equations more linear but the non-linear constraint of equation 9.1 must still be maintained.

$$\sqrt{\cos(\text{yaw})^2 + \sin(\text{yaw})^2} = 1 \quad (9.1)$$

A similar non-linear constraint must be used with the quaternion based methods as well.

Another more complicated but possibly more useful method would be to separate the yaw independent information in the observation from the yaw dependent information. This way the yaw independent information, such as the observed distance between two poses, can be fused in the normal filter, and another higher level filter can then be implemented with the yaw dependent information to estimate the alignment of the yaw of the local frame to the global frame.

9.2.2 Loop closure for forced independence smoothing

When sliding window forced independence is used and loop closure is detected, the way to close the loop is not straight forward and may be dependent on the application. Loop closure has not been discussed in this thesis as the focus has been on initialisation of the inertial navigation system.

Since a probabilistic representation of the whole trajectory is not maintained, it is not possible to propagate the effects of the loop closure back through the entire trajectory in a simple way.

If only local metric navigation is required, and global navigation can be achieved by topological means, then correcting the trajectory as a result of the loop closure may not even be necessary. A simple possibility could be to just move the body coordinate used for the navigation frame to the previous pose location where the loop closure was detected. This would align the recent poses

that are currently being estimated with the previous part of the trajectory and navigation can continue as normal.

This method will cause a discontinuity between the current poses and the poses that have recently been marginalised out of the sliding window, but if the vehicle returns to this part of the map the process can be repeated in reverse to realign the maps again.

Other possible methods could include forming something similar to star nodes [23] with groups of previous poses that have left the sliding window and attempt to adjust the global coordinates of these star nodes once loop closure is detected. This may or may not be necessary depending on the desired use of the application and would most probably only be necessary if it is required for display to the user as a single continuous map.

9.2.3 More stable monocular scale estimation

In section 8.6 the slow convergence rates experienced when using implicit scale estimation was discussed. This was due to the oscillation experienced in the map scale when new observations are added when scale observability is weak.

Even with poor scale observability it was shown in section 8.6 that accurate scale estimates can be achieved, the only problem was in the speed of convergence. It would be useful if a different practical method could be found for scale estimation that avoids this oscillation in the estimate.

This method could simply involve a modified graph relaxation technique, such as conditioning on older poses to fix the scale of the map until the new pose and observed landmark estimates have converged. After this step the conditioning on the older poses can be removed to obtain an unconstrained solution which hopefully converges without oscillation to achieve the same result as the method used in the result in section 8.6 but in fewer iterations.

Another possibility could be to find an alternative parameterisation, perhaps somewhere between the explicit and implicit parameterisations discussed in section 6.6 that results in a more linear estimation process that avoids this oscillation.

Bibliography

- [1] Motilal Agrawal, Kurt Konolige, and Morten Rufus Blas. Censure: Center surround extrema for realtime feature detection and matching. In *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 102–115. Springer, 2008.
- [2] Jamshaid Ali and Fang Jiancheng. Alignment of strapdown inertial navigation system: A literature survey spanned over the last 14 years. In *School of Instrumentation Science and Optoelectronics Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083, China*.
- [3] T. Bailey. Constrained initialisation for bearing-only slam. In *IEEE International Conference on Robotics and Automation*, 2003.
- [4] Tim Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. Ph. D. thesis, The University of Sydney, Sydney, Australia, 2002.
- [5] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.
- [6] D.O. Benson. A comparison of two approaches to pure-inertial and doppler-inertial error analysis. In *IEEE Transactions on Aerospace and Electronic Systems*, volume 11, pages 447–455, 1975.

- [7] Michael Bosse, Paul Newman, John Leonard, Martin Soika, Wendelin Feiten, and Seth Teller. An atlas framework for scalable mapping. In *IEEE International Conference on Robotics and Automation*, pages 1899–1906, 2003.
- [8] M. Bryson and S. Sukkarieh. Bearing-only SLAM for an airborne vehicle. In *ACRA*, Sydney, Dec 2005.
- [9] Mitch Bryson and Salah Sukkarieh. Building a Robust Implementation of Bearing-only Inertial SLAM for a UAV. *Journal of Field Robotics*, 24(1-2):113–143, 2007.
- [10] J. A. Castellanos, J. Neira, and J. D. Tardos. Limits to the consistency of the EKF-based SLAM. In *Intelligent Autonomous Vehicles (IAV-2004)*, 2004.
- [11] Laura A. Clemente, Andrew J. Davison, Ian D. Reid, Jose Neira, and Juan D. Tardos. Mapping large loops with a single hand-held camera. In *Robotics: Science and Systems*. The MIT Press, 2007.
- [12] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
- [13] K. Daniilidis and H. Nagel. The coupling of rotation and translation in motion estimation of planar surfaces. In *IEEE Computer Vision and Pattern Recognition*, pages 188–193, 1993.
- [14] A. Davison. Real-time simultaneous localisation and mapping with a single camera. In *International Conference on Computer Vision*, Nice, October 2003.
- [15] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *IJRR*, 25(12):1181–1203, 2006.

- [16] M.W.M Gamini Dissanayake, Paul Newman, Steve Clark, H.F. Durrant-Whyte, and M. Corba. A solution to the simultaneous localization and map building SLAM problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.
- [17] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localisation and mapping (slam): Part I the essential algorithms. *IEEE Robotics and Automation Magazine*, 2:2006, 2006.
- [18] E. Eade and T. Drummond. Monocular slam as a graph of coalesced observations. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, December 2007.
- [19] R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22, 2006.
- [20] Ryan Eustice and Hanumant Singh. Visually navigating the RMS Titanic with SLAM information filters. In *Proceedings of Robotics: Science and Systems*, pages 57–64, 2005.
- [21] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [22] John Folkesson and Henrik I. Christensen. Robust SLAM. In *IFAC Symposium on Intelligent Autonomous Vehicles (IAV2004)*, 2004.
- [23] John Folkesson and Henrik I. Christensen. Graphical SLAM for outdoor applications. *Journal of Field Robotics*, 24:51–70, 2007.
- [24] John Folkesson and Henrik I. Christensen. SIFT Based Graphical SLAM on a Packbot. In *FSR*, pages 317–328, 2007.
- [25] Friedrich Fraundorfer, Christopher Engels, and David Nister. Topological mapping, localization and navigation using image collections. In *Proceedings*

- of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3872–3877, October 2007.
- [26] U. Frese. Treemap: An $O(\log n)$ algorithm for indoor simultaneous localisation and mapping. *Autonomous Robots*, 21:103–122, 2006.
 - [27] R. Hartley. In defence of the 8-point algorithm. In *ICCV 95: Proceedings of the Fifth International Conference on Computer Vision*, page 1064, Washington, DC, USA, 1995. IEEE Computer Society.
 - [28] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.
 - [29] Richard I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision*, pages 579–587, London, UK, 1992. Springer-Verlag.
 - [30] A. 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, Sept. 2008.
 - [31] S. Julier and J. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *IEEE International Conference on RObotics and Automation*, 2001.
 - [32] R.E. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 82:35–45, 1960.
 - [33] J.H. Kim and S. Sukkarieh. Airborne simultaneous localization and map building. In *IEEE International Conference on Robotics and Automation*, 2003.
 - [34] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234, June 2008.

- [35] Georg Klein and David Murray. Improving the agility of keyframe-based slam. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.
- [36] Xiaoying Kong. *Inertial Navigation System Algorithms for Low Cost IMU*. Ph. D. thesis, The University of Sydney, Sydney, Australia, 2000.
- [37] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *Robotics, IEEE Transactions on*, 24(5):1066–1077, Oct. 2008.
- [38] Thomas Lemaire, Cyrille Berger, Il-Kyun Jung, and Simon Lacroix. Vision-based slam: Stereo and monocular approaches. *Int. J. Comput. Vision*, 74(3):343–364, 2007.
- [39] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.
- [40] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.
- [41] Todd Lupton and Salah Sukkarieh. Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial. In *ICRA*, pages 3698–3703, 2008.
- [42] J. M. M. Montiel, J. Civera, and A. Davison. Unified inverse depth parametrization for monocular SLAM. In *RSS*, Philadelphia, Pennsylvania, 2006.
- [43] K. P. Murphy, Y. Weiss, and M. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence*, pages 467–475, 1999.

- [44] C. Moller. *The Theory of Relativity*. Delhi: Oxford University Press, 1972.
- [45] Juan Nieto, Tim Bailey, and Eduardo Nebot. Scan-SLAM: Combining EKF-SLAM and scan correlation. In *Field and Service Robotics*, 2005.
- [46] Edwin Olson, John Leonard, and Seth Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *ICRA*, pages 2262–2269, 2006.
- [47] L.M. Paz, P. Pinies, J.D. Tardos, and J. Neira. Large-scale 6-DOF SLAM with stereo-in-hand. *Robotics, IEEE Transactions on*, 24(5):946–957, Oct. 2008.
- [48] P. Pinies, T. Lupton, S. Sukkarieh, and J. D. Tardos. Inertial aiding of inverse depth SLAM using a monocular camera. In *ICRA*, 2007.
- [49] G. Quian, Q. Zheng, and R. Chellappa. Reduction of inherent ambiguities in structure from motion problem using inertial data. In *IEEE International Conference on Image Processing*, 2000.
- [50] Paul Savage. A unified architecture for strapdown integration algorithms. *Journal of Guidance, Control, and Dynamics*, 29(2):237–249, 2006.
- [51] B.M. Scherzinger. Inertial navigator error models for large heading uncertainty. In *Position Location and Navigation Symposium, 1996., IEEE 1996*, pages 477–484, Apr 1996.
- [52] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.
- [53] T. Soler and M. Chin. On transformation of covariance matrices between local cartesian coordinate systems and commutative diagrams. In *45th Annual Meeting ACSM*, 1985.
- [54] Christel-Loic Tisse, Hugh Durrant-Whyte, and R. Andrew Hicks. An optical navigation sensor for micro aerial vehicles. *Computer Vision and Image Understanding*, 105(1):21 – 29, 2007.

- [55] D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. The American Institute of Aeronautics and Astronautics, second edition, 2004.
- [56] Zhan Wang, Shoudong Huang, and Gamini Dissanayake. D-slam: Decoupled localization and mapping for autonomous robots. In *Proceedings of the 12th International Symposium of Robotics Research*, 2005.
- [57] S.B. Williams. *Efficient Solutions to Autonomous Mapping and Navigation Problems*. Ph. D. thesis, The University of Sydney, Sydney, Australia, 2001.
- [58] Yuanxin Wu, Xiaoping Hu, Meiping Wu, and Dewen Hu. Strapdown inertial navigation using dual quaternion algebra : Error analysis. In *IEEE transactions on aerospace and electronic systems*, volume 42, pages 259–266, 2006.

Appendix A

Simulation setup

A simulation was set up to help in the analysis of the performance of inertial pre-integrated delta observations which can be seen in chapter 5. The use of simulated results over real data is handy for analysis as ground truth is available and Monte-Carlo simulations can be performed by running the simulation over the same trajectory many times with different landmark distributions and sensor noises.

A.1 Simulation model

Unless otherwise specified the set-up used in the simulations was modelled on the specifications of the experimental sensor suite presented in chapter 8 with a Honeywell HG1900 IMU and a Point Grey Research Bumblebee2 stereo camera. The IMU provides samples at a rate of 600Hz and images are taken at 6.25Hz. The simulated sensor characteristics used are the same as those measured from actual hardware and can be seen in tables 8.1 and 8.2.

Poses are generated at the IMU sample rate and used to calculate the inertial observations that would be observed for these motions. White noise and a randomly selected biases is then added to each observations taken from a Normal distribution with a mean of 0 and the standard shown in table 8.1. A different random bias is selected for each simulation run, however the same bias is used

throughout the run, to simulate the turn-on to turn-on bias variance of a real IMU and so that Monte-Carlo simulations can be performed.

Landmarks are randomly positioned around the trajectory and the true simulated poses at the camera sampling rate is used to calculate the observations that would be taken by the camera. Normally distributed white noise is then added to the visual observations as well using the sensor characteristics shown in table 8.2.

A.2 Simulated trajectory

The trajectory used for the simulations involves sinusoidal excitation in all three translational axes as well as in roll, pitch and yaw. The equations used for the generation of the trajectory are:

```
pose.north = sin(time/2)
pose.east = (sin(time/2) + cos(time/2))
pose.down = cos(time/2)
pose.roll = sin(time/2)
pose.pitch = cos(time/2)
pose.yaw = sin(time/2)
```

This trajectory can be seen in figure A.1 which shows the orientation of the platform as well as some of the randomly placed visual landmarks.

The sinusoidal trajectory was chosen as it provides full excitation of all the modes of motion in the full 6 degrees of freedom of the platform. It also has a non-zero initial velocity so that the observability and estimation of the initial conditions can be analysed. A gravity vector of $[0\ 0\ 9.81]^T$ was used in the simulations so that gravity vector observability could be analysed as well.

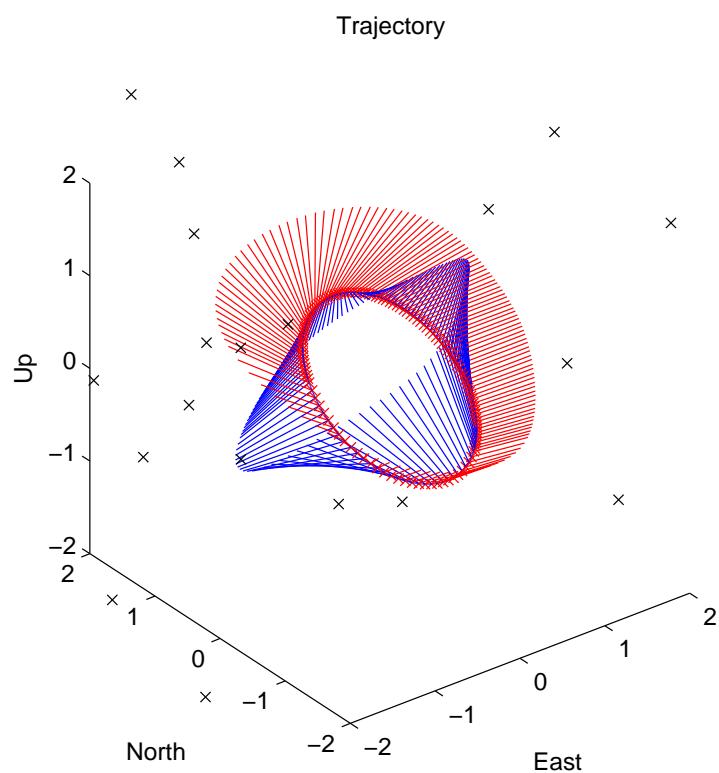


Figure A.1: Trajectory used for simulation results. Red line points in the vehicle frame up direction, blue lines point in the vehicle frame forward direction for each pose that an image is taken at. The black crosses are the randomly distributed visual landmarks.

Appendix B

Fundamental matrix calculation with known rotation

Adapted from the fundamental matrix derivation given in example 9.2 of [28]. Some of the notation has been redefined to remain consistent with the coordinate systems and notation in the rest of this thesis.

Projective matrices for the two camera poses

$$P = K[I|0] \quad (\text{B.1})$$

$$P' = K'[R^{-1}| - R^{-1}t] \quad (\text{B.2})$$

Epipole location in the second image

$$e' = -K'R^{-1}t \quad (\text{B.3})$$

Fundamental matrix

$$F = [e']_\times K'R^{-1}K^{-1} \quad (\text{B.4})$$

Skew symmetric matrix of the epipole

$$[e']_{\times} = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \quad (\text{B.5})$$

Fundamental matrix constraint for a single observed feature tracked over two frames

$$x'^T F x = 0 \quad (\text{B.6})$$

B.1 Three point fundamental matrix algorithm with known rotation

Let

$$x = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (\text{B.7})$$

$$x' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad (\text{B.8})$$

$$A = K'R^{-1}K^{-1} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (\text{B.9})$$

Then

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (\text{B.10})$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (\text{B.11})$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} -e_3 a_{21} + e_2 a_{31} & -e_3 a_{22} + e_2 a_{32} & -e_3 a_{23} + e_2 a_{33} \\ e_3 a_{11} - e_1 a_{31} & e_3 a_{12} - e_1 a_{32} & e_3 a_{13} - e_1 a_{33} \\ -e_2 a_{11} + e_1 a_{21} & -e_2 a_{12} + e_1 a_{22} & -e_2 a_{13} + e_1 a_{23} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad (\text{B.12})$$

$$\begin{bmatrix} x' & y' & 1 \end{bmatrix} \begin{bmatrix} (-e_3 a_{21} + e_2 a_{31})x + (-e_3 a_{22} + e_2 a_{32})y + (-e_3 a_{23} + e_2 a_{33}) \\ (e_3 a_{11} - e_1 a_{31})x + (e_3 a_{12} - e_1 a_{32})y + (e_3 a_{13} - e_1 a_{33}) \\ (-e_2 a_{11} + e_1 a_{21})x + (-e_2 a_{12} + e_1 a_{22})y + (-e_2 a_{13} + e_1 a_{23}) \end{bmatrix} = 0 \quad (\text{B.13})$$

$$\begin{aligned} & (-e_3 a_{21} + e_2 a_{31})xx' + (-e_3 a_{22} + e_2 a_{32})yx' + (-e_3 a_{23} + e_2 a_{33})x' \\ & + (e_3 a_{11} - e_1 a_{31})xy' + (e_3 a_{12} - e_1 a_{32})yy' + (e_3 a_{13} - e_1 a_{33})y' \\ & + (-e_2 a_{11} + e_1 a_{21})x + (-e_2 a_{12} + e_1 a_{22})y + (-e_2 a_{13} + e_1 a_{23}) = 0 \end{aligned} \quad (\text{B.14})$$

$$\begin{aligned} & -e_3 a_{21}xx' + e_2 a_{31}xx' - e_3 a_{22}yx' + e_2 a_{32}yx' - e_3 a_{23}x' + e_2 a_{33}x' \\ & + e_3 a_{11}xy' - e_1 a_{31}xy' + e_3 a_{12}yy' - e_1 a_{32}yy' + e_3 a_{13}y' - e_1 a_{33}y' \\ & - e_2 a_{11}x + e_1 a_{21}x - e_2 a_{12}y + e_1 a_{22}y - e_2 a_{13} + e_1 a_{23} = 0 \end{aligned} \quad (\text{B.15})$$

$$\begin{aligned} & e_1 a_{22}y + e_1 a_{23} + e_1 a_{21}x - e_1 a_{33}y' - e_1 a_{32}yy' - e_1 a_{31}xy' \\ & - e_2 a_{13} - e_2 a_{11}x - e_2 a_{12}y + e_2 a_{33}x' + e_2 a_{32}yx' + e_2 a_{31}xx' \\ & - e_3 a_{21}xx' - e_3 a_{22}yx' - e_3 a_{23}x' + e_3 a_{12}yy' + e_3 a_{13}y' + e_3 a_{11}xy' = 0 \end{aligned} \quad (\text{B.16})$$

$$\begin{bmatrix} e_1 & e_2 & e_3 \end{bmatrix} \begin{bmatrix} a_{22}y + a_{23} + a_{21}x - a_{33}y' - a_{32}yy' - a_{31}xy' \\ -a_{13} - a_{11}x - a_{12}y + a_{33}x' + a_{32}yx' + a_{31}xx' \\ -a_{21}xx' - a_{22}yx' - a_{23}x' + a_{12}yy' + a_{13}y' + a_{11}xy' \end{bmatrix} = 0 \quad (\text{B.17})$$

$$\begin{bmatrix} a_{22}y + a_{23} + a_{21}x - a_{33}y' - a_{32}yy' - a_{31}xy' \\ -a_{13} - a_{11}x - a_{12}y + a_{33}x' + a_{32}yx' + a_{31}xx' \\ -a_{21}xx' - a_{22}yx' - a_{23}x' + a_{12}yy' + a_{13}y' + a_{11}xy' \end{bmatrix}^T \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = 0 \quad (\text{B.18})$$

let

$$\begin{aligned} b &= a_{22}y + a_{23} + a_{21}x - a_{33}y' - a_{32}yy' - a_{31}xy' \\ c &= -a_{13} - a_{11}x - a_{12}y + a_{33}x' + a_{32}yx' + a_{31}xx' \\ d &= -a_{21}xx' - a_{22}yx' - a_{23}x' + a_{12}yy' + a_{13}y' + a_{11}xy' \end{aligned} \quad (\text{B.19})$$

$$\begin{bmatrix} b & c & d \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = 0 \quad (\text{B.20})$$

For n tracked features equation B.20 can be expanded to give equation B.21

$$\begin{bmatrix} b_1 & c_1 & d_1 \\ \vdots & \vdots & \vdots \\ b_n & c_n & d_n \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = 0 \quad (\text{B.21})$$

This is of the form of the standard $Ax = b$ system of linear equations for which there are many method of solution available.

With at least 3 tracked features the epipole, and therefore the fundamental matrix can be recovered linearly up to a scale factor. This is done by finding the least squares solution to the set of linear equations in equation B.21. If only two tracked features are available, the epipole can still be recovered in a non-

linear way if the singularity constraint is imposed, in a similar way as the 7-point algorithm presented in [28].

Once a solution to equation B.21 has been found giving an estimate of the epipole, e' , the fundamental matrix can be formed using equation B.4.

B.2 Normalised three point fundamental matrix algorithm with known rotation

Normalisation of the feature points is usually desired to improve the conditioning of the problem [28]. This is normally performed by a translation and scaling of each point in the image by a transformation matrix, T , such that the points are centred around the origin and have an RMS distance from the origin of $\sqrt{2}$.

The points are then transformed such that

$$\begin{aligned}\hat{x} &= Tx \\ \hat{x}' &= T'x\end{aligned}\tag{B.22}$$

Therefore by equation B.6

$$\hat{x}'^T \hat{F} \hat{x} = 0\tag{B.23}$$

where

$$\hat{F} = T'^{-T} F T^{-1}\tag{B.24}$$

If this is applied to equation B.4 we get

$$\hat{F} = T'^{-T} [e']_\times K' R^{-1} K^{-1} T^{-1}\tag{B.25}$$

which can also be written as

$$\hat{F} = [T'e']_\times T' K' R^{-1} K^{-1} T^{-1}\tag{B.26}$$

This normalisation can be applied to the 3 point algorithm from section B.1

to be able to calculate the fundamental matrix using normalised image points.

x and x' will be replaced by \hat{x} and \hat{x}' respectively and the A matrix from equation B.9 will be replaced by

$$A = T'K'R^{-1}K^{-1}T^{-1} \quad (\text{B.27})$$

Then the resultant vector calculated from equation B.21 will be

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = T'e' \quad (\text{B.28})$$

From this the epipole can be found as

$$e' = T'^{-1} \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \quad (\text{B.29})$$

and the fundamental matrix formed using equation B.4 as usual.