Observability and Disparity in Monocular SLAM: A Study

by

Talha Manzoor

Submitted to the Department of Computer Science
on May 14, 2013, in partial fulfillment of the
requirements for the degree of
Master of Science in Computer Engineering

Abstract

In this thesis, some properties of monocular or bearing-only SLAM have been investigated for a planar robot. An observability analysis can be used to determine necessary conditions for the error to converge in an EKF. However, such an analysis is too complex to be carried out even by a computer algebraic package in its fully non-linear form. Such an analysis for low dimensional state spaces has been presented here where the robot is restricted to move on a line in a 2 dimensional world. During this analysis it is observed that including disparity as a separate measurement influences the observability of the system. Taking inspiration from this, an EKF for monocular SLAM has been setup and the effects of including disparity measurements have been investigated in simulation. In the process, the fact that the disparity measurement is dependent on the previous state, has been catered for by generalizing the EKF in the basic bayes filter framework. As a metric for observability, the eigenvalues of the state covariance matrix have been used. The results suggest that including disparity measurements in monocular SLAM increases observability in the state space, especially the states representing the pose of the robot.

Thesis Supervisor: Abubakr Muhammad
Title: Assistant Professor

Acknowledgements

Before I acknowledge anyone else, I thank Allah the Almighty (Glorified and Exalted be He), without the will of whom, none of this would be possible.

Next I would like to thank my supervisor Dr. Abubakr Muhammad for his mentorship and guidance during these past few years. It is because of him and the environment he has provided that I have been able to continue work in the field of robotics and to realize that there are academic horizons beyond those which I previously envisioned. I would like to thank all members of the CYPHYNETS laboratory at LUMS. Thanks to Hasan Arshad Nasir, Mhequb Hayat, Zeeshan Shareef and Zubair Ahmed for their guidance and encouragement during the overlapping time of our stays here. Especially to Syed Atif Adnan who turned working as colleagues into an excuse to be great friends. Thanks to Syed Muhammad Abbas who gave me the satisfaction of true companionship as we continue to pursue our academic careers together. Thanks to Bilal Talat Choudhary for his friendly ways in the lab. Thanks also to Adnan Munawar for the late-night, stress-relieving visits to Mc Donalds. Thanks to my roommate Faiz Alam for a wonderful and memorable time together. And to Asim Raza for everything, especially the late-night tea-breaks.

In the end I would like to thank Mehwish Pervaiz for being a source of comfort during my hard times, for sharing my happiness in good times and for making me feel special both inside and outside the academic world. But of everyone, I am most thankful to my parents who have made me what I am today. To my father, Dr. Manzoor Hussain for letting me be what my heart desired, for teaching me the ways of the world and providing for me, a life without liabilities. To my mother, Naheed Manzoor for whom I have absolutely no words. Thank you mother. This thesis is dedicated to you.

Contents

1 Introduction **1**

2 Related work **5**

2.1 Observability of Monocular SLAM 5

2.1.1 For a localizing robot in 2D 5

2.1.2 Piecewise constant approximated bearing only SLAM 7

2.1.3 Unobservable configurations for positioned based visual servoing 8

2.2 Combining disparity and projective measurements 10

2.3 Motion stereo for terrain mapping on the Mars exploration project . . 11

2.4 Observability of the Kalman Filter 12

3 Observability for a Monobot performing monocular SLAM **13**

3.1 The observability criterion for linear systems 14

3.2 Case 1: Monobot localizing in a 1D world 14

3.3 Case 2: Monobot performing SLAM in a 1D world 16

3.4 Case 3: Monobot performing SLAM in 1D with a beacon 18

3.4.1 For a general number of landmarks 20

3.5 The observability criterion for non-linear systems 22

3.6 Case 4: Monobot performing SLAM in a 2D world with a beacon . . 23

3.7 Case 6: Monobot performing SLAM in a 2D world with disparity measurements 27

3.8 The unsolvable cases 30

4 A generalized EKF for measurement models dependent on the previous state **33**

4.1 Definition of functions and variables 33

4.1.1 Variables 34

4.1.2 Functions 34

4.2 Modified Extended Kalman Filter equations 36

4.3 Mathematical Derivation 36

4.3.1 Part 1: Prediction 37

4.3.2 Part 2: Measurement Covariance 44

4.3.3 Part3: Correction / Update 47

5 Setting up the EKF: calculations, simulation and results **51**

5.1 Mathematical models 51

5.1.1 The constant velocity model 51

5.1.2 Model for projective measurements 53

5.1.3 Model for disparity measurements 55

5.1.4 Simulating the disparity measurements 57

5.2 Simulation results 58

5.2.1 Linear trajectory 58

5.2.2 Rhombus shaped trajectory 60

6 Future works **63**

6.1 Dealing with beacon obscurity through Beaconization 63

6.1.1 Procedure 63

6.1.2 Finding the steady state covariance 64

6.1.3 Increasing the measurement noise covariance 65

6.1.4 Revisiting a beacon with less covariance 65

6.2 Other future works 66

7 Conclusion **67**

A MATLAB: Process model **71**

B MATLAB: Measurement Model **72**

C MATLAB: Generating disparity **76**

D MATLAB: EKF Simulation **78**

List of Figures

1-1 Projective measurement of a landmark on the image plane of a camera 2

1-2 The disparity measurement of a landmark 3

2-1 While observing 2 landmarks and in motion, the robot cannot localize if it is moving on Δ_1, Δ_2 or Δ_3. 6

2-2 The circle defined by the 3 beacons. The robot cannot localize if it is stationary and on this circle. 6

3-1 The monobot localizng in a 1D world. Known quantities are highlighted in blue. 15

3-2 The monobot performing SLAM in a 1D world 17

3-3 The monobot performing SLAM in 1D with a beacon. Known quantities are highlighted in blue and the beacon is shown in yellow. 19

3-4 The monobot performing SLAM in a 2D world with 1 beacon. Known quantities are highlighted in blue. 24

3-5 The monobot performing SLAM in a 2D world with motion stereo . . 27

3-6 Monocular SLAM in 2D with a cartesian robot (no rotation) 31

3-7 2D monocular SLAM with full degrees of freedom. 32

5-1 Monocular state sequence in 2D 53

5-2 The disparity measurement 56

5-3 Estimated trajectory and 95% confidence ellipses for monocular SLAM (without disparity measurements) with 2 beacons. The beacons are highlighted in yellow and the landmark begin estimated is white. . . . 59

5-4 Estimated trajectory and 95% confidence ellipses for monocular SLAM (with disparity measurements) with 2 beacons. 60
5-5 A comparison of the trace of the covariance matrices and SSD errors in the robot pose and landmark states for Figures 5-3 and 5-4. 61
5-6 Estimated trajectory and 95% confidence ellipses for monocular SLAM (without disparity measurements) with 3 beacons. 61
5-7 Estimated trajectory and 95% confidence ellipses for monocular SLAM (with disparity measurements) with 3 beacons. 62
5-8 A comparison of the trace of the covariance matrices and SSD errors in the robot pose and landmark states for Figures 5-6 and 5-7. 62

Chapter 1

Introduction

Simultaneous Localization And Mapping or SLAM refers to the problem of a robot simultaneously building a map of the environment and using that map to keep track of its location at all times. The ability of solving the SLAM problem is essential for any mobile robot which is required to operate autonomously to some extent. Since the idea was first presented [18], SLAM has been studied and implemented in many mathematical frameworks most of which are direct or indirect forms of the Kalman, particle or basic bayes filter. Apart from various mathematical frameworks, SLAM has been studied and developed using many different sensors which include but are not limited to laser range finders, sonars, infra red sensors, cameras and even wifi.

A camera is a very attractive choice for using as a sensor in SLAM because of its low cost, minimal hardware and the amount of information that can be extracted from its reading. The first ever monocular SLAM system developed by Davison [5] used the Extended Kalman Filter (EKF) framework for SLAM while using the particle filter independently for feature initialization. Since then, the EKF framework has remained more or less the same with improvements being made in feature initialization, loop closing, depth parameterization and so on.

A camera reading gives the projection of a point feature on the image plane of the camera. The measurement system is shown in Figure 1-1. It can be seen that given the projective measurement of a point feature on the image plane, one is able to tell the ray along which the feature lies in 3D but the depth along that ray is unknown hence

the term bearing-only SLAM. Thus there is an inherent loss of information which is not present in range finding sensors like lasers and sonars. This generates a question of the extent to which projective measurements can reconstruct the state (robot pose and map). Although successful implementation of such a system may indicate empirically that SLAM is possible through projective or bearing measurements only, a theoretical analysis is necessary to gain an overall understanding of the properties of the system.

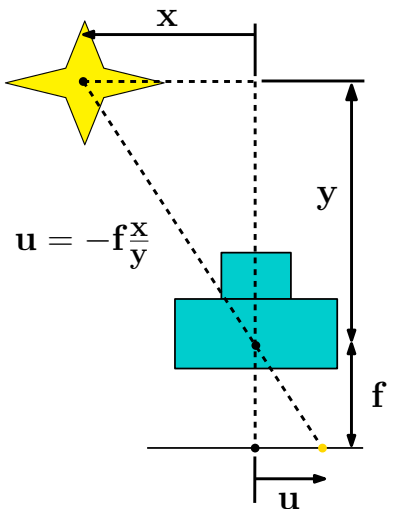

Figure 1-1: Projective measurement of a landmark on the image plane of a camera

Control theorists carry out such analysis by studying the observability of such systems. Observability answers exactly the question of whether or not state reconstruction is possible, given the system dynamics and sensor model. Such analysis have been carried out in various forms for concluding whether or not monocular SLAM is observable. Criteria for observability have also been used to answer questions like how many landmarks must the robot observe in order to preserve observability, and given a placement of landmarks which parts of the environment constitute the observable subspace for monocular SLAM.

An observability analysis amounts to checking the rank of a certain observability matrix. While the construction of this matrix is fairly straight forward for linear systems, it is slightly more involved for nonlinear ones. Whereas nonlinear observability

conditions for SLAM using other sensors have been calculated in the past, a fully nonlinear observability analysis for monocular SLAM has not yet been carried out at least to the author's knowledge. Chapter 3 of this thesis gives a bottom up observability analysis starting from the simplest case possible (localization in a 1D world) to a 1.5D scenario (SLAM in a 2D world with motion restricted along a straight line). An observability analysis for the full 2D case is computationally too complex even for a computer algebraic package.

Another type of measurement which can be provided by a camera is disparity. Disparity gives the difference between the projection of a point on the image planes of 2 row aligned cameras. A visual representation of the disparity measurement is shown in Figure 1-2. It can be seen that while projective measurements give information about the bearing to a point, disparity gives information about the depth. Disparity is commonly measured by a stereo camera pair. However as in the case of monocular SLAM where only one camera is available, disparity is measured by the same camera but at different different positions in time. Such stereo is commonly known as wide baseline stereo or motion stereo. In Chapter 3 it has been shown theoretically that including disparity along with projective measurements in monocular SLAM, does indeed effect its observability.

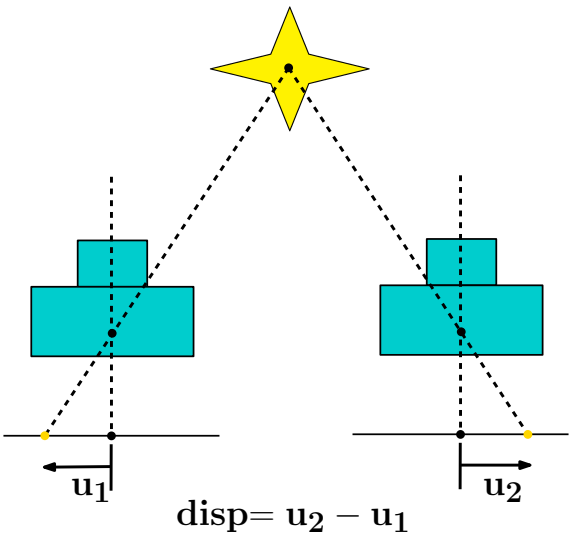

Figure 1-2: The disparity measurement of a landmark

If disparity measurements are extracted from a single camera in the motion stereo setup, it is dependent on the location of the camera during the last time step. Thus disparity is a measurement that is dependent on previous state variables. The standard Kalman Filter however assumes that the measurement is dependent only on the current state. In Chapter 4, the effect of such a dependence has been calculated. This has been done by deriving the Extended Kalman filter from the basic Bayes filter following the outline of the proof given in [22]. It turns out that this dependence will cause an increase in the measurement noise covariance. This increase depends on the process noise covariance and state dynamics.

After the EKF has been derived for measurement models dependent on the previous state, a simulation has been implemented to compare monocular SLAM with and without disparity measurements. It has been shown in [8] that the degree of observability can be seen by the magnitude of the eigenvalues of the state covariance matrix. Thus the trace of the covariance matrix has been used to compare between both cases. A decrease in the trace has been observed both for the robot location, and map variables. The decrease in the trace for the map is significantly lower than that of the robot pose, which makes this technique quite suitable for monocular SLAM setups in which maps are maintained by separate high level algorithms anyway.

Chapter 2

Related work

This chapter Includes a discussion of prior work related to this thesis.

2.1 Observability of Monocular SLAM

All practical monocular SLAM systems are inherently non-linear. Criteria of observability for non-linear systems were presented first by Hermann and Krener in 1977 [9]. The criteria is equivalent to calculating the rank of an observability matrix. However, this matrix is extremely complex to construct for systems of our interest and hence as of yet, a complete non-linear analysis for monocular SLAM has not been carried out, at least to the author's knowledge. However, researchers have carried out work in this domain by geometrical reasoning, simulation or approximating the system to some workable form.

2.1.1 For a localizing robot in 2D

Bonnifait and Garcia in their work [4] have considered a 2-D mobile robot which uses odometry and azimuth angles of known landmarks or beacons placed in the environment in order to localize itself. This is not a SLAM problem, but a localization problem. However, it is of interest over here because of the bearing-only measurements. Bonnifait and Garcia have conducted an observability analysis using

the criteria presented in [9] and also through a geometric reasoning.

For the case in which the robot observes only 1 beacon, they find the system unobservable. For the case of 2 beacons, the system is always unobservable if the robot is stationary. If the robot is moving, they find the system observable for all locations except the 3 lines given in Figure 2-1. For the case of 3 beacons, if the robot is stationary, the system is unobservable only if the robot is located on the circle shown in Figure 2-2. If it is moving, the system is observable everywhere, including this circle.

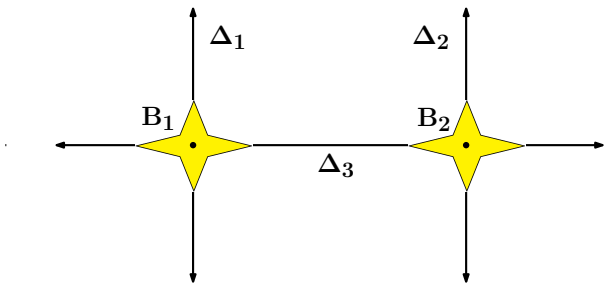

Figure 2-1: While observing 2 landmarks and in motion, the robot cannot localize if it is moving on Δ_1, Δ_2 or Δ_3.

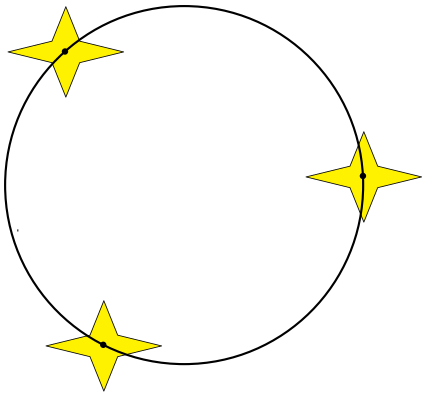

Figure 2-2: The circle defined by the 3 beacons. The robot cannot localize if it is stationary and on this circle.

2.1.2 Piecewise constant approximated bearing only SLAM

Vidal-Calleja et al, give an observability analysis for a 2-D bearing-only SLAM process in their work [25]. They evaluate the observability criterion analytically by performing the following steps

- Recast the SLAM filter in its error form so that instead of estimating the state directly, the filter estimates the error in the state.
- Model the recasted system as piecewise continuous in the time domain. The same procedure has been described in detail by Kim and Sukkarieh for inertial SLAM [10].
- Use the observability analysis of piecewise constant systems as given by Goshen-Meskin and Bar-Itzhack [7].

The observability rank criterion for piecewise constant systems is evaluated on a certain "stripped" observability matrix (SOM). Observability may be calculated over a single time segment (the interval during which the system dynamics are constant) or over multiple time segments. Vidal-Calleja et al extend the analysis further to calulate the instantaneous unobservable modes of the system and also the unobservable directions in the state space. The latter is done by observing the null space of the SOM. They perform the analysis for 3 different scenarios and get the following results

- **world centric model**: In this scenario, the robot performs absolute coordinates SLAM. If only a single landmark is being sensed by the robot, the following is observed
  - Over a single time segment, the robot orientation is the only observable state. The nullspace basis of the SOM (which represent the unobservable directions) is orthogonal to the orientation state. This applies only if the robot is in motion. If it is stationary, then the orientation becomes unobservable.

  - Adding more time segments to the SOM reduces the dimension of the unobservable subspace, but the process remains only partially observable no matter how many time segments are considered.

- **world centric with anchors:** In this scenario, the robot is provided with features whose locations are known beforehand (anchors). These are the same types of features that are reffered to as beacons in this thesis. It is found in this case that the SLAM process is observable over 2 time segments if at least 1 anchor is available.

- **sensor centric model:** In this scenario, the robot performs SLAM in relative coordinates. Note that no assumption on the availability of beacons has been made. The following results are obtained

  - The sytem is unobservable over a single time segment.
  - The system becomes fully observable over 2 time segments. The 2 exceptions for this are when the robot undergoes purely rotational motion, or when one of the features exactly aligns with the line of sight of the robot.

2.1.3 Unobservable configurations for positioned based visual servoing

Tribou et al [23] have considered the pose estimation problem for a position based visual servoing system. The setup consists of a camera being mounted on the end-effector of a robot, to keep track of the pose (position and orientation) of a moving target relative to the camera coordinate frame. The problem directly relates to monocular SLAM in relative coordinates. The observability analysis in this work focuses on deriving unobservable configurations over the state space. This problem differs from the one considered by Vidal-Calleja et al [25] because Vidal-Calleja et al have modelled the robot dynamics with a unicycle whereas Tribou et al have used a constant velocity model. The unobservable subspace of the unicycle is a subspace of the unobservable subspace of the constant velocity model.

The estimated state here consists of the tracked object's position and orientation. The orientation is reconstructed from multiple points located on the object. Two different scenarios are considered

- **known object model observability:** In this scenario, the model of the object is given in terms of the coordinates of a set of points on the body, relative to its local frame. The system is linearized at each point and observability is evaluated using the criterion for linear systems. The following results are obtained
  - If the point features and camera center lie on a common circle, the system is unobservable. This is the same circle shown in Figure 2-2.
  - The system is also unobservable if the depth of any of the feature points goes to infinity.
  - The sytem is observable everywhere else.
- **unknown object model observability:** In this scenario, the model of the object is not known, and all feature points on the model are being tracked by the camera. The peicewise constant systems observability theory of [7] has been used over here whereby observability is evaluated by calculating the rank of the stripped observability matrix. The following results are obtained
  - The system is always unobservable over a single time segment.
  - The system is unobservable over 2 time segments if the depth of any of the feature points goes to infinity.
  - The system is unobservable over 2 time segments if the projection of the first feature point on the image plane of the camera remains same during both time segments.
  - The system is unobservable over 2 time segments if the camera undergoes a pure rotational motion or when the feature points are collinear and so give the same projective measurements at both time segments.
  - For 2 time segments, the system is observable everywhere else.

2.2 Combining disparity and projective measurements

While performing SLAM from a camera(s), there is a choice between using a single camera only, or a stereo pair. Single camera systems use the projective measurements of Figure 1-1 and stereo pair systems use the disparity measurements of Figure 1-2. Many SLAM systems have successfully been implemented separately for both. The suggestion for using both measurements in in a single system was given by Sola et al [19]. However, Paz et al [14] were the first to implement a visual SLAM system combining both types of measurements. Their paper [14] also contains a brief comparison between monocular and stereo SLAM systems.

- **Monocular SLAM systems:** Such systems are capable of detecting very distance features. However, the overall system is observable only up to a scale factor. The scale is determined either by observing some known features (like a calibration procedure) or by providing the scale manually. Over a large period of time, drift in scale can occur. This drift can be fixed by applying map corrections separately like in hierarchical SLAM.

- **Stereo vision SLAM systems:** Stereo camera pairs provide measurements scaled according to their baseline. The baseline is determined beforehand during calibration and so the scale ambiguity inherent in monocular systems is not present here. However, stereo fails to provide informative measurements for features that are located at large distances and hence, do not adapt well to large scale outdoor environments.

Paz et al use a hand-held stereo camera pair for SLAM. The stereo provides 3D information about nearby points and each camera individually provides bearing only information about distant points. The stereo pair is calibrated and hence it provides the scale which was otherwise unobservable by using the single cameras alone.

2.3 Motion stereo for terrain mapping on the Mars exploration project

As mentioned in Section 2.2, stereo vision does not provide meaningful measurements for distant objects. This is because disparity approaches zero as the depth of the feature approaches infinity. Disparity can be increased for a distant object by increasing the baseline of the stereo camera pair. As the environment to map grows larger and larger, the required length of the baseline becomes physically impossible to construct. An alternative solution to this is wide-baseline stereo also known as motion stereo. In conventional stereo, there are two physically separate cameras linked together by a rigid baseline, taking snapshots at the same point in time. In motion stereo there is a single camera which takes a snapshot at one location and then moves to another location and takes another snapshot. The two images are regarded as being taken from a stereo camera pair at the same instant in time. It is assumed that the environment is static, at least during the time it takes to move the camera from one location to the other. And so, we have a virtual baseline which can become larger than a physically constructed baseline for conventional stereo, hence wide-baseline stereo.

Not much literature exists for the consideration of motion stereo for monocular SLAM. Olson and Abi-Rached have explained an implementation of motion stereo for terrain mapping [13]. They have done this for the mars exploration project in which the terrain to be mapped is at large distance from the robot and so conventional stereo is not feasible. They mention the following concerns while implementing motion stereo

- The baseline between the two camera positions is not known precisely due to the uncertainty induced due to the motion of the robot. This is in contrast to conventional stereo in which the baseline is known exactly.

- Motion stereo systems are meant for large changes in viewpoints. This means that their is very less overlap between the images used for stereo reconstruction.

In the work presented in [13], the authors have dealt with both of the issues. For uncertainty in baseline, they employ a motion refinement algorithm to reduce the error,

and use template matching techniques to deal with increased change in appearance between the stereo image pairs.

2.4 Observability of the Kalman Filter

Reif et al [15] have shown that the error in the Extended Kalman Filter remains bounded if the system satisfies the non-linear observability rank condition given in [9] and the initial errors and disturbing noise terms are small enough. As mentioned earlier, this criterion may become too complex to evaluate for a typical monocular SLAM system.

Another metric for observability of the Kalman Filter has been presented by Ham and Brown [8] in terms of the eigenvalues of the state covariance matrix obtained from the Kalman Filter estimate. Ham and Brown show that the eigenvector with the largest eigenvalue represents the least observable direction in the state space, whereas the eigenvector with the smallest eigenvalue represents the most observable direction. The magnitude of the eigenvalue can be thought of as representing the "degree" of observability in that particular direction. This study has been utilized in this thesis for comparing observability between different approaches for monocular SLAM.

Chapter 3

Observability for a Monobot performing monocular SLAM

This chapter presents an observability analysis of monocular SLAM for a monobot. The "monobot" is a robot with a 1D configuration space. It is capable of moving only along a single line. A bottom up approach is used i.e an extremely simple scenario is assumed at first (monobot localizing in a 1D world), and degrees of freedom are added one by one to the problem with the observability being calculated for each scenario. The observability criterion though easy to calculate in the first few cases, becomes more and more difficult as the system progresses towards full 2D monocular SLAM. The analysis is left at the case after which the observability criterion is not calculable analytically (monobot performing SLAM in a 2D world). Although these calculations (especially for the first few cases) may seem trivial to an experienced individual, the analysis in this bottom up manner is insightful and in the author's view, carries a tutorial value for someone new to this problem. Observability criteria for linear and nonlinear systems are defined on the fly as the need for them arises. The constant velocity model [2] has been used throughout for modeling the system dynamics

3.1 The observability criterion for linear systems

Consider the following discrete time system

$$\begin{aligned} X_k &= FX_{k-1} \\ Y_k &= HX_k \end{aligned} \tag{3.1}$$

Note that there is no assumption about control input in the system. This is because the systems dealt with in this thesis are assumed to be of such type only.
The system of Equation 3.1 is said to be completely observable if every state X_0 can be determined from observation of the measurement Y_k over a finite number of time steps. This is a fundamental concept for modern control and its treatment and description can be found in many control systems textbooks [12, 6, 20].
The given system is observable if the rank of the following *observability* matrix is full

$$\mathcal{O} = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \\ HF^{n-1} \end{bmatrix}$$

where n is the dimension of the state vector. This is the observability rank criterion for linear systems.

3.2 Case 1: Monobot localizing in a 1D world

Consider the robot contrained to move on a line and observing a single known landmark to estimate its position (Figure 3-1).
The state vector of the system is given by

$$X_k = \begin{bmatrix} {}^c x_k \\ {}^c \dot{x}_k \end{bmatrix}$$

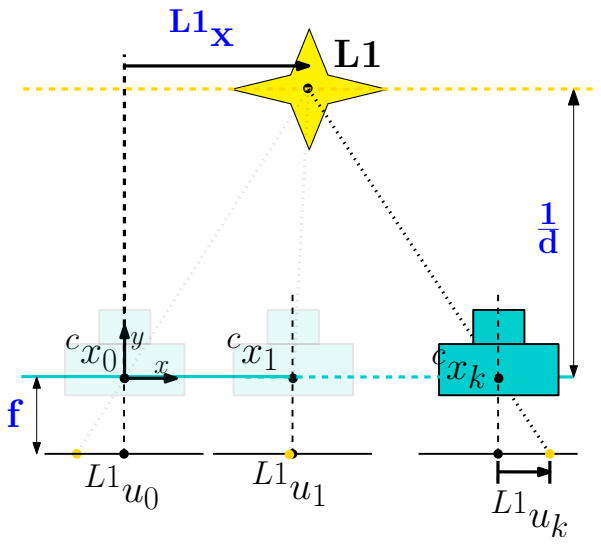

Figure 3-1: The monobot localizng in a 1D world. Known quantities are highlighted in blue.

Process is taken to be the constant velocity model. It is given as

$$X_k = FX_{k-1} \iff \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_{k-1} \\ {}^{c}\dot{x}_{k-1} \end{bmatrix}$$

The measurement is the projection of the landmark on the image plane and can be found through similar triangles as

$$\frac{{}^{L1}u_k}{{}^{L1}x - {}^{c}x_k} = \frac{f}{1/d}$$
$${}^{L1}u_k = {}^{L1}xfd - {}^{c}x_k fd$$

As the position of the landmark is known, the first term on the right hand side is just a bias. With this term included, we cannot write the measurement as a linear combination of the state variables. To do so, we define a new measurement

$${}^{L1}h_k = {}^{L1}u_k - {}^{L1}xfd = -{}^{c}x_k fd$$

or in matrix form as

$$Y_k = HX_k \iff \begin{bmatrix} {}^{L1}h_k \end{bmatrix} = \begin{bmatrix} -fd & 0 \end{bmatrix} \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \end{bmatrix}$$

Hence for our process

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}; \quad H = \begin{bmatrix} -fd & 0 \end{bmatrix}$$

The observability matrix will be given by

$$O = \begin{bmatrix} H \\ HF \end{bmatrix} = \begin{bmatrix} -fd & 0 \\ -fd & -\Delta t fd \end{bmatrix}$$

The observability matrix turns out to be full rank and hence the process is observable.

Result: Monocular localization of the monobot in a 1D world is observable.

3.3 Case 2: Monobot performing SLAM in a 1D world

Now assume that d remains known and ${}^{L1}x$ is unknown, i.e it is to be estimated (Figure 3-2).

The state vector will now become

$$X_k = \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L1}x_k \end{bmatrix}$$

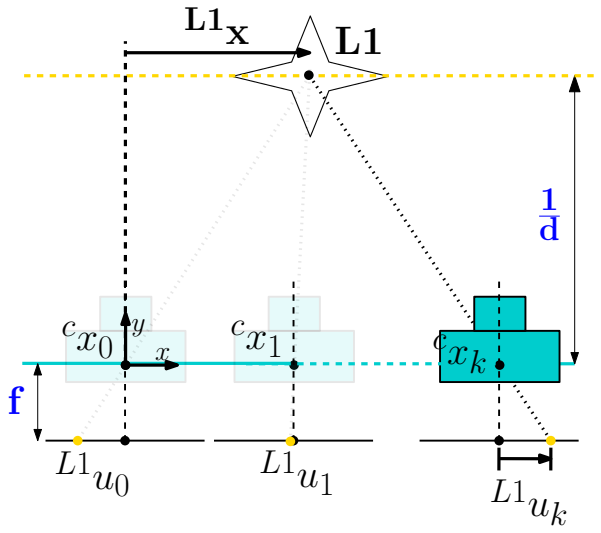

Figure 3-2: The monobot performing SLAM in a 1D world

Process is taken to be the constant velocity model given as

$$X_k = FX_{k-1} \iff \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L1}x_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_{k-1} \\ {}^{c}\dot{x}_{k-1} \\ {}^{L1}x_{k-1} \end{bmatrix}$$

The measurement is given by

$${}^{L1}u_k = {}^{L1}\, xfd - {}^{c}x_k fd$$

or in matrix form as

$$Y_k = HX_k \iff Y_k = [-fd \quad 0 \quad fd] \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L1}x_k \end{bmatrix}$$

Hence for our process

$$F = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad H = [-fd \quad 0 \quad fd]$$

The observability matrix will be given by

$$O = \begin{bmatrix} H \\ HF \\ HF^2 \end{bmatrix} = \begin{bmatrix} -fd & 0 & fd \\ -fd & -\Delta t fd & fd \\ -fd & -2\Delta t fd & fd \end{bmatrix}$$

So the observability matrix is rank deficient.

Result: Monocular SLAM of the monobot in a 1D world is unobservable.

3.4 Case 3: Monobot performing SLAM in 1D with a beacon

The same scenario as Case 2 is assumed except that a known landmark (beacon) has now been added (Figure 3-3).

The state vector will now become

$$X_k = \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \end{bmatrix}$$

Process is taken to be the constant velocity model given as

$$X_k = FX_{k-1} \Longleftrightarrow \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_{k-1} \\ {}^{c}\dot{x}_{k-1} \\ {}^{L2}x_{k-1} \end{bmatrix}$$

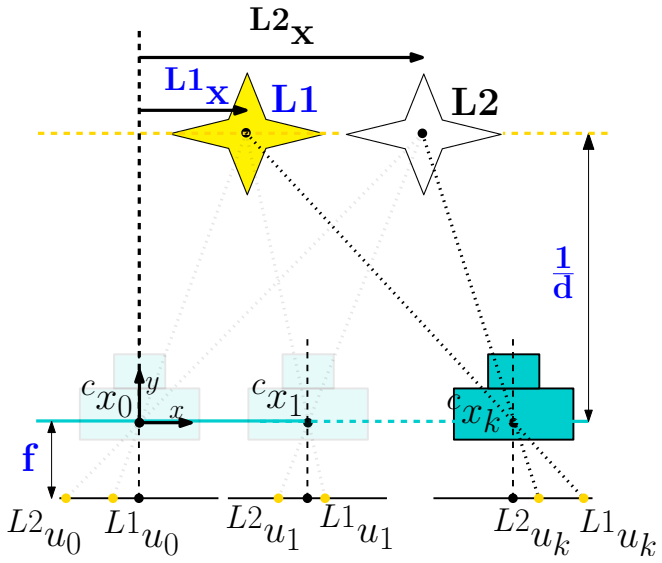

Figure 3-3: The monobot performing SLAM in 1D with a beacon. Known quantities are highlighted in blue and the beacon is shown in yellow.

We now have two measurements. They are given by

$$^{L1}h_k = {}^{L1}u_k - {}^{L1}xfd = -{}^{c}x_k fd \tag{3.2}$$

$$^{L2}h_k = {}^{L2}u_k = {}^{L2}x_k fd - {}^{c}x_k fd \tag{3.3}$$

or in matrix form as

$$Y_k = HX_k \iff \begin{bmatrix} {}^{L1}h_k \\ {}^{L2}h_k \end{bmatrix} = \begin{bmatrix} -fd & 0 & 0 \\ -fd & 0 & fd \end{bmatrix} \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \end{bmatrix}$$

Hence for our process

$$F = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; \quad H = \begin{bmatrix} -fd & 0 & 0 \\ -fd & 0 & fd \end{bmatrix}$$

The observability matrix will be given by

$$O = \begin{bmatrix} H \\ HF \\ HF^2 \end{bmatrix} = \begin{bmatrix} -fd & 0 & 0 \\ -fd & 0 & fd \\ -fd & -\Delta t fd & 0 \\ -fd & -\Delta t fd & fd \\ -fd & -2\Delta t fd & 0 \\ -fd & -2\Delta t fd & fd \end{bmatrix}$$

The observability matrix has full rank and hence the process is now observable.

Result: Monocular SLAM of the monobot in a 1D world with a single beacon is observable.

3.4.1 For a general number of landmarks

It is discovered over here, what will the observability of the same case be if the robot is observing any number of landmarks. Will one beacon be enough? Consider a scenario with i number of landmarks. Out of these i landmarks, one is perfectly known and is not included in the state vector. It can be seen that the system matrices will be given as

$$Y_k = HX_k \iff \begin{bmatrix} {}^{L1}h_k \\ {}^{L2}h_k \\ {}^{L3}h_k \\ \vdots \\ {}^{Li}h_k \end{bmatrix}_{i\times 1} = \begin{bmatrix} -fd & 0 & 0 & 0 & \cdots & 0 \\ -fd & 0 & fd & 0 & \cdots & 0 \\ -fd & 0 & 0 & fd & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -fd & 0 & 0 & 0 & \cdots & fd \end{bmatrix}_{i\times(i+1)} \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \\ \vdots \\ {}^{Li}x_k \end{bmatrix}_{(i+1)\times 1}$$

and

$$X_k = FX_{k-1} \iff \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \\ \vdots \\ {}^{Li}x_k \end{bmatrix}_{(i+1)\times 1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}_{(i+1)\times(i+1)} \begin{bmatrix} {}^{c}x_{k-1} \\ {}^{c}\dot{x}_{k-1} \\ {}^{L2}x_{k-1} \\ \vdots \\ {}^{Li}x_{k-1} \end{bmatrix}_{(i+1)\times 1}$$

For i landmarks, the system is of order $i+1$ and hence the observability matrix will be given by

$$O = \begin{bmatrix} H \\ HF \\ HF^2 \\ \vdots \\ HF^i \end{bmatrix} = \begin{bmatrix} -fd & 0 & 0 & 0 & \cdots & 0 \\ -fd & 0 & fd & 0 & \cdots & 0 \\ -fd & 0 & 0 & fd & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -fd & 0 & 0 & 0 & \cdots & fd \\ -fd & -\Delta tfd & 0 & 0 & \cdots & 0 \\ -fd & -\Delta tfd & fd & 0 & \cdots & 0 \\ -fd & -\Delta tfd & 0 & fd & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -fd & -\Delta tfd & 0 & 0 & \cdots & fd \\ -fd & -2\Delta tfd & 0 & 0 & \cdots & 0 \\ -fd & -2\Delta tfd & fd & 0 & \cdots & 0 \\ -fd & -2\Delta tfd & 0 & fd & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -fd & -2\Delta tfd & 0 & 0 & \cdots & fd \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -fd & -i\Delta tfd & 0 & 0 & \cdots & 0 \\ -fd & -i\Delta tfd & fd & 0 & \cdots & 0 \\ -fd & -i\Delta tfd & 0 & fd & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ -fd & -i\Delta tfd & 0 & 0 & \cdots & fd \end{bmatrix}_{(i+i^2)\times(i+1)}$$

It can be observed that this matrix has full rank hence the process is observable.

Result: Monocular SLAM of the monobot in a 1D world and a single beacon is observable for any number of landmarks.

3.5 The observability criterion for non-linear systems

Consider the following non-linear discrete time system

$$\begin{aligned} X_k &= f(X_{k-1}) \\ Y_k &= h(X_k) \end{aligned} \tag{3.4}$$

Define a matrix G, given by

$$G = \begin{bmatrix} \mathcal{L}_f^0(h_1) \\ \cdots \\ \mathcal{L}_f^0(h_p) \\ \vdots \\ \mathcal{L}_f^{n-1}(h_1) \\ \cdots \\ \mathcal{L}_f^{n-1}(h_p) \end{bmatrix}$$

where $\mathcal{L}_f^n(h_p)$ is the nth order Lie Derivative [17] of the scalar field of the pth measurement with respect to the vector field f. The first order Lie Derivative is defined as

$$\mathcal{L}_f h = \nabla h.f = \begin{bmatrix} \frac{\partial h}{\partial x_1} \cdots \frac{\partial h}{\partial x_n} \end{bmatrix} \begin{bmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{bmatrix}$$

where

$f : \mathbb{R}^n \to \mathbb{R}^n$ is a vector field in $\mathbb{R}^n$ and $h : \mathbb{R}^n \to \mathbb{R}$ is a smooth scalar function.

By definition, the zeroth order Lie Derivative is given as,

$$\mathcal{L}_f^0(h) = h$$

and the nth order Lie Derivative is given by

$$\mathcal{L}_f^n(h) = \frac{\partial}{\partial x}\left[\mathcal{L}_f^{n-1}(h)\right].f$$

Next define a gradient operator d such that

$$dG = \begin{bmatrix} d\mathcal{L}_f^0(h_1) \\ \vdots \\ d\mathcal{L}_f^0(h_p) \\ \vdots \\ d\mathcal{L}_f^{n-1}(h_1) \\ \vdots \\ d\mathcal{L}_f^{n-1}(h_p) \end{bmatrix}$$

this is in actual, the Observability matrix, which is given as

$$O = dG = \begin{bmatrix} \frac{\partial \mathcal{L}_f^0(h_1)}{\partial x_1} & \cdots & \frac{\partial \mathcal{L}_f^0(h_1)}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial \mathcal{L}_f^0(h_p)}{\partial x_1} & \cdots & \frac{\partial \mathcal{L}_f^0(h_p)}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial \mathcal{L}_f^{n-1}(h_1)}{\partial x_1} & \cdots & \frac{\partial \mathcal{L}_f^0(h_1)}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial \mathcal{L}_f^{n-1}(h_p)}{\partial x_1} & \cdots & \frac{\partial \mathcal{L}_f^{n-1}(h_p)}{\partial x_n} \end{bmatrix}$$

The system given by Equation 3.4 is observable if the rank of this matrix is full [1].

3.6 Case 4: Monobot performing SLAM in a 2D world with a beacon

The same scenario as Case 3 is assumed. Now however, the parameter d is also to be estimated. Thus the landmarks now have 2 degrees of freedom (Figure 3-4)

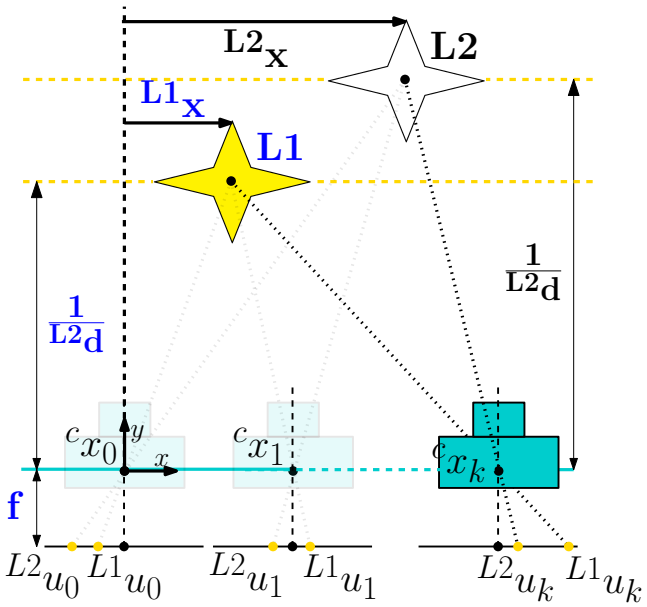

Figure 3-4: The monobot performing SLAM in a 2D world with 1 beacon. Known quantities are highlighted in blue.

The state vector is now be given by

$$X_k = \begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \\ {}^{L2}d_k \end{bmatrix}$$

The process is still linear (constant velocity) and will be given by

$$\begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \\ {}^{L2}d_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_{k-1} \\ {}^{c}\dot{x}_{k-1} \\ {}^{L2}x_{k-1} \\ {}^{L2}d_{k-1} \end{bmatrix}$$

We represent this with the vector field g (f is being used for the focal length) given by in the current case, the vector function f is given as,

$$g = \begin{bmatrix} {}^{c}x_k + \Delta t {}^{c}\dot{x}_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \\ {}^{L2}d_k \end{bmatrix}$$

The two measurements are given by the following functions

$$\begin{aligned} {}^{L1}h_k &= {}^{L1}u_k - f\ {}^{L1}x\ {}^{L1}d = -f\ {}^{L1}d\ {}^{c}x_k \\ {}^{L2}h_k &= {}^{L2}u_k = -f\ {}^{c}x_k\ {}^{L2}d_k + f\ {}^{L2}x_k\ {}^{L2}d_k \end{aligned}$$

For the given dimensions of the measurement and state vectors, the observability matrix in this case will be given by

$$O = \begin{bmatrix} \frac{\partial \mathcal{L}_g^0({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^0({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^0({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^0({}^{L1}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^0({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^0({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^0({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^0({}^{L2}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^1({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^1({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^1({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^1({}^{L1}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^1({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^1({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^1({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^1({}^{L2}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^2({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^2({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^2({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^2({}^{L1}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^2({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^2({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^2({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^2({}^{L2}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^3({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^3({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^3({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^3({}^{L1}h_k)}{\partial x_4} \\ \frac{\partial \mathcal{L}_g^3({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_g^3({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_g^3({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_g^3({}^{L2}h_k)}{\partial x_4} \end{bmatrix}$$

After calculating the Lie Derivatives, G turns out to be,

$$G = \begin{bmatrix} -f\ {}^{L1}d\ {}^{c}x_k \\ -f\ {}^{c}x_k\ {}^{L2}d_k + f\ {}^{L2}x_k\ {}^{L2}d_k \\ -f\ {}^{L1}d\ {}^{c}x_k - \Delta t f\ {}^{L1}d\ {}^{c}\dot{x}_k \\ -2f\ {}^{c}x_k\ {}^{L2}d_k - \Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + 2f\ {}^{L2}x_k\ {}^{L2}d_k \\ -f\ {}^{L1}d\ {}^{c}x_k - 2\Delta t f\ {}^{L1}d\ {}^{c}\dot{x}_k \\ -4f\ {}^{c}x_k\ {}^{L2}d_k - 4\Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + 4f\ {}^{L2}x_k\ {}^{L2}d_k \\ -f\ {}^{L1}d\ {}^{c}x_k - 3\Delta t f\ {}^{L1}d\ {}^{c}\dot{x}_k \\ -8f\ {}^{c}x_k\ {}^{L2}d_k - 12\Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + 8f\ {}^{L2}x_k\ {}^{L2}d_k \end{bmatrix}$$

The Observability Matrix will then be,

$$O = dG = \begin{bmatrix} -f\ {}^{L1}d & 0 & 0 & 0 \\ -f\ {}^{L2}d_k & 0 & f\ {}^{L2}d_k & -f\ {}^{c}x_k + f\ {}^{L2}x_k \\ -f\ {}^{L1}d & -\Delta t f\ {}^{L1}d & 0 & 0 \\ -2f\ {}^{L2}d_k & -\Delta t f\ {}^{L2}d_k & 2f\ {}^{L2}d_k & -2f\ {}^{c}x_k - \Delta t f\ {}^{c}\dot{x}_k + 2f\ {}^{L2}x_k \\ -f\ {}^{L1}d & -2\Delta t f\ {}^{L1}d & 0 & 0 \\ -4f\ {}^{L2}d_k & -4\Delta t f\ {}^{L2}d_k & 4f\ {}^{L2}d_k & -4f\ {}^{c}x_k - 4\Delta t f\ {}^{c}\dot{x}_k + 4f\ {}^{L2}x_k \\ -f\ {}^{L1}d & -3\Delta t f\ {}^{L1}d & 0 & 0 \\ -8f\ {}^{L2}d_k & -12\Delta t f\ {}^{L2}d_k & 8f\ {}^{L2}d_k & -8f\ {}^{c}x_k - 12\Delta t f\ {}^{c}\dot{x}_k + 8f\ {}^{L2}x_k \end{bmatrix}$$

We see that this matrix has full rank for all physically realizable values of the state variables.

Result: Monocular SLAM of the monobot in a 2D world and a single beacon is observable.

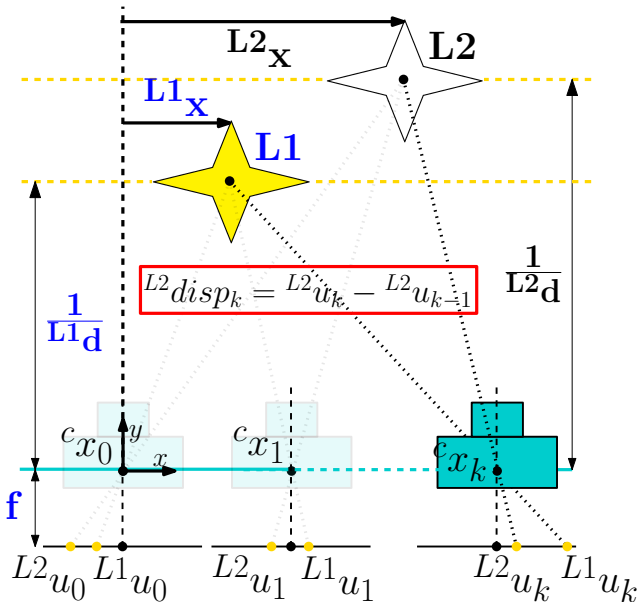

Figure 3-5: The monobot performing SLAM in a 2D world with motion stereo

3.7 Case 6: Monobot performing SLAM in a 2D world with disparity measurements

We now assume the same scenario as Case 4 except that we now use disparity (motion stereo) as an additional measurement to calculate depth (Figure 3-5).

The process model will be the same

$$\begin{bmatrix} {}^{c}x_k \\ {}^{c}\dot{x}_k \\ {}^{L2}x_k \\ {}^{L2}d_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^{c}x_{k-1} \\ {}^{c}\dot{x}_{k-1} \\ {}^{L2}x_{k-1} \\ {}^{L2}d_{k-1} \end{bmatrix}$$

We now have three non-linear measurements, the third one being disparity for the estimated landmark

$$
\begin{aligned}
{}^{L1}h_k &= {}^{L1}u_k - f\ {}^{L1}x\ {}^{L1}d = -f\ {}^{L1}d\ {}^{c}x_k \\
{}^{L2}h_k &= {}^{L2}u_k = -f\ {}^{c}x_k\ {}^{L2}d_k + f\ {}^{L2}x_k\ {}^{L2}d_k \\
{}^{L2}disp_k &= {}^{L2}u_k - {}^{L2}u_{k-1} = f(-\ {}^{c}x_k + \ {}^{c}x_{k-1})\ {}^{L2}d_k \\
&= -f\ {}^{c}x_k\ {}^{L2}d_k + f\ {}^{c}x_{k-1}\ {}^{L2}d_k
\end{aligned}
\tag{3.5}
$$

The Observability Matrix for this system will then be given as

$$
O = \begin{bmatrix}
\frac{\partial \mathcal{L}_f^0({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^0({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^0({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^0({}^{L1}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^0({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^0({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^0({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^0({}^{L2}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^0({}^{L2}disp_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^0({}^{L2}disp_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^0({}^{L2}disp_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^0({}^{L2}disp_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^1({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^1({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^1({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^1({}^{L1}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^1({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^1({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^1({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^1({}^{L2}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^1({}^{L2}disp_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^1({}^{L2}disp_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^1({}^{L2}disp_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^1({}^{L2}disp_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^2({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^2({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^2({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^2({}^{L1}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^2({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^2({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^2({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^2({}^{L2}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^2({}^{L2}disp_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^2({}^{L2}disp_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^2({}^{L2}disp_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^2({}^{L2}disp_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^3({}^{L1}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^3({}^{L1}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^3({}^{L1}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^3({}^{L1}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^3({}^{L2}h_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^3({}^{L2}h_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^3({}^{L2}h_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^3({}^{L2}h_k)}{\partial x_4} \\
\frac{\partial \mathcal{L}_f^3({}^{L2}disp_k)}{\partial x_1} & \frac{\partial \mathcal{L}_f^3({}^{L2}disp_k)}{\partial x_2} & \frac{\partial \mathcal{L}_f^3({}^{L2}disp_k)}{\partial x_3} & \frac{\partial \mathcal{L}_f^3({}^{L2}disp_k)}{\partial x_4}
\end{bmatrix}
$$

The matrix G turns out to be

$$G = \begin{bmatrix} -f\ {}^{L1}d\ {}^{c}x_k \\ -f\ {}^{c}x_k\ {}^{L2}d_k + f\ {}^{L2}x_k\ {}^{L2}d_k \\ -f\ {}^{c}x_k\ {}^{L2}d_k + f\ {}^{c}x_{k-1}\ {}^{L2}d_k \\ -f\ {}^{L1}d\ {}^{c}x_k - \Delta t f\ {}^{L1}d\ {}^{c}\dot{x}_k \\ -2f\ {}^{c}x_k\ {}^{L2}d_k - \Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + 2f\ {}^{L2}x_k\ {}^{L2}d_k \\ -2f\ {}^{c}x_k\ {}^{L2}d_k - \Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + f\ {}^{c}x_{k-1}\ {}^{L2}d_k \\ -f\ {}^{L1}d\ {}^{c}x_k - 2\Delta t f\ {}^{L1}d\ {}^{c}\dot{x}_k \\ -4f\ {}^{c}x_k\ {}^{L2}d_k - 4\Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + 4f\ {}^{L2}x_k\ {}^{L2}d_k \\ -4f\ {}^{c}x_k\ {}^{L2}d_k - 4\Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + f\ {}^{c}x_{k-1}\ {}^{L2}d_k \\ -f\ {}^{L1}d\ {}^{c}x_k - 3\Delta t f\ {}^{L1}d\ {}^{c}\dot{x}_k \\ -8f\ {}^{c}x_k\ {}^{L2}d_k - 12\Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + 8f\ {}^{L2}x_k\ {}^{L2}d_k \\ -8f\ {}^{c}x_k\ {}^{L2}d_k - 12\Delta t f\ {}^{c}\dot{x}_k\ {}^{L2}d_k + f\ {}^{c}x_{k-1}\ {}^{L2}d_k \end{bmatrix}$$

The Observability Matrix will then be,

$$O = dG = \begin{bmatrix} -f\ {}^{L1}d & 0 & 0 & 0 \\ -f\ {}^{L2}d_k & 0 & f\ {}^{L2}d_k & -f\ {}^{c}x_k + f\ {}^{L2}x_k \\ -f\ {}^{L2}d_k & 0 & 0 & -f\ {}^{c}x_k + f\ {}^{c}x_{k-1} \\ -f\ {}^{L1}d & -\Delta t f\ {}^{L1}d & 0 & 0 \\ -2f\ {}^{L2}d_k & -\Delta t f\ {}^{L2}d_k & 2f\ {}^{L2}d_k & -2f\ {}^{c}x_k - \Delta t f\ {}^{c}\dot{x}_k + 2f\ {}^{L2}x_k \\ -2f\ {}^{L2}d_k & -\Delta t f\ {}^{L2}d_k & 0 & -2f\ {}^{c}x_k - \Delta t f\ {}^{c}\dot{x}_k + f\ {}^{c}x_{k-1} \\ -f\ {}^{L1}d & -2\Delta t f\ {}^{L1}d & 0 & 0 \\ -4f\ {}^{L2}d_k & -4\Delta t f\ {}^{L2}d_k & 4f\ {}^{L2}d_k & -4f\ {}^{c}x_k - 4\Delta t f\ {}^{c}\dot{x}_k + 4f\ {}^{L2}x_k \\ -4f\ {}^{L2}d_k & -4\Delta t f\ {}^{L2}d_k & 0 & -4f\ {}^{c}x_k - 4\Delta t f\ {}^{c}\dot{x}_k + f\ {}^{c}x_{k-1} \\ -f\ {}^{L1}d & -3\Delta t f\ {}^{L1}d & 0 & 0 \\ -8f\ {}^{L2}d_k & -12\Delta t f\ {}^{L2}d_k & 8f\ {}^{L2}d_k & -8f\ {}^{c}x_k - 12\Delta t f\ {}^{c}\dot{x}_k + 8f\ {}^{L2}x_k \\ -8f\ {}^{L2}d_k & -12\Delta t f\ {}^{L2}d_k & 0 & -8f\ {}^{c}x_k - 12\Delta t f\ {}^{c}\dot{x}_k + f\ {}^{c}x_{k-1} \end{bmatrix}$$

we can see that the condition for violation of the observability rank criterion will be

$$^{c}x_k = {}^{c}x_{k-1} = 0$$

this dictates that at least 2 different measurements (from different positions) of the same landmark must be made in order to make SLAM observable in this case. This can be interpreted as an initialization criterion of a landmark for inclusion in the state vector. Hence this case is observable. However this is not a significant result as this case was observable even without the disparity measurement. However if we ignore the measurement of the beacon, our Observability Matrix turns out to be

$$O = \begin{bmatrix} -f\ {}^{L2}d_k & 0 & f\ {}^{L2}d_k & -f\ {}^{c}x_k + f\ {}^{L2}x_k \\ -f\ {}^{L2}d_k & 0 & 0 & -f\ {}^{c}x_k + f\ {}^{c}x_{k-1} \\ -2f\ {}^{L2}d_k & -\Delta t f\ {}^{L2}d_k & 2f\ {}^{L2}d_k & -2f\ {}^{c}x_k - \Delta t f\ {}^{c}\dot{x}_k + 2f\ {}^{L2}x_k \\ -2f\ {}^{L2}d_k & -\Delta t f\ {}^{L2}d_k & 0 & -2f\ {}^{c}x_k - \Delta t f\ {}^{c}\dot{x}_k + f\ {}^{c}x_{k-1} \\ -4f\ {}^{L2}d_k & -4\Delta t f\ {}^{L2}d_k & 4f\ {}^{L2}d_k & -4f\ {}^{c}x_k - 4\Delta t f\ {}^{c}\dot{x}_k + 4f\ {}^{L2}x_k \\ -4f\ {}^{L2}d_k & -4\Delta t f\ {}^{L2}d_k & 0 & -4f\ {}^{c}x_k - 4\Delta t f\ {}^{c}\dot{x}_k + f\ {}^{c}x_{k-1} \\ -8f\ {}^{L2}d_k & -12\Delta t f\ {}^{L2}d_k & 8f\ {}^{L2}d_k & -8f\ {}^{c}x_k - 12\Delta t f\ {}^{c}\dot{x}_k + 8f\ {}^{L2}x_k \\ -8f\ {}^{L2}d_k & -12\Delta t f\ {}^{L2}d_k & 0 & -8f\ {}^{c}x_k - 12\Delta t f\ {}^{c}\dot{x}_k + f\ {}^{c}x_{k-1} \end{bmatrix}$$

which is still full rank. This means the motion stereo measurement has eliminated the need for the beacon.

Result: Monocular SLAM of the monobot in a 2D world with disparity measurements is observable even without the beacon.

3.8 The unsolvable cases

There are 2 cases yet to be considered before the full nonlinear observability analysis can be done for general monocular SLAM in a 2D world. The 2 degrees of freedom left to be added are the robot translation in the y direction and the robot rotation. Adding any of these degrees of freedom to Case 5 makes the Obervability matrix

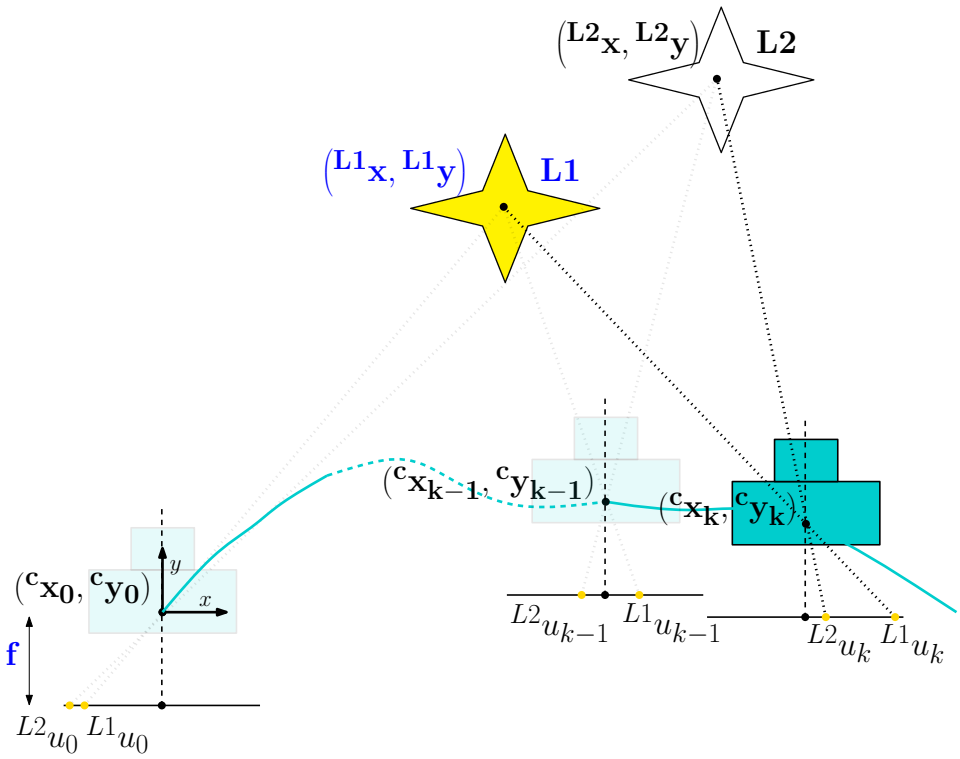

Figure 3-6: Monocular SLAM in 2D with a cartesian robot (no rotation)

too complex to be computed analytically even by a symbolic editor like MATLAB or Mapple. The Cases are depicted in Figures 3-6 and 3-7 and posed as open problems to interested readers. Perhaps some method other than the brute force analysis can be adopted like Lee et al have done in [11] for range and bearing SLAM.

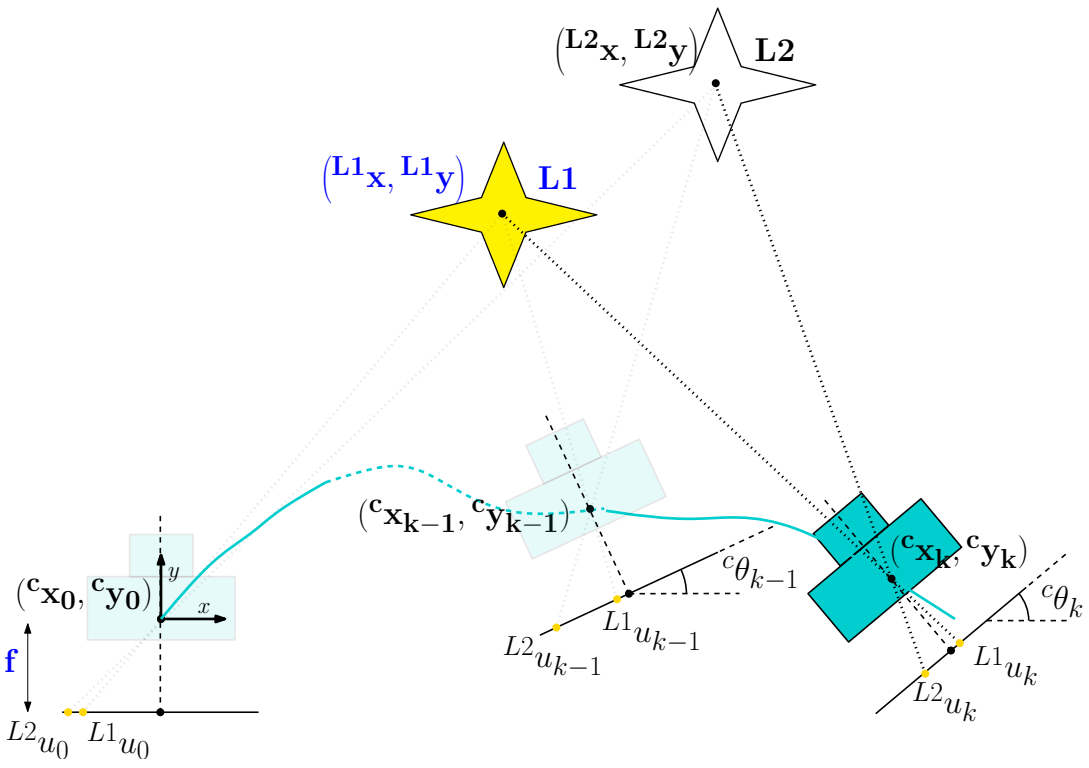

Figure 3-7: 2D monocular SLAM with full degrees of freedom.

Chapter 4

A generalized EKF for measurement models dependent on the previous state

The model for the disparity measurement as shown in Equation 3.5 is dependent on the previous and current states. However, the conventional Kalman filter assumes that the measurement is dependent on the current state only. The Kalman filter is rederived here for this case from the basic Bayes Filter following the method given by Thrun et al in [22]. In the end, it will turn out that this dependence causes an increase in the measurement noise covariance. The increase is dependent on the process noise covariance and the function relating the current and previous states.

4.1 Definition of functions and variables

This section gives the notation which will be used throughout this chapter.

4.1.1 Variables

X_k - The state vector at time step k

X_{k-1} - The state vector at time step $k-1$

μ_k - The control vector at time step k

Y_k - The measurement vector at time step k

$\hat{X}_{k|k-1}$ - The estimate of X_k given $k-1$ measurements

$P_{k|k-1}$ - The covariance of $\hat{X}_{k|k-1}$

$\hat{X}_{k|k}$ - The estimate of X_k given k measurements

$P_{k|k}$ - The covariance of $\hat{X}_{k|k}$

$\hat{Y}_k$ - The expected measurement at time step k

W_k - The covariance of $\hat{Y}_k$

4.1.2 Functions

Motion Model

The motion or process model is given by

$$X_k = f(X_{k-1}, \mu_k, \omega_k)$$

Where ω_k is the process noise at step k. Its covariance is given by Q_k. After linearization we get the following form

$$X_k = f(\hat{X}_{k-1}, \mu_k, 0) + F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + F_{\omega_k}\omega_k$$

Where

$F_{X_{k-1}}$ - is the Jacobian of f w.r.t X_{k-1}

F_{ω_k} - is the Jacobian of f w.r.t ω_k

Inverse Motion Model

The inverse motion model is given by

$$X_{k-1} = g(X_k, \mu_k, \omega_k)$$

Note that for same time steps, ω_k has the same value as its instance appearing in the motion model. After linearization we get

$$X_{k-1} = g(\hat{X}_k, \mu_k, 0) + G_{X_k}(X_k - \hat{X}_k) + G_{\omega_k}\omega_k$$

Where

G_{X_k} - is the Jacobian of g w.r.t X_k

G_{ω_k} - is the Jacobian of g w.r.t ω_k

This function can be simply written as the inverse of $f(.)$.

Measurement Model

The measurement model is represented by

$$Y_k = h(X_k, X_{k-1}) + v_k$$

Where

v_k is the sensor noise. Its covariance is given by R_k

After linearization we get

$$Y_k = h(\hat{X}_k, \hat{X}_{k-1}) + H_{X_k}(X_k - \hat{X}_k) + H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + v_k$$

Where

H_{X_k} - is the Jacobian of h w.r.t X_k

$H_{X_{k-1}}$ - is the Jacobian of h w.r.t X_{k-1}

4.2 Modified Extended Kalman Filter equations

The Extended Kalman Filter Equations for this type of measurement model are given by

Prediction:

$$\hat{X}_{k|k-1} = f(\hat{X}_{k-1|k-1}, \mu_k, 0) \tag{4.1}$$

$$P_{k|k-1} = F_{X_{k-1}} P_{k-1|k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T \tag{4.2}$$

Measurement Covariance:

$$W_k = R_k + H_{X_{k-1}} G_{\omega_k} Q_k {G_{\omega_k}}^T {H_{X_{k-1}}}^T \tag{4.3}$$

Kalman Gain:

$$K_k = P_{k|k-1} {H_{X_k}}^T (H_{X_k} P_{k|k-1} {H_{X_k}}^T + W_k)^{-1} \tag{4.4}$$

Correction:

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k(Y_k - h(\hat{X}_{k|k-1}, \hat{X}_{k-1|k-1})) \tag{4.5}$$

$$P_{k|k} = (I - K_k H_{X_k}) P_{k|k-1} \tag{4.6}$$

The difference between the EKF equations for simple measurement models [16, 22, 2] (not dependent on the previous state) and this one lies in the calculation in equation (4.3). For simple measurement models, W_k would just have been equal to the sensor noise R_k. And the rest of the equations would have remained unchanged.

4.3 Mathematical Derivation

The equations are derived first by extending the basic Bayes Filter Algorithm, to include the previous state in the measurement step. Then the probability distributions are solved for the required models. The conventional Bayes Filter algorithm is given

by

Step 1: law of total probability

$$\overline{bel(X_k)} = p(X_k|Y_{0:k-1}, \mu_k) = \int p(X_k|X_{k-1}, Y_{0:k-1}, \mu_k)p(X_{k-1}|Y_{0:k-1}, \mu_k)\mathrm{d}X_{k-1}$$

Step 2: Bayes rule

$$bel(X_k) = \eta p(Y_k|X_k, Y_{0:k-1}, \mu_k)p(X_k|Y_{0:k-1}, \mu_k)$$

To include the dependency of Y_k on X_{k-1} we include an intermediate step through the theorem of total probability. Our algorithm is now given as follows

Step 1: law of total probability

$$\overline{bel(X_k)} = p(X_k|Y_{0:k-1}, \mu_k) = \int p(X_k|X_{k-1}, Y_{0:k-1}, \mu_k)p(X_{k-1}|Y_{0:k-1}, \mu_k)\mathrm{d}X_{k-1} \quad (4.7)$$

Step 2: law of total probability

$$p(Y_k|X_k, Y_{0:k-1}, \mu_k) = \int p(Y_k|X_k, Y_{0:k-1}, \mu_k, X_{k-1})p(X_{k-1}|X_k, Y_{0:k-1}, \mu_k)\mathrm{d}X_{k-1} \quad (4.8)$$

Step 3: Bayes rule

$$bel(X_k) = \eta p(Y_k|X_k, Y_{0:k-1}, \mu_k)p(X_k|Y_{0:k-1}, \mu_k) \quad (4.9)$$

We begin our derivation, with this extended form of the Bayes filter

4.3.1 Part 1: Prediction

We begin with step 1 of the Bayes filter. Equation (4.7) has been rewritten here, with the mean and covariances of the normal distributions appearing in this step, stated

as well

$$\overline{bel(X_k)} = p(X_k|Y_{0:k-1}, \mu_k) =$$

$$\int \underbrace{p(X_k|X_{k-1}, Y_{0:k-1}, \mu_k)}_{N\sim(X_k; f(\hat{X}_{k-1},\mu_k,0)+F_{X_{k-1}}(X_{k-1}-\hat{X}_{k-1}); F_{\omega_k}Q_k{F_{\omega_k}}^T)} \underbrace{p(X_{k-1}|Y_{0:k-1}, \mu_k)}_{N\sim(X_{k-1};\hat{X}_{k-1};P_{k-1})} \mathrm{d}X_{k-1} \tag{4.10}$$

We shall now see that the outcome of this equation is also a Gaussian with mean $\hat{X}_{k|k-1}$ and covariance $P_{k|k-1}$. Writing (4.10) in its Gaussian form we get,

$$\overline{bel(X_k)} = \eta \int \exp\{-L_k\}\mathrm{d}X_{k-1}$$

where

$$\begin{aligned} L_k = \frac{1}{2}&(X_k - f(\hat{X}_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}))^T ({F_{\omega_k}}{Q_k}{F_{\omega_k}}^T)^{-1} \\ &(X_k - f(\hat{X}_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) \\ +\frac{1}{2}&(X_{k-1} - \hat{X}_{k-1})^T {P_{k-1}}^{-1}(X_{k-1} - \hat{X}_{k-1}) \end{aligned} \tag{4.11}$$

Note that L_k is quadratic both in X_k and X_{k-1} . To solve the integral in closed form, we will decompose L_k into two functions so that it is given by

$$L_k = L(X_k, X_{k-1}) + L(X_k) \tag{4.12}$$

This decomposition will enable us to move $L(X_K)$ outside the integral and so the integral will simplify to

$$\overline{bel(X_k)} = \eta\exp\{-L(X_k)\} \int \exp\{-L(X_k, X_{k-1})\}\mathrm{d}X_{k-1}$$

Also, we will choose $L(X_K, X_{k-1})$ such that the value of the integral does not depend upon X_k, so that it becomes a constant relative to the problem of determining the

belief distribution over X_k. In fact, we will choose $L(X_K, X_{k-1})$ to be of the form of the exponential function in the PDF of a Gaussian distribution. The mean and variance will be given by the minimum and inverse of the curvature of L_k, respectively. And so, the resulting distribution will be defined entirely through $L(X_k)$ as

$$\overline{bel(X_k)} = \eta \exp\{-L(X_k)\}$$

We now perform this decomposition. We are looking for a fuction $L(X_k, X_{k-1})$ which is quadratic in X_{k-1}. To determine the coefficients of this quadratic, we calculate the first two derivatives of L_k with respect to X_{k-1}

$$\begin{aligned}\frac{\partial L_k}{\partial X_{k-1}} =& - {F_{X_{k-1}}}^T (F_{X_{k-1}} Q_k {F_{X_{k-1}}}^T)^{-1} (X_k - f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) \\ & + {P_{k-1}}^{-1} (X_{k-1} - \hat{X}_{k-1})\end{aligned}$$

and

$$\frac{\partial^2 L_k}{\partial {X^2}_{k-1}} = {F_{X_{k-1}}}^T (F_{X_{k-1}} Q_k {F_{X_{k-1}}}^T)^{-1} F_{X_{k-1}} + {P_{k-1}}^{-1} = \Psi_k^{-1}$$

Where Ψ_k defines the curvature of $L(X_k, X_{k-1})$. We get the mean by setting the first derivative of L_k equal to zero. Solving and simplifying for X_{k-1} gives us

$$X_{k-1} = \Psi_k {F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} (X_k - f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + {P_{k-1}}^{-1} \hat{X}_{k-1})$$

So we have the quadratic function $L(X_k, X_{k-1})$ given as

$$\begin{aligned} L(X_k, X_{k-1}) = \\ & \frac{1}{2}(X_{k-1} - \Psi_k {F_{X_{k-1}}}^T ({F_{\omega_k} Q_k {F_{\omega_k}}^T})^{-1} \\ & (X_k - f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + {P_{k-1}}^{-1}\hat{X}_{k-1}))^T \\ & \Psi_k^{-1}(X_{k-1} - \Psi_k {F_{X_{k-1}}}^T ({F_{\omega_k} Q_k {F_{\omega_k}}^T})^{-1} \\ & (X_k - f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + {P_{k-1}}^{-1}\hat{X}_{k-1})) \end{aligned} \tag{4.13}$$

Note that this is not the only function satisfying our decomposition. However, we now have a function which is of the common quadratic form of the negative exponent of a normal distribution. The function

$$\det(2\pi\Psi_k)^{-\frac{1}{2}} \exp\{-L(X_k, X_{k-1}\}$$

is a valid probabilty density function (PDF) for the variable X_{k-1}. And we know that PDFs integrate to 1. Thus,

$$\int \det(2\pi\Psi_k)^{-\frac{1}{2}} \exp\{-L(X_k, X_{k-1})\} \mathrm{d}X_{k-1} = 1$$

From this it follows that,

$$\det(2\pi\Psi_k)^{\frac{1}{2}} = \int \exp\{-L(X_k, X_{k-1})\} \mathrm{d}X_{k-1}$$

The important thing to note is that the value of this integral is independent of our target variable X_k. Thus, for our problem of calculating a distribution over X_k, this integral is constant. Subsuming this constant into the normalizer we get the following expression for our belief

$$\overline{bel(X_k)} = \eta \exp\{-L(X_k)\} \int \exp\{-L(X_k, X_{k-1})\} \mathrm{d}X_{k-1} = \eta \exp\{-L(X_k)\}$$

Note that the normalizers η on both sides of the equal sign are *not* the same. Now we need to determine the function $L(X_k)$ which is given by (4.11), (4.12) and (4.13) as

$$\begin{aligned}
L(X_k) =& L_k - L(X_k, X_{k-1}) \\
=& \frac{1}{2}(X_k - f(\hat{X}_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}))^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} \\
& (X_k - f(\hat{X}_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) \\
& + \frac{1}{2}(X_{k-1} - \hat{X}_{k-1})^T {P_{k-1}}^{-1} (X_{k-1} - \hat{X}_{k-1}) \\
& - \frac{1}{2}(X_{k-1} - \Psi_k {F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} \\
& (X_k - f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + {P_{k-1}}^{-1} \hat{X}_{k-1}))^T \\
& \Psi_k^{-1} (X_{k-1} - \Psi_k {F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} \\
& (X_k - f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}) + {P_{k-1}}^{-1} \hat{X}_{k-1}))
\end{aligned}$$

After replacing the value of Ψ_k and a tedious simplification, all terms containing X_{k-1} cancel out. We get our required function as

$$\begin{aligned}
L(X_k) =& \frac{1}{2} {X_k}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} X_k - {X_k}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} f(\hat{X}_{k-1}, \mu_k, 0) \\
& + {X_k}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} \hat{X}_{k-1} \\
& + \frac{1}{2} f(\hat{X}_{k-1}, \mu_k, 0)^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} f(\hat{X}_{k-1}, \mu_k, 0) + \frac{1}{2} \hat{X}_{k-1}{}^T {P_{k-1}}^{-1} \hat{X}_{k-1} \\
& + f(\hat{X}_{k-1}, \mu_k, 0)^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} \hat{X}_{k-1} \\
& + \frac{1}{2} \hat{X}_{k-1}{}^T {F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} \hat{X}_{k-1} \\
& - \frac{1}{2} ({F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} (X_k - f(X_{k-1}, \mu_k, 0) + F_{X_{k-1}} \hat{X}_{k-1}) + {P_{k-1}}^{-1} \hat{X}_{k-1})^T \\
& (F_{X_{k-1}}{}^T (F_{X_{k-1}} Q_k {F_{X_{k-1}}}^T)^{-1} F_{X_{k-1}} + {P_{k-1}}^{-1})^{-1} \\
& ({F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} (X_k - f(X_{k-1}, \mu_k, 0) + F_{X_{k-1}} \hat{X}_{k-1}) + {P_{k-1}}^{-1} \hat{X}_{k-1})
\end{aligned} \tag{4.14}$$

Note that $L(X_k)$ is quadratic in X_k. This means that $\overline{bel(X_k)}$ is indeed normally distributed. The mean and the covariance of this distribution are the minimum and

curvature of $L(X_k)$, which is obtained by computing the first and second derivatives of (4.14) with respect to X_k

$$\begin{aligned}\frac{\partial L(X_k)}{\partial(X_k)} =&(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}X_k - (F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}f(\hat{X}_{k-1},\mu_k,0)\\ &+(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_k-1}\hat{X}_{k-1}\\ &-(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_k-1}({F_{X_{k-1}}}^T(F_{X_{k-1}}Q_k{F_{X_{k-1}}}^T)^{-1}F_{X_{k-1}}+{P_{k-1}}^{-1})^{-1}\\ &\quad({F_{X_{k-1}}}^T(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}(X_k - f(X_{k-1},\mu_k,0)+F_{X_{k-1}}\hat{X}_{k-1})+{P_{k-1}}^{-1}\hat{X}_{k-1})\end{aligned}$$

After simplification we get

$$\begin{aligned}\frac{\partial L(X_k)}{\partial(X_k)} =&\Big((F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}\\ &-(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_{k-1}}({F_{X_{k-1}}}^T(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_{k-1}}+{P_{k-1}}^{-1})^{-1}\\ &\quad{F_{X_{k-1}}}^T(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}\Big)\Big(X_k - f(X_{k-1},\mu_k,0)+F_{X_{k-1}}\hat{X}_{k-1}\Big)\\ &-\Big((F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_{k-1}}({F_{X_{k-1}}}^T(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_{k-1}}+{P_{k-1}}^{-1})^{-1}\Big)\\ &\quad{P_{k-1}}^{-1}\hat{X}_{k-1}\end{aligned} \tag{4.15}$$

The *inversion lemma* states that for any invertible quadratic matrices R and Q and any matrix P of appropriate dimensions, the following holds true

$$(R+PQP^T)^{-1} = R^{-1} - R^{-1}P(Q^{-1}+P^TR^{-1}P)^{-1}P^TR^{-1} \tag{4.16}$$

Applying this statement to the first factor in (4.15), our derivative can now be expressed as

$$\begin{aligned}\frac{\partial L(X_k)}{\partial(X_k)} =&\Big((F_{\omega_k}Q_k{F_{\omega_k}}^T)+F_{X_{k-1}}P_{k-1}{F_{X_{k-1}}}^T\Big)\\ &\quad\Big(X_k - f(X_{k-1},\mu_k,0)+F_{X_{k-1}}\hat{X}_{k-1}\Big)\\ &-\Big((F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_{k-1}}({F_{X_{k-1}}}^T(F_{\omega_k}Q_k{F_{\omega_k}}^T)^{-1}F_{X_{k-1}}+{P_{k-1}}^{-1})^{-1}\Big)\\ &\quad{P_{k-1}}^{-1}\hat{X}_{k-1}\end{aligned}$$

The minimum of $L(X_k)$ is obtained when its first derivative is put equal to zero. Thus we have

$$\begin{aligned}
&\Big((F_{\omega_k} Q_k {F_{\omega_k}}^T) + F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T\Big)^{-1} \Big(X_k - f(X_{k-1}, \mu_k, 0) + F_{X_{k-1}} \hat{X}_{k-1}\Big) \\
&= \Big((F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} ({F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} + {P_{k-1}}^{-1})^{-1}\Big) {P_{k-1}}^{-1} \hat{X}_{k-1}
\end{aligned}$$

Solving this for the target variable X_k gives us

$$\begin{aligned}
X_k =& f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}} \hat{X}_{k-1} \\
&+ \Big((F_{\omega_k} Q_k {F_{\omega_k}}^T) + F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T\Big) \\
&\quad \Big((F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} ({F_{X_{k-1}}}^T (F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} F_{X_{k-1}} + {P_{k-1}}^{-1})^{-1}\Big) \\
&\quad {P_{k-1}}^{-1} \hat{X}_{k-1}
\end{aligned}$$

Multiplying and solving gives us the surprisingly compact result

$$X_k = f(X_{k-1}, \mu_k, 0) - F_{X_{k-1}} \hat{X}_{k-1} + F_{X_{k-1}} \hat{X}_{k-1}$$

Hence, in the recursive notation we have,

$$\boxed{\hat{X}_{k|k-1} = f(X_{k-1}, \mu_k, 0)} \tag{4.17}$$

Which proves the correctness of (4.1). This is the mean of $\overline{bel(X_k)}$. To find the variance, we proceed by calculating the curvature of $L(X_k)$, which is given by

$$\frac{\partial^2 L(X_k)}{\partial {X_k}^2} = \Big((F_{\omega_k} Q_k {F_{\omega_k}}^T) + F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T\Big)^{-1}$$

This is the curvature of the quadratic function $L(X_k)$, whose inverse is the covariance of the belief $\overline{bel(X_k)}$. Hence,

$$\boxed{P_{k|k-1} = (F_{\omega_k} Q_k {F_{\omega_k}}^T) + F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T} \tag{4.18}$$

This proves the correctness of (4.2).

4.3.2 Part 2: Measurement Covariance

We start with step 2 of the extended Bayes filter given by (4.8). The distribution of the measurement is calculated as follows

$$p(Y_k|X_k, Y_{0:k-1}, \mu_k) = \int T \mathrm{d}X_{k-1} \tag{4.19}$$

where

$$T = \underbrace{p(Y_k|X_k, Y_{0:k-1}, \mu_k, X_{k-1})}_{(\sim N(Y_K; h(\hat{X}_k, \hat{X}_{k-1}) + H_{x_k}(X_k - \hat{X}_k) + H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1}); R_k))}$$

$$\underbrace{p(X_{k-1}|X_k, Y_{0:k-1}, \mu_k)}_{(\sim N(X_{k-1}; g(\hat{x}_k, \mu_k, 0) + G_{X_k}(X_k - \hat{X}_k); G_{\omega_k} Q_k {G_{\omega_k}}^T))}$$

The derivation is exactly the same as that of step 1. In this case the integral can be expressed as

$$\int \exp\{-M_k\} \mathrm{d}X_{k-1}$$

Where

$$\begin{aligned} M_k = \frac{1}{2}\Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) - H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})\Big)^T {R_k}^{-1} \\ \Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) - H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})\Big) \\ + \frac{1}{2}\Big(X_{k-1} - g(\hat{x}_k, \mu_k, 0) - G_{X_k}(X_k - \hat{X}_k)\Big)^T \Big(G_{\omega_k} Q_k {G_{\omega_k}}^T\Big)^{-1} \\ \Big(X_{k-1} - g(\hat{x}_k, \mu_k, 0) - G_{X_k}(X_k - \hat{X}_k)\Big) \end{aligned} \tag{4.20}$$

As in the previous section, we decompose M_k into two functions. These functions are given by

$$M_k = M(Y_k, X_k, X_{k-1}) + M(Y_k, X_k) \tag{4.21}$$

By differentiating twice with respect to X_{k-1}, we determine the minimum and covariance of M_k. Thus we get

$$\begin{aligned}
&M(Y_k, X_k, Xk-1) \\
&= \frac{1}{2}\Big(X_{k-1} - \Phi_k\big({H_{X_{k-1}}}^T {R_k}^{-1}(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) \\
&\quad - H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) + (G_{\omega_k} Q_k {G_{\omega_k}}^T)^{-1}(g(\hat{x}_k, \mu_k, 0) - G_{X_k}(X_k - \hat{X}_k))\big)\Big)^T \\
&\quad \Phi_k^{-1}\Big(X_{k-1} - \Phi_k\big({H_{X_{k-1}}}^T {R_k}^{-1}(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) \\
&\quad - H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) + (G_{\omega_k} Q_k {G_{\omega_k}}^T)^{-1}(g(\hat{x}_k, \mu_k, 0) - G_{X_k}(X_k - \hat{X}_k))\big)\Big)
\end{aligned} \tag{4.22}$$

Where

$$\Phi_k = \Big({H_{X_{k-1}}}^T {R_k}^{-1} H_{X_{k-1}} + (G_{\omega_k} Q_k {G_{\omega_k}}^T)^{-1}\Big)^{-1}$$

Hence, from (4.20), (4.21) and (4.22), we have,

$$
\begin{aligned}
M(Y_k, X_k) = \frac{1}{2}\Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) - H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})\Big)^T {R_k}^{-1} \\
\Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) - H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})\Big) \\
+\frac{1}{2}\Big(X_{k-1} - g(\hat{x}_k, \mu_k, 0) - G_{X_k}(X_k - \hat{X}_k)\Big)^T \Big(G_{\omega_k} Q_k {G_{\omega_k}}^T\Big)^{-1} \\
\Big(X_{k-1} - g(\hat{x}_k, \mu_k, 0) - G_{X_k}(X_k - \hat{X}_k)\Big) \\
-\frac{1}{2}\Big(X_{k-1} - \Phi_k\big({H_{X_{k-1}}}^T {R_k}^{-1}(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) \\
- H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) + (G_{\omega_k} Q_k {G_{\omega_k}}^T)^{-1}(g(\hat{x}_k, \mu_k, 0) \\
- G_{X_k}(X_k - \hat{X}_k))\big)\Big)^T \Phi_k^{-1} \\
\Big(X_{k-1} - \Phi_k\big({H_{X_{k-1}}}^T {R_k}^{-1}(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{x_k}(X_k - \hat{X}_k) \\
- H_{X_{k-1}}(X_{k-1} - \hat{X}_{k-1})) + (G_{\omega_k} Q_k {G_{\omega_k}}^T)^{-1}(g(\hat{x}_k, \mu_k, 0) \\
- G_{X_k}(X_k - \hat{X}_k))\big)\Big)
\end{aligned}
\tag{4.23}
$$

This is the required distribution. The mean and covariance of this distribution is given by the minimum and inverse of the curvature of $M(Y_k, X_k)$ respectively.Differentiating with respect to Y_k, we get the mean as

$$h(\hat{X}_k, \hat{X}_{k-1}) + H_{X_k}(X_k - \hat{X}_k)$$

And the covariance, the inverse of the curvature, given as

$$\boxed{W_k = R_k + H_{X_{k-1}} G_{\omega_k} Q_k {G_{\omega_k}}^T {H_{X_{k-1}}}^T} \tag{4.24}$$

This proves the correctness of (4.3).

4.3.3 Part3: Correction / Update

We begin with step 3 of the extended Bayes filter

$$bel(X_k) = \eta \underbrace{p(Y_k|X_k, Y_{0:k-1}, \mu_k)}_{\left(\sim N(Y_k; h(\hat{X}_k, \hat{X}_{k-1}) + H_{X_{k-1}}(X_k - \hat{X}_k))\right)} \underbrace{p(X_k|Y_{0:k-1}, \mu_k)}_{\left(\sim N(X_k; f(\hat{X}_{k-1}, \mu_k, 0); F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T)\right)}$$

As easily observed, the product is given by the exponential

$$bel(X_k) = \eta \exp\{-J_k\}$$

where

$$\begin{aligned} J_k = & \frac{1}{2}\Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{X_k}(X_k - \hat{X}_k)\Big)^T \Big(R_k + H_{X_{k-1}}(G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T\Big)^{-1} \\ & \Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{X_k}(X_k - \hat{X}_k)\Big) \\ & + \frac{1}{2}\Big(X_k - f(\hat{X}_{k-1}, \mu_k, 0)\Big)^T \\ & \Big(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T\Big)^{-1} \Big(X_k - f(\hat{X}_{k-1}, \mu_k, 0)\Big) \end{aligned}$$

This function is quadratic in X_k and hence it is a Gaussian. To find the parameters of this Gaussian, we take the first two derivatives of this function with respect to X_k

$$\begin{aligned} \frac{\partial J_k}{\partial X_k} = & -{H_{X_k}}^T \Big(R_k + H_{X_{k-1}}(G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T\Big)^{-1} \\ & \Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) - H_{X_k}(X_k - \hat{X}_k)\Big) \\ & + \Big(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T\Big)^{-1} \Big(X_k - f(\hat{X}_{k-1}, \mu_k, 0)\Big) \end{aligned} \tag{4.25}$$

and

$$\begin{aligned} \frac{\partial^2 J_k}{\partial {X_k}^2} = & {H_{X_k}}^T \Big(R_k + H_{X_{k-1}}(G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T\Big)^{-1} H_{X_k} \\ & + \Big(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T\Big)^{-1} \\ = & {P_{k|k}}^{-1} \end{aligned} \tag{4.26}$$

Putting the first derivative equal to zero and solving for X_k, we get the mean

$$\hat{X}_{k|k} = \hat{X}_{k|k-1} + P_{k|k} {H_{X_k}}^T \Big(R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T \Big)^{-1} \Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) \Big) \tag{4.27}$$

Define the Kalman Gain as

$$K_k = P_{k|k} {H_{X_k}}^T \Big(R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T \Big)^{-1}$$

Hence from (4.27), we have

$$\boxed{\hat{X}_{k|k} = \hat{X}_{k|k-1} + K_k \Big(Y_k - h(\hat{X}_k, \hat{X}_{k-1}) \Big)} \tag{4.28}$$

This proves the correctness of (4.5).

The Kalman Gain as given above, is a function of $P_{k|k}$ which is at odds with the fact that we utilize K_k to calculate $P_{k|k}$ in (4.6). So we need to transform our current expression of K_k, to include covariances other than $P_{k|k}$. This transformation is carried out as follows.

$$K_k = P_{k|k} {H_{X_k}}^T \Big(R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T \Big)^{-1}$$

$$\begin{aligned} K_k = & P_{k|k} {H_{X_k}}^T \Big(R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T \Big)^{-1} \\ & \Big(H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T + \\ & (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T) \Big) \\ & \Big(H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T + \\ & (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T) \Big)^{-1} \end{aligned}$$

$$
\begin{aligned}
K_k =& P_{k|k} \\
& \Big({H_{X_k}}^T (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T)^{-1} \\
& H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T + {H_{X_k}}^T \Big) \\
& \Big(H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T + (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T) \Big)^{-1}
\end{aligned}
$$

$$
\begin{aligned}
K_k =& P_{k|k} \\
& \Big({H_{X_k}}^T (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T)^{-1} H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T \\
& + \underbrace{(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T)(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1}}_{I} {H_{X_k}}^T \Big) \\
& \Big(H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T + (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T) \Big)^{-1}
\end{aligned}
$$

$$
\begin{aligned}
K_k =& P_{k|k} \\
& \underbrace{\Big({H_{X_k}}^T (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T)^{-1} H_{X_k} + (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1} \Big)}_{{P_{k|k}}^{-1}} \\
& \Big(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T \Big) {H_{X_k}}^T \\
& \Big(H_{X_k} (F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T) {H_{X_k}}^T + (R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T) \Big)^{-1}
\end{aligned}
$$

$$
\begin{aligned}
K_k =& \underbrace{\Big(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T \Big)}_{P_{k|k-1}} {H_{X_k}}^T \\
& \Big(H_{X_k} \underbrace{(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T)}_{P_{k|k-1}} {H_{X_k}}^T + \underbrace{(R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T)}_{W_k} \Big)^{-1}
\end{aligned}
$$

Hence we have

$$
\boxed{K_k = P_{k|k-1} {H_{X_k}}^T (H_{X_k} P_{k|k-1} {H_{X_k}}^T + W_k)^{-1}} \tag{4.29}
$$

This proves the correctness of (4.4)
The expression (4.26) for $P_{k|k}$ that we have calculated involves an inversion which is a heavy computation for high dimensional state spaces. So we carry out a transformation to express $P_{k|k}$ in a more convenient way

$$
P_{k|k} = \Big({H_{X_k}}^T \underbrace{(R_k + H_{X_{k-1}} (G_{\omega_k} Q_k {G_{\omega_k}}^T) {H_{X_{k-1}}}^T)^{-1}}_{{W_k}^{-1}} H_{X_k} + \underbrace{(F_{X_{k-1}} P_{k-1} {F_{X_{k-1}}}^T + F_{\omega_k} Q_k {F_{\omega_k}}^T)^{-1}}_{{P_{k|k-1}}^{-1}} \Big)^{-1}
$$

Thus

$$P_{k|k} = (H_{X_k}{}^T W_k{}^{-1} H_{X_k} + P_{k|k-1}{}^{-1})^{-1}$$

Applying the *inversion lemma* given in (4.16) to this expression, we have

$$P_{k|k} = P_{k|k-1} - P_{k|k-1} H_{X_k}{}^T (W_k + H_{X_k} P_{k|k-1} H_{X_k}{}^T)^{-1} H_{X_k} P_{k|k-1}$$

$$P_{k|k} = \big(I - \underbrace{P_{k|k-1} H_{X_k}{}^T (W_k + H_{X_k} P_{k|k-1} H_{X_k}{}^T)^{-1}}_{K_k} H_{X_k}\big) P_{k|k-1}$$

Hence we have,

$$\boxed{P_{k|k} = (I - K_k H_{X_k}) P_{k|k-1}}$$

This proves the correctness of (4.6).

Chapter 5

Setting up the EKF: calculations, simulation and results

This chapter discusses the calculations and implementation of the EKF for full planar monocular SLAM. In the end, the results of a simulation are shown and discussed.

5.1 Mathematical models

This section includes a derivation of the mathematical models and jacobian matrices as included in the Kalman filter equations.

5.1.1 The constant velocity model

The process model assumed for the system is the constant velocity motion model. An in depth treatment can be found in standard state estimation textbooks [2, 16] and only the final equations are represented here. The constant velocity model for a 2 dimensional state vector is given as

$$\begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ \dot{x}_{k-1} \end{bmatrix} + w_k$$

where w_k is the process noise vector with zero mean and covariance Q given by

$$\bar{Q} = q \begin{bmatrix} \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix}$$

where q is the covariance of noise in velocity in the continuous time domain. Assume the following state vector representing the pose of the robot and the location of a single landmark

$$X_k = \begin{bmatrix} {}^c x_k & {}^c x'_k & {}^c y_k & {}^c y'_k & {}^c \theta_k & {}^c \theta'_k & {}^{L2} x_k & {}^{L2} y_k \end{bmatrix}^T$$

Let

$$\bar{F} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}$$

Then the overall state transition matrix for the system is given by

$$F = \begin{bmatrix} \bar{F} & 0 & 0 & 0 \\ 0 & \bar{F} & 0 & 0 \\ 0 & 0 & \bar{F} & 0 \\ 0 & 0 & 0 & \mathbb{I} \end{bmatrix}$$

where $\mathbb{I}$ is the 2×2 identity matrix. The covariance of the process noise will be given as

$$Q = \begin{bmatrix} q_v \bar{Q} & 0 & 0 & 0 \\ 0 & q_v \bar{Q} & 0 & 0 \\ 0 & 0 & q_\theta \bar{Q} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Where q_v and q_θ are the variances of the noise in linear and angular velocities in the continuous time domain. For practical reasons, it is advisable to inject some

artificial process noise even for stationary states. Therefore the noise covariance matrix actually used for the simulation is

$$Q = \begin{bmatrix} q_v\bar{Q} & 0 & 0 & 0 \\ 0 & q_v\bar{Q} & 0 & 0 \\ 0 & 0 & q_\theta\bar{Q} & 0 \\ 0 & 0 & 0 & q_l\mathbb{I} \end{bmatrix}$$

where $\mathbb{I}$ is the 2×2 identity matrix and q_l is covariance of the process noise for the landmark. The value of this covariance is very small as compared to q_v and q_θ.

5.1.2 Model for projective measurements

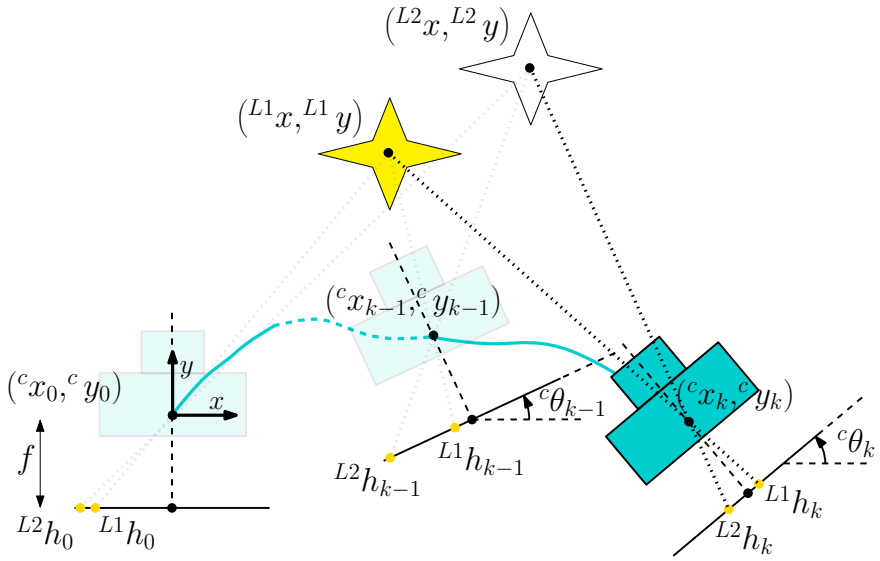

Figure 5-1: Monocular state sequence in 2D

Consider Figure 5-1, in which a camera performs SLAM in a 2D world. The state we take is given by

$$X_k = \begin{bmatrix} {}^{c}x_k & {}^{c}x'_k & {}^{c}y_k & {}^{c}y'_k & {}^{c}\theta_k & {}^{c}\theta'_k & {}^{L2}x_k & {}^{L2}y_k \end{bmatrix}^T$$

the measurement equation requires ${}^{L1}x$, ${}^{L1}y$, ${}^{L2}x$ and ${}^{L2}y$ to be described in the

camera coordinate frame. So we must first get the transformation matrix from the world frame to the camera frame.

The transformation matrix that describes the pose of the camera in the world frame is given as

$$ {}^{W}_{C}T = \begin{bmatrix} \cos({}^{c}\theta) & -\sin({}^{c}\theta) & {}^{c}x \\ \sin({}^{c}\theta) & \cos({}^{c}\theta) & {}^{c}y \\ 0 & 0 & 1 \end{bmatrix} $$

Our required transformation is given by the inverse of ${}^{W}_{C}T$ as

$$ {}^{C}_{W}T = {}^{W}_{C}T^{-1} = \begin{bmatrix} \cos({}^{c}\theta) & \sin({}^{c}\theta) & -{}^{c}x\cos({}^{c}\theta) - {}^{c}y\sin({}^{c}\theta) \\ -\sin({}^{c}\theta) & \cos({}^{c}\theta) & {}^{c}x\sin({}^{c}\theta) - {}^{c}y\cos({}^{c}\theta) \\ 0 & 0 & 1 \end{bmatrix} $$

and so the description of the landmark in the camera coordinate frame will be given by

$$ {}^{C}L1 = {}^{C}_{W}T\,{}^{W}L1 = \begin{bmatrix} {}^{C}{}_{L1}x \\ {}^{C}{}_{L1}y \end{bmatrix} = \begin{bmatrix} ({}^{L1}x - {}^{c}x)\cos({}^{c}\theta) + ({}^{L1}y - {}^{c}y)\sin({}^{c}\theta) \\ -({}^{L1}x - {}^{c}x)\sin({}^{c}\theta) + ({}^{L1}y - {}^{c}y)\cos({}^{c}\theta) \end{bmatrix} $$

the measurement model for the camera is given by

$$ {}^{L1}h = f\frac{{}^{C}{}_{L1}x}{{}^{C}{}_{L1}y} = f\frac{({}^{L1}x - {}^{c}x)\cos({}^{c}\theta) + ({}^{L1}y - {}^{c}y)\sin({}^{c}\theta)}{-({}^{L1}x - {}^{c}x)\sin({}^{c}\theta) + ({}^{L1}y - {}^{c}y)\cos({}^{c}\theta)} $$

$$ {}^{L2}h = f\frac{{}^{C}{}_{L2}x}{{}^{C}{}_{L2}y} = f\frac{({}^{L2}x - {}^{c}x)\cos({}^{c}\theta) + ({}^{L2}y - {}^{c}y)\sin({}^{c}\theta)}{-({}^{L2}x - {}^{c}x)\sin({}^{c}\theta) + ({}^{L2}y - {}^{c}y)\cos({}^{c}\theta)} $$

We can find the Jacobian matrices for both measurements. They will be given as

$$
{}^{L1}H = \frac{1}{\Omega(L1)^2}
\begin{bmatrix}
-f({}^{L1}y - {}^{c}y) \\
0 \\
f({}^{L1}x - {}^{c}x) \\
0 \\
f\left(({}^{L1}x - {}^{c}x)^2 + ({}^{L1}y - {}^{c}y)^2\right) \\
0 \\
0 \\
0
\end{bmatrix}^T
$$

and

$$
{}^{L2}H = \frac{1}{\Omega(L2)^2}
\begin{bmatrix}
-f({}^{L2}y - {}^{c}y) \\
0 \\
f({}^{L2}x - {}^{c}x) \\
0 \\
f\left(({}^{L2}x - {}^{c}x)^2 + ({}^{L2}y - {}^{c}y)^2\right) \\
0 \\
f({}^{L2}y - {}^{c}y) \\
-f({}^{L2}x - {}^{c}x)
\end{bmatrix}^T
$$

where

$$
\Omega(Li) = -({}^{Li}x - {}^{c}x)\sin({}^{c}\theta) + ({}^{Li}y - {}^{c}y)\cos({}^{c}\theta)
$$

5.1.3 Model for disparity measurements

Consider Figure 5-2 which depicts the disparity mesurement for a camera performing motion stereo between two consecutive time steps. The perpendicular distance of the

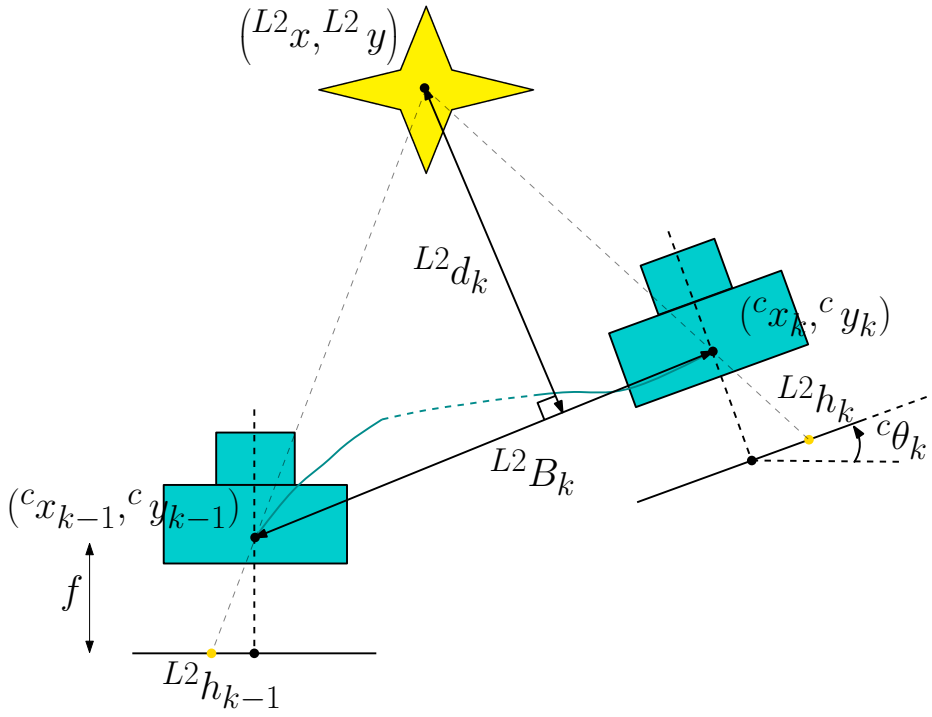

Figure 5-2: The disparity measurement

landmark from the baseline ${}^{L2}B_k$ is given by

$${}^{L2}d_k = \frac{({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})}{\sqrt{({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2}}$$

and the magnitude of the baseline is given by

$${}^{L2}B_k = \sqrt{({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2}$$

disparity is given by

$${}^{L2}\text{disp}_k = f\frac{{}^{L2}B_k}{{}^{L2}d_k}$$

and so we get

$${}^{L2}\text{disp}_k = \frac{f\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)}{({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})}$$

the Jacobian can be found as

$$
{}^{L2}H = f\begin{bmatrix}
\frac{2({}^{c}x_k - {}^{c}x_{k-1})}{({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})} - \frac{\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)\left({}^{c}y_{k-1} - {}^{L2}y_k\right)}{\left(({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})\right)^2} \\
0 \\
\frac{2({}^{c}y_k - {}^{c}y_{k-1})}{({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})} + \frac{\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)\left({}^{c}x_{k-1} - {}^{L2}x_k\right)}{\left(({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})\right)^2} \\
0 \\
0 \\
0 \\
- \frac{\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)({}^{c}y_k - {}^{c}y_{k-1})}{\left(({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})\right)^2} \\
\frac{\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)({}^{c}x_k - {}^{c}x_{k-1})}{\left(({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L2}y) - ({}^{c}x_{k-1} - {}^{L2}x)({}^{c}y_k - {}^{c}y_{k-1})\right)^2}
\end{bmatrix}
$$

Similarly, the Jacobian for the disparity measurement of a beacon, say $L1$ will be given as

$$
{}^{L1}H = f\begin{bmatrix}
\frac{2({}^{c}x_k - {}^{c}x_{k-1})}{({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L1}y) - ({}^{c}x_{k-1} - {}^{L1}x)({}^{c}y_k - {}^{c}y_{k-1})} - \frac{\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)\left({}^{c}y_{k-1} - {}^{L1}y_k\right)}{\left(({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L1}y) - ({}^{c}x_{k-1} - {}^{L1}x)({}^{c}y_k - {}^{c}y_{k-1})\right)^2} \\
0 \\
\frac{2({}^{c}y_k - {}^{c}y_{k-1})}{({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L1}y) - ({}^{c}x_{k-1} - {}^{L1}x)({}^{c}y_k - {}^{c}y_{k-1})} + \frac{\left(({}^{c}x_k - {}^{c}x_{k-1})^2 + ({}^{c}y_k - {}^{c}y_{k-1})^2\right)\left({}^{c}x_{k-1} - {}^{L1}x_k\right)}{\left(({}^{c}x_k - {}^{c}x_{k-1})({}^{c}y_{k-1} - {}^{L1}y) - ({}^{c}x_{k-1} - {}^{L1}x)({}^{c}y_k - {}^{c}y_{k-1})\right)^2} \\
0 \\
0 \\
0 \\
0 \\
0
\end{bmatrix}
$$

5.1.4 Simulating the disparity measurements

Although Section 5.1.3 gives the correct model to use in the Kalman filter equations, it does not represent the true process by which disparity is obtained practically and so, should not be used to simulate it. A detailed treatment can be found in any standard computer vision text like [24] and [21]. As an overview, disparity is obtained through the following steps

- transform the image planes from the case in Figure 5-1 so that they appear to

be taken from row aligned cameras as in Figure 3-5. Let

$$\alpha = \tan^{-1}\left(\frac{{}^{c}y_k - {}^{c}y_{k-1}}{{}^{c}y_k - {}^{c}y_{k-1}}\right)$$

- transform ${}^{Li}u_{k-1}$ to appear as if the camera at time step $k-1$ were rotated by $(\alpha - {}^{c}\theta_{k-1})$ to get ${}^{Li}\bar{u}_{k-1}$. Similarly transform ${}^{Li}u_k$ to appear as if the camera at time step k were rotated by $(\alpha - {}^{c}\theta_k)$ to get ${}^{Li}\bar{u}_k$. The rectified measurements turn out to be

$${}^{Li}\bar{u}_k = \frac{-f\,{}^{Li}u_k \cos(\alpha - {}^{c}\theta_k) + f^2 \sin(\alpha - {}^{c}\theta_k)}{{}^{Li}u_k \sin(\alpha - {}^{c}\theta_k) + f\cos(\alpha - {}^{c}\theta_k)}$$

and

$${}^{Li}\bar{u}_{k-1} = \frac{-f\,{}^{Li}u_{k-1} \cos(\alpha - {}^{c}\theta_{k-1}) + f^2 \sin(\alpha - {}^{c}\theta_{k-1})}{{}^{Li}u_{k-1} \sin(\alpha - {}^{c}\theta_{k-1}) + f\cos(\alpha - {}^{c}\theta_{k-1})}$$

- disparity is given by the difference between the rectified measurements.

$${}^{Li}\text{disp}_k = {}^{Li}\bar{u}_k - {}^{Li}\bar{u}_{k-1}$$

5.2 Simulation results

The EKF just set up was implemented in MATLAB for simulation. The results given over here are as a comparison between the performance of Monocular SLAM with and without including disparity measurements for 2 sample trajectories, a linear trajectory and a rhombus like trajectory.

5.2.1 Linear trajectory

Here results are presented for the robot moving along a linear trajectory in a 2D world, performing SLAM with 1 landmark and 2 beacons. Figure 5-3 show the true and estimated trajectories when only projective measurements are recorded. The

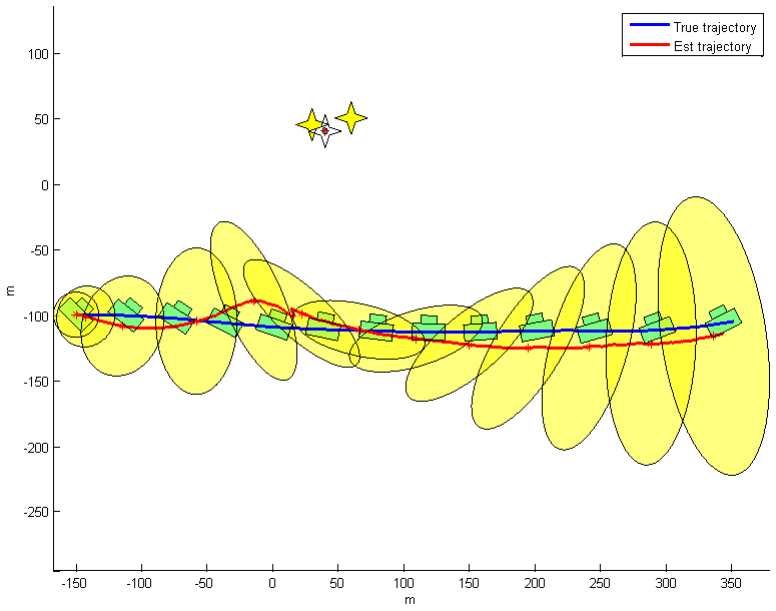

Figure 5-3: Estimated trajectory and 95% confidence ellipses for monocular SLAM (without disparity measurements) with 2 beacons. The beacons are highlighted in yellow and the landmark begin estimated is white.

focal length f has been kept at 20 pixel units with $q_v = 0.5(m/s^2)^2$, $q_\theta = 0.1\text{deg}^2$ and $q_l = 0.001m^2$. The measurement noise covariance has been kept at 0.074pixels^2 for projective and 74pixels^2 for disparity. The same parameters have been assumed for all simulations presented henceforth. Figure 5-4 shows the same for monocular SLAM after including disparity measurements.

It can be seen that there is a significant difference in the area covered by the uncertainty ellipses in both cases. This is reflected in the graphs shown in Figure5-5. Ham and Brown [8] show that the magnitude of the eigenvalues of the state covariance matrices can be used as a metric for the "degree" of observability with the smaller eigenvalues representing linear combinations of states in more observable directions whereas larger eigenvalues representing linear combination of states in less observable directions. Hence this property has been compared with the trace of the covariance matrix which is equal to the sum of the eigenvalues. Monocular SLAM with disparity and projective measurements gives better performance in this regard both in the

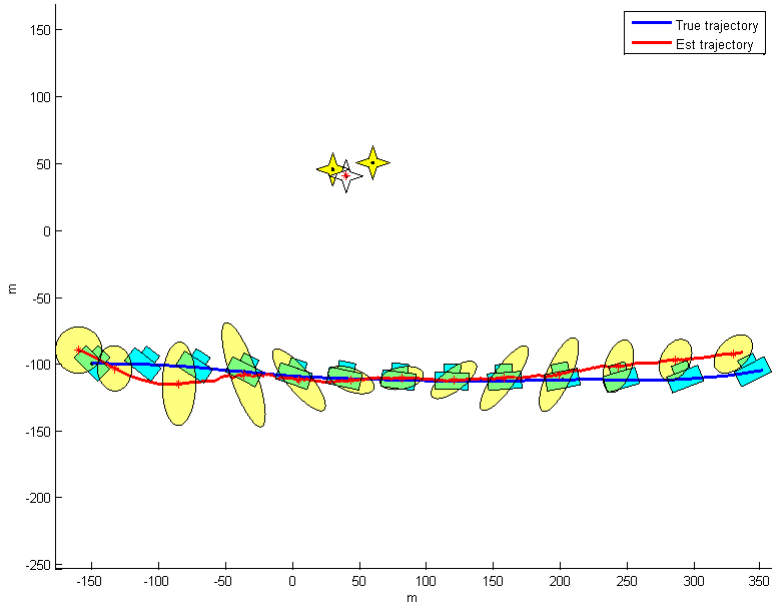

Figure 5-4: Estimated trajectory and 95% confidence ellipses for monocular SLAM (with disparity measurements) with 2 beacons.

map and robot pose variables with the improvement in the robot pose much more significant than the map.

5.2.2 Rhombus shaped trajectory

Figures 5-6 and 5-7 show the results of monocular SLAM with 3 beacons this time for a rhombus like trajectory. The findings are similar to that of the linear trajectory. The comparison given in Figure 5-8 reflects this.

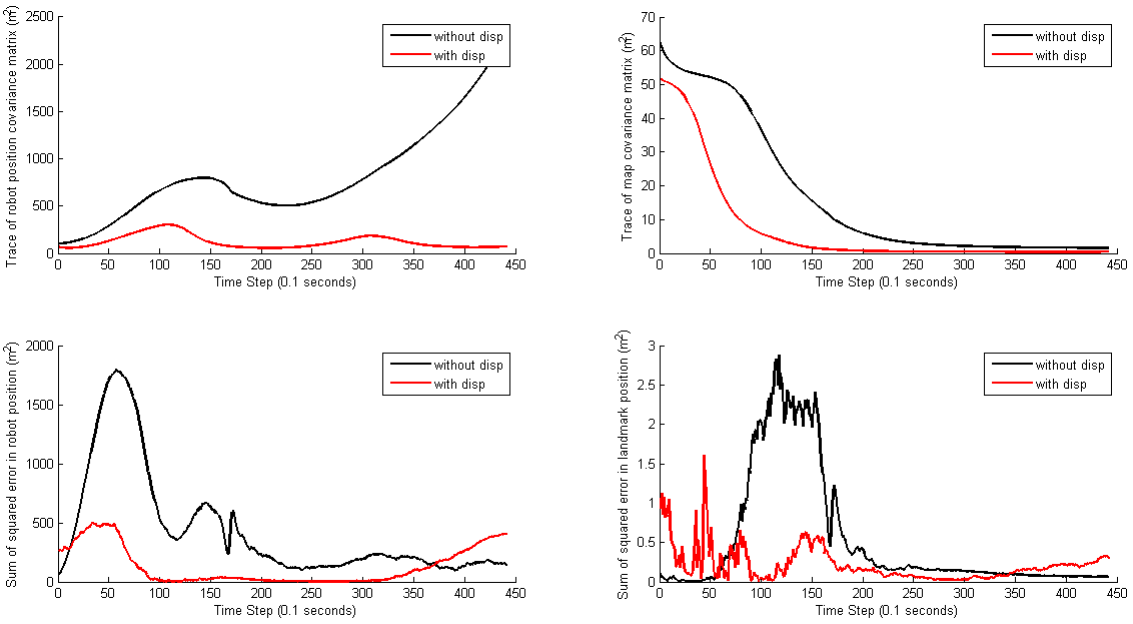

Figure 5-5: A comparison of the trace of the covariance matrices and SSD errors in the robot pose and landmark states for Figures 5-3 and 5-4.

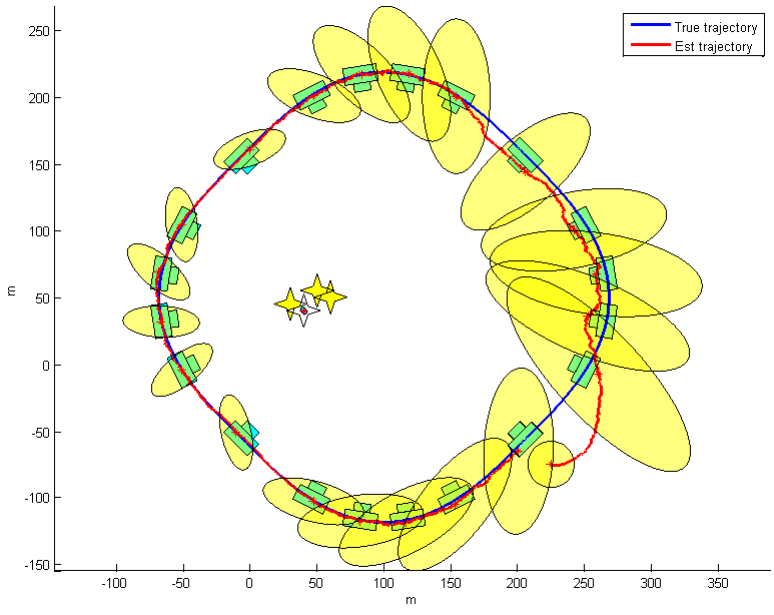

Figure 5-6: Estimated trajectory and 95% confidence ellipses for monocular SLAM (without disparity measurements) with 3 beacons.

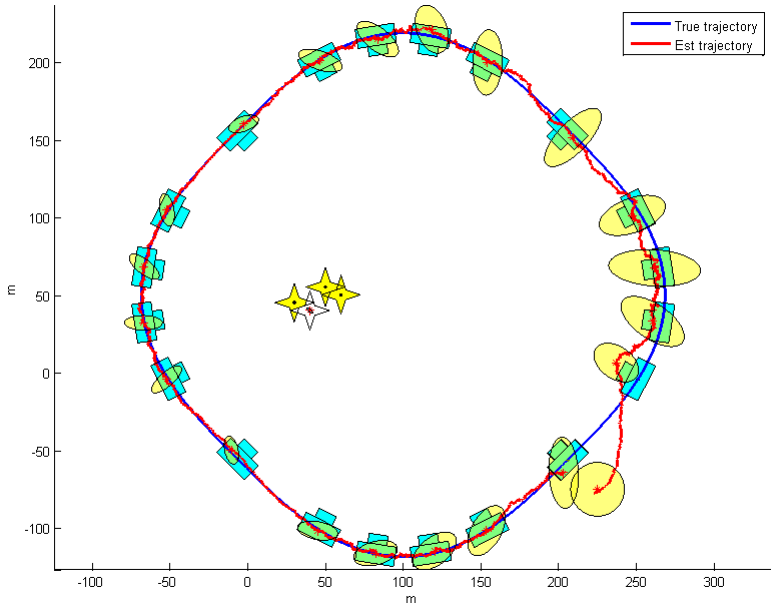

Figure 5-7: Estimated trajectory and 95% confidence ellipses for monocular SLAM (with disparity measurements) with 3 beacons.

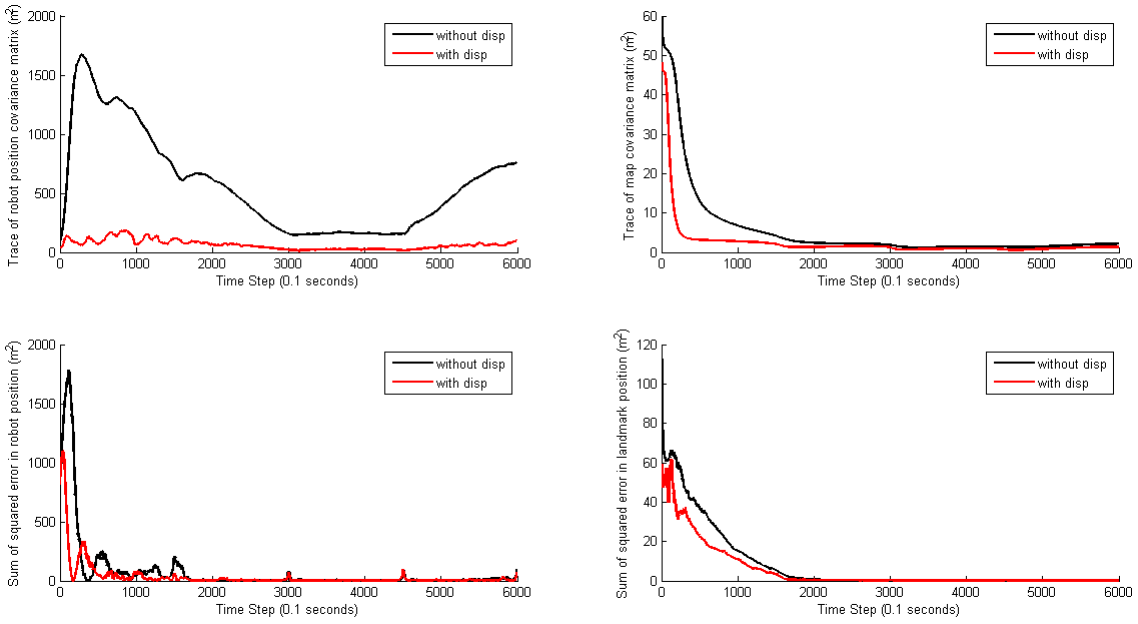

Figure 5-8: A comparison of the trace of the covariance matrices and SSD errors in the robot pose and landmark states for Figures 5-6 and 5-7.

Chapter 6

Future works

6.1 Dealing with beacon obscurity through Beaconization

The discussion over here is an extension of Case 3 from Section 3.4. Consider the possibility that after some time, the robot will lose sight of the known landmark (Figure 3-3), and hence the process will reduce to that of Case 2 (Figure 3-2) , in which the robot is observing only such landmarks which are unknown. In this case, to keep the process observable, The following measures can be taken

6.1.1 Procedure

1. We will always treat at least one landmark as the beacon. When the original beacon is obscured, one of the landmarks will be "beaconized". Apart from the primary beacon which is known perfectly, every other secondary beacon will have an associated covariance.

2. Only those landmarks qualify for beaconization whose state covariance has saturated i.e. it cannot be improved further. When the covariance reaches this value, the respective landmark will be excluded from the state and treated as a beacon in the measurement equation. The measurement model will be given

by Equation 3.2, which has been reproduced here

$$^{Li}h_k = {}^{Li}u_k - {}^{Li}xfd = -{}^{c}x_k fd$$

for the ith landmark. The covariance of ^{Li}x will be added to the covariance of $^{Li}u_k$ in accordance with the relation given above to form the overall covariance of the measurement $^{Li}h_k$. Hence the measurement covariance will increase, which is the price we have to pay to maintain observability.

3. Each time a beacon is obscured and a landmark is beaconized, the respective measurement noise covariance will be increased as a function of the covariance of ^{Li}x. We will talk about this increase shortly.

6.1.2 Finding the steady state covariance

The first step in the procedure just given would be to find the steady state covariance of the observable landmarks so that we know when the values of their respective covariances are saturated and hence qualify for beaconization. The steady state value of the state covariance can be found beforehand by solving the following discrete algebraic riccatti equation [16]

$$P_\infty = FP_\infty F^T - FP_\infty H^T \left(HP_\infty H^T + R\right)^{-1} HP_\infty F^T + Q \tag{6.1}$$

where F is the state transition matrix

H is the measurement matrix

R is the measurement noise covariance

Q is the process noise covariance

this value will remain true till R remains constant. When R changes in event of a new beacon, the steady state covariance will then have to be updated. Then from P_∞ we can extract the new steady state covariance of the landmark of interest.

6.1.3 Increasing the measurement noise covariance

Next we need to determine by how much to increase the measurement noise covariance on exclusion of a landmark from the state after its steady state covariance has been achieved. Consider the associated measurement equation before beaconization

$$^{Li}h_k = {}^{Li}u_k - {}^{Li}xfd = -{}^{c}x_k fd$$

when $\sigma^2_{Li_x}$ is zero (for a beacon), then $\sigma^2_{Li_{h_k}} = \sigma^2_{Li_{u_k}}$. This would have been the case if the landmark was perfectly known. For a landmark with an associated covariance, we would have a non zero $\sigma^2_{x_{Li}}$ and can be found out by basic probability theory [3] to be

$$\sigma^2_{Li_{h_k}} = \sigma^2_{Li_{u_k}} + f^2d^2\sigma^2_{Li_x} + 2fd\mathrm{Cov}(^{Li}u_k, {}^{Li}x)$$

hence the increase in the measurement noise covariance is given by the term

$$f^2d^2\sigma^2_{Li_x} + 2fd\mathrm{Cov}(^{Li}u_k, {}^{Li}x)$$

6.1.4 Revisiting a beacon with less covariance

The beaconization procedure implies that as more and more landmarks are observed and eventually excluded from the state (after the covariance saturates), the measurement noise covariance will increase continuously. Eventually, a point will be reached when it would not be practical to continue the process, as the measurement noise will affect the lower limit for the state error regardless of the observability.

One possible way to handle this would be to make the robot re-observe a less uncertain beacon every now and then, so as to calibrate itself before the measurement error grows out of bounds. This may be accomplished with an active vision setup. The period after which to re-observe that beacon would be a function of the accuracy of the estimate we require from our application.

We represent the required accuracy as an upper bound on P_∞. This in turn, will

specify an upper bound on the covariance matrix of the measurement noise, given by the following rearrangement of Equation 6.1

$$R = FP_{\infty}H^T \left(FP_{\infty}F^T - P_{\infty} + Q\right)^{-1} HP_{\infty}F^T - HP_{\infty}H^T \tag{6.2}$$

We can use this upper bound on R to determine when it is essential to return to re-observe a relatively certain beacon.

Note: Although the process of beaconization has been discussed here with regards to the monobot, it can be readily extended to higher dimensional cases.

6.2 Other future works

Full non linear observability analysis The observability analysis given in this thesis was able to reach only to the case of 1.5 D SLAM. A solution to the full 2D problem is a natural extension.

Real world experiments This study gave a comparison of monocular SLAM with and without disparity measurements in a simulated environment. Results from real world experiments would be required to fully asses the validity of conclusions drawn from the simulation.

Chapter 7

Conclusion

In this thesis, three different studies have been conducted. First of all, an attempt has been made to solve the observability of monocular SLAM in its full non linear form. The methodology adopted here is the direct calculation of the observability rank criterion which is too complex to solve analytically by hand, or by a computer (at least for now). So a bottom up approach has been used, starting from the simplest possible case of 1D localization, and gradually adding degrees of freedom to the problem. The direct calculation turns out to be computable till the case of 1.5D monocular SLAM in which a robot is constricted to move along a straight line in a 2D world. The rest of the unsolvable cases have been shown and posed as open problems for the interested reader. In the end, it was shown that including disparity in addition to projective measurements in the 1.5D case relaxes the observability rank criterion in terms of the minimum number of required beacons. This suggests that including motion stereo measurements in monocular SLAM may improve the overall performance in terms of observability.

Second of all, the use of motion stereo measurements has been considered for monocular SLAM. The dependence of the disparity measurement on the previous camera location has been incorporated into the EKF by re-deriving it from the basic Bayes filter. It was shown that this results in an increase in the measurement noise covariance. This increase depends on the process noise covariance and state dynamics.

Lastly, an EKF was setup for comparison of monocular SLAM in a 2D world with and without including disparity measurements. All calculations involved were shown and the results suggested that including disparity measurements in monocular SLAM does improve the performance in terms of observability (measured by the trace of the state covariance) especially in the robot pose variables.

Bibliography

[1] Milena Anguelova. *Observability and identifiability of nonlinear systems with applications in biology*. Chalmers University of Technology, 2007.

[2] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. Wiley-Interscience, 2004.

[3] Dimitri P Bertsekas and John N Tsitsiklis. *Introduction to probability*, volume 1. Athena Scientific Belmont, MA, 2002.

[4] Philippe Bonnifait and Gaëtan Garcia. Design and experimental validation of an odometric and goniometric localization system for outdoor robot vehicles. *Robotics and Automation, IEEE Transactions on*, 14(4):541–548, 1998.

[5] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1403–1410. IEEE, 2003.

[6] Gene F Franklin, J David Powell, and Abbas Emami-Naeini. Feedback control of dynamics systems. *Pretince Hall Inc*, 1986.

[7] DRORA Goshen-Meskin and IY Bar-Itzhack. Observability analysis of piece-wise constant systems. i. theory. *Aerospace and Electronic Systems, IEEE Transactions on*, 28(4):1056–1067, 1992.

[8] Fredric M Ham and R Grover Brown. Observability, eigenvalues, and kalman filtering. *Aerospace and Electronic Systems, IEEE Transactions on*, (2):269–273, 1983.

[9] Robert Hermann and Arthur Krener. Nonlinear controllability and observability. *Automatic Control, IEEE Transactions on*, 22(5):728–740, 1977.

[10] Jonghyuk Kim and Salah Sukkarieh. Improving the real-time efficiency of inertial slam and understanding its observability. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 1, pages 21–26. IEEE, 2004.

[11] Kwang Wee Lee, W Sardha Wijesoma, and Ibanez Guzman Javier. On the observability and observability analysis of slam. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 3569–3574. IEEE, 2006.

[12] Katsuhiko Ogata and Yanjuan Yang. Modern control engineering. 1970.

[13] Clark F Olson and Habib Abi-Rached. Wide-baseline stereo vision for terrain mapping. *Machine Vision and Applications*, 21(5):713–725, 2010.

[14] Lina M Paz, Pedro Piniés, Juan D Tardós, and José Neira. Large-scale 6-dof slam with stereo-in-hand. *Robotics, IEEE Transactions on*, 24(5):946–957, 2008.

[15] Konrad Reif, Stefan Gunther, Engin Yaz, and Rolf Unbehauen. Stochastic stability of the discrete-time extended kalman filter. *Automatic Control, IEEE Transactions on*, 44(4):714–728, 1999.

[16] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley-Interscience, 2006.

[17] Jean-Jacques E Slotine, Weiping Li, et al. *Applied nonlinear control*, volume 199. Prentice-Hall Englewood Cliffs, NJ, 1991.

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

[19] Joan Sola, André Monin, and Michel Devy. Bicamslam: Two times mono is more than stereo. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4795–4800. IEEE, 2007.

[20] RT Stefani, B Shahian, CJ Savant, and GH Hostetter. Design of feedback control systems, 2002.

[21] Richard Szeliski. *Computer vision: algorithms and applications*. Springer, 2010.

[22] Sebastian Thrun, Wolfram Burgard, Dieter Fox, et al. *Probabilistic robotics*, volume 1. MIT press Cambridge, MA, 2005.

[23] Michael J Tribou, David WL Wang, and William J Wilson. Observability of planar combined relative pose and target model estimation using monocular vision. In *American Control Conference (ACC), 2010*, pages 4546–4551. IEEE, 2010.

[24] Emanuele Trucco and Alessandro Verri. *Introductory techniques for 3-D computer vision*, volume 93. Prentice Hall Englewood Cliffs, 1998.

[25] Teresa Vidal-Calleja, Mitch Bryson, Salah Sukkarieh, Alberto Sanfeliu, and Juan Andrade-Cetto. On the observability of bearing-only slam. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 4114–4119. IEEE, 2007.

Appendix A

MATLAB: Process model

```
function [state jcbn] = process_model(X,time_step)
%PROCESS_MODEL constant velocity motion model
%[new_state jacobian] = process_model(state,time_step)implements the
%constant velocity dynamic model for the Kalman Filter.e motion model
n = size(X,1); %dimension of the state vector
state = zeros(n,1);
%apply constant velocity model to the robot states
state(1:6,:) = [ 1 time_step 0 0 0 0;
                 0 1 0 0 0 0;
                 0 0 1 time_step 0 0;
                 0 0 0 1 0 0;
                 0 0 0 0 1 time_step;
                 0 0 0 0 0 1 ] * X(1:6,:);
%the landmark states remain stationary
state(7:n,:) = X(7:n,:);
%jacobian
jcbn = eye(n);
jcbn(1,2) = time_step;
jcbn(3,4) = time_step;
jcbn(5,6) = time_step;
end
```

Appendix B

MATLAB: Measurement Model

```
function [Y jcbn jcbn_prev] = measurement_model(X,Xprev,B,f)
%MEASUREMENT_MODEL model for a single camera measurement

%dimension of the state vector
n = size(X,1);
%total number of beacons
b = size(B,1);
%total number of landmarks being estimated
l = (n - 3)/2;
%dimension of the measurement vector
m = 2*(l + b);

Y = zeros(m,1);
jcbn = zeros(m,n);
jcbn_prev = zeros(m,n);

%measurement of the landmarks
for i = 1:l
    %calculate indices of landmarks in state vector
    lx = 2*i + 2;
    ly = 2*i + 3;
    %bearing measurement
```

```
23      Y(2*i-1) = f*( (X(lx)-X(1))*cosd(X(3))+(X(ly)-X(2))*sind(X(3)) ...
            )/( -(X(lx)-X(1))*sind(X(3))+(X(ly)-X(2))*cosd(X(3)) );
24      %disparity measurement
25      Y(2*i) = f*( (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )/( ...
            (X(1)-Xprev(1))*(Xprev(2)-X(ly)) - ...
            (X(2)-Xprev(2))*(Xprev(1)-X(lx)) );
26  end
27  %measurement of the beacons
28  for i = 1:b
29      %bearing measurement
30      Y((2*i-1)+2*l) = f*( ...
            (B(i,1)-X(1))*cosd(X(3))+(B(i,2)-X(2))*sind(X(3)) )/( ...
            -(B(i,1)-X(1))*sind(X(3))+(B(i,2)-X(2))*cosd(X(3)) );
31      %disparity measurement
32      Y(2*i+2*l) = f*( (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )/( ...
            (X(1)-Xprev(1))*(Xprev(2)-B(i,2)) - ...
            (X(2)-Xprev(2))*(Xprev(1)-B(i,1)) );
33  end
34
35  %jacobian of the landmarks wrt the current state
36  for i = 1:l
37      %calculate indices of landmarks in state vector
38      lx = 2*i + 2;
39      ly = 2*i + 3;
40      %bearing measurement
41      den = ( -(X(lx)-X(1))*sind(X(3))+(X(ly)-X(2))*cosd(X(3)) )^2;
42      jcbn(2*i-1,1:3) = [-f*(X(ly)-X(2)) f*(X(lx)-X(1)) ...
            f*((X(lx)-X(1))^2+(X(ly)-X(2))^2)]/den;
43      jcbn(2*i-1,lx) = f*(X(ly)-X(2))/den;
44      jcbn(2*i-1,ly) = -f*(X(lx)-X(1))/den;
45      %disparity measurement
46      den =  (X(1)-Xprev(1))*(Xprev(2)-X(ly)) - ...
            (X(2)-Xprev(2))*(Xprev(1)-X(lx));
47      jcbn(2*i,1) = f*( 2*(X(1)-Xprev(1))/(den) - ( (X(1)-Xprev(1))^2 ...
            + (X(2)-Xprev(2))^2 )*( Xprev(2)-X(ly) )/( den^2 ) );
48      jcbn(2*i,2) = f*( 2*(X(2)-Xprev(2))/(den) + ( (X(1)-Xprev(1))^2 ...
```

```
        + (X(2)-Xprev(2))^2 )*( Xprev(1)-X(lx) )/( den^2 ) );
49      jcbn(2*i,lx) = -f*( ( (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( ...
        X(2)-Xprev(2) )/( den^2 ) );
50      jcbn(2*i,ly) = f*( ( (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( ...
        X(1)-Xprev(1) )/( den^2 ) );
51  end
52  %jacobian of the beacons wrt the current state
53  for i = 1:b
54      %bearing measurement
55      den = ( -(B(i,1)-X(1))*sind(X(3))+(B(i,2)-X(2))*cosd(X(3)) )^2;
56      jcbn((2*i-1)+2*l,1:3) = [-f*(B(i,2)-X(2)) f*(B(i,1)-X(1)) ...
        f*((B(i,1)-X(1))^2+(B(i,2)-X(2))^2)]/den;
57      %disparity measurement
58      den =  (X(1)-Xprev(1))*(Xprev(2)-B(i,2)) - ...
        (X(2)-Xprev(2))*(Xprev(1)-B(i,1));
59      jcbn(2*i+2*l,1) = f*( 2*(X(1)-Xprev(1))/(den) - ( ...
        (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( Xprev(2)-B(i,2) ...
        )/( den^2 ) );
60      jcbn(2*i+2*l,2) = f*( 2*(X(2)-Xprev(2))/(den) + ( ...
        (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( Xprev(1)-B(i,1) ...
        )/( den^2 ) );
61  end
62
63  %jacobian of the landmarks wrt the previous state
64  for i = 1:l
65      %calculate indices of landmarks in state vector
66      lx = 2*i + 2;
67      ly = 2*i + 3;
68      %bearing measurement - no dependence
69
70      %disparity measurement
71      den =  (X(1)-Xprev(1))*(Xprev(2)-X(ly)) - ...
        (X(2)-Xprev(2))*(Xprev(1)-X(lx));
72      jcbn_prev(2*i,1) = f*( -2*(X(1)-Xprev(1))/(den) - ( ...
        (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( X(ly)-X(2) )/( ...
        den^2 ) );
```

```
73      jcbn_prev(2*i,2) = f*( -2*(X(2)-Xprev(2))/(den) - ( ...
            (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( X(1)-X(lx) )/( ...
            den^2 ) );
74  end
75  %jacobian of the beacons wrt the prev state
76  for i = 1:b
77      %bearing measurement - no dependence
78
79      %disparity measurement
80      den =  (X(1)-Xprev(1))*(Xprev(2)-B(i,2)) - ...
            (X(2)-Xprev(2))*(Xprev(1)-B(i,1));
81      jcbn_prev(2*i+2*l,1) = f*( -2*(X(1)-Xprev(1))/(den) - ( ...
            (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( B(i,2)-X(2) )/( ...
            den^2 ) );
82      jcbn_prev(2*i+2*l,2) = f*( -2*(X(2)-Xprev(2))/(den) - ( ...
            (X(1)-Xprev(1))^2 + (X(2)-Xprev(2))^2 )*( X(1)-B(i,1) )/( ...
            den^2 ) );
83  end
84
85
86
87
88  end
```

Appendix C

MATLAB: Generating disparity

```
function [Y] = measurement_model_2(X,Xprev,Yprev,B,f)
%MEASUREMENT_MODEL model for a single camera measurement

%dimension of the state vector
n = size(X,1);
%total number of beacons
b = size(B,1);
%total number of landmarks being estimated
l = (n - 3)/2;
%dimension of the measurement vector
m = 2*(l + b);

Y = zeros(m,1);

%measurement of the landmarks
for i = 1:l
    %calculate indices of landmarks in state vector
    lx = 2*i + 2;
    ly = 2*i + 3;
    %bearing measurement
    Y(2*i-1) = f*( (X(lx)-X(1))*cosd(X(3))+(X(ly)-X(2))*sind(X(3)) ...
        )/( -(X(lx)-X(1))*sind(X(3))+(X(ly)-X(2))*cosd(X(3)) );
```

```
22      %disparity measurement
23      alpha = radtodeg(atan2( (X(2)-Xprev(2)),(X(1)-Xprev(1)) ));
24      Y(2*i) =  -  ( f*Y(2*i-1)*cosd(alpha-X(3)) - ...
            (f^2)*sind(alpha-X(3)) )/( Y(2*i-1)*sind(alpha-X(3)) + ...
            f*cosd(alpha-X(3)) ) ...
25                  +  ( f*Yprev(2*i-1)*cosd(alpha-Xprev(3)) - ...
                        (f^2)*sind(alpha-Xprev(3)) )/( ...
                        Yprev(2*i-1)*sind(alpha-Xprev(3)) + ...
                        f*cosd(alpha-Xprev(3)) );
26  end
27  %measurement of the beacons (no difference from landmarks case)
28  for i = 1:b
29      %bearing measurement
30      Y((2*i-1)+2*l) = f*( ...
            (B(i,1)-X(1))*cosd(X(3))+(B(i,2)-X(2))*sind(X(3)) )/( ...
            -(B(i,1)-X(1))*sind(X(3))+(B(i,2)-X(2))*cosd(X(3)) );
31      %disparity measurement
32      Y(2*i+2*l) =  - ( f*Y(2*i-1+2*l)*cosd(alpha-X(3)) - ...
            (f^2)*sind(alpha-X(3)) )/( Y(2*i-1+2*l)*sind(alpha-X(3)) + ...
            f*cosd(alpha-X(3)) ) ...
33                      + ( f*Yprev(2*i-1+2*l)*cosd(alpha-Xprev(3)) - ...
                            (f^2)*sind(alpha-Xprev(3)) )/( ...
                            Yprev(2*i-1+2*l)*sind(alpha-Xprev(3)) + ...
                            f*cosd(alpha-Xprev(3)) );
34  end
35
36
37  end
```

Appendix D

MATLAB: EKF Simulation

```
%EKF simulation
clear all;

load('scene1.mat'); %load landmarks and camera trajectory
l = 1; %number of landmarks to estimate
b = 2; %number of beacons
n = 2*l + 6; %dimension of the state vector
m = 2*(l + b); %dimension of the measurement vector
T = 0.1; %time interval between measurements in seconds
t = traj(2,4) - traj(1,4); %time step in given trajectory

X = traj(1:(T/t):size(traj,1),1:3)'; %the true state
it = size(X,2); %number of iterations
%append the landmark states
X_L = zeros(2*l,it);
for i = 1:l
    X_L(2*l-1,:) = ones(1,it)*landmarks(i,1);
    X_L(2*l,:) = ones(1,it)*landmarks(i,2);
end
X = [X;X_L];

%create matrices
```

```
23 K = zeros(n,m,it);  %Kalman gain
24 Xb = zeros(n,it);   %State estimate
25 Yn = zeros(m,it); %Measurement vector
26 Pb = zeros(n,n,it); %Estimate covariance
27 lambda = zeros(n, it); %State eigenvalues
28 B = landmarks(l+1:b+l,:); %Beacons
29 R = zeros(m,m); %Measurement noise covariance
30 F = zeros(n,n); %Process model jacobian
31 H = zeros(m,n); %Measurement model jacobian
32
33 stdm = 0.074;      %standard deviation of the camera resolution noise
34 f = 20;             %the focal length of the camera
35
36
37 %set noise covariance matrices
38 sigma_ax = 0.5;
39 sigma_ay = 0.5;
40 sigma_at = 0.1;
41 sigma_al = 0.001;
42 Q_x = [(T^3/3), (T^2/2);
43        (T^2/2), T]*sigma_ax;
44 Q_y = [(T^3/3), (T^2/2);
45        (T^2/2), T]*sigma_ay;
46 Q_t = [(T^3/3), (T^2/2);
47        (T^2/2), T]*sigma_at;
48 Q_l = eye(2*l)*0.001;
49 Q = [Q_x, zeros(2), zeros(2), zeros(2*l);
50      zeros(2), Q_y, zeros(2), zeros(2*l);
51      zeros(2), zeros(2), Q_t, zeros(2*l);
52      zeros(2), zeros(2), zeros(2), Q_l];
53
54  R = R + eye(m)*stdm;
55 for i = 2:2:m
56     R(i,i-1) = stdm*10;
57     R(i,i) = stdm*1000;
58     R(i-1,i) = stdm*10;
```

```
59 end
60
61
62
63 %initialize matrices
64
65 Xb(:,1) = [ -160;          %the initial estimate
66             1.2;
67             -90;
68             0;
69             -45;
70             1;
71             38;
72             42 ];
73
74 Pb(:,:,1) = [ 50 0 0 0 0 0 0 0;   %the initial state covariance
75               0 2.5 0 0 0 0 0 0;
76               0 0 50 0 0 0 0 0;
77               0 0 0 2.5 0 0 0 0;
78               0 0 0 0 20 0 0 0;
79               0 0 0 0 0 2.5 0 0;
80               0 0 0 0 0 0 50 0;
81               0 0 0 0 0 0 0 50 ];
82
83 %generate first measurement
84 [Yn(:,2), ~, ~] = measurement_model(X(:,2),X(:,1),B,f);
85 Yn(:,2) = Yn(:,2) + normrnd(zeros(m,1),stdm); %add noise
86 [Xb(:,2), ~] = process_model(Xb(:,1),T);
87 Pb(:,:,2) = Pb(:,:,1);
88
89 %start the filter
90 for k=3:it
91
92     %generate measurement vector
93     [Yn(:,k), ~, ~] = ...
           measurement_model_2(X(:,k),X(:,k-1),Yn(:,k-1),B,f);
```

```
94      Yn(:,k) = Yn(:,k) + normrnd(zeros(m,1),stdm); %add noise
95      %now estimate
96
97      [Xa, F] = process_model(Xb(:,k-1),T); %predicted state
98
99      Pa = F*Pb(:,:,k-1)*F' + Q;  %predicted covariance
100     [Ya, H1_temp, H2_temp] = ...
           measurement_model([Xa(1);Xa(3);Xa(5);Xa(7:n)],[Xb(1,k-1);Xb(3,k-1)],B,f); ...
           %predicted measurement
101     H = [H1_temp(:,1), zeros(m,1), H1_temp(:,2), zeros(m,1), ...
            H1_temp(:,3), zeros(m,1), H1_temp(:,4:size(H1_temp,2))];
102     H_prev = [H2_temp(:,1), zeros(m,1), H2_temp(:,2), zeros(m,1), ...
            H2_temp(:,3), zeros(m,1), H2_temp(:,4:size(H2_temp,2))];
103     %update step
104     W = R + H_prev*Q*(H_prev');
105     K(:,:,k) = Pa*H'/(H*Pa*H' + W); %noise is additive (constant ...
            velocity model) so G_w is identity
106     Xb(:,k) = Xa + K(:,:,k)*(Yn(:,k) - Ya);
107     Pb(:,:,k) = (eye(n) - K(:,:,k)*H)*Pa;
108
109     %covariance matrix eigenvalues
110     lambda(:,k) = eig(Pb(:,:,k));
111 end
112
113 figure('Name','Case 6: without beacon no ...
       disparity','NumberTitle','off');
114 subplot(5,1,1);
115 hold on;
116 title('Position X Estimate');
117 plot(X(1,:),'g');
118 plot(Xb(1,:),'r');
119 legend('True State','Estimate','Location','NorthEastOutside');
120 subplot(5,1,2);
121 hold on;
122 title('Position Y Estimate');
123 plot(X(2,:),'g');
```

```
124 plot(Xb(3,:),'r');
125 legend('True State','Estimate','Location','NorthEastOutside');
126 subplot(5,1,3);
127 hold on;
128 title('Angle Estimate');
129 plot(X(3,:),'g');
130 plot(Xb(5,:),'r');
131 legend('True State','Estimate','Location','NorthEastOutside');
132 subplot(5,1,4);
133 hold on;
134 title('Landmark X Estimate');
135 plot(X(4,:),'g');
136 plot(Xb(7,:),'r');
137 legend('True State','Estimate','Location','NorthEastOutside');
138 subplot(5,1,5);
139 hold on;
140 title('Landmark Y Estimate');
141 plot(X(5,:),'g');
142 plot(Xb(8,:),'r');
143 legend('True State','Estimate','Location','NorthEastOutside');
144
145 figure('Name','Case 6: no beacon no disparity, ...
       Eigenvalues','NumberTitle','off');
146 hold on;
147 subplot(8,1,1);
148 title('Lambda 1');
149 plot(lambda(1,:),'b');
150 subplot(8,1,2);
151 title('Lambda 2');
152 plot(lambda(2,:),'b');
153 subplot(8,1,3);
154 title('Lambda 3');
155 plot(lambda(3,:),'b');
156 subplot(8,1,4);
157 title('Lambda 4');
158 plot(lambda(4,:),'b');
```

```
159 subplot(8,1,5);
160 title('Lambda 5');
161 plot(lambda(5,:),'b');
162 subplot(8,1,6);
163 title('Lambda 6');
164 plot(lambda(6,:),'b');
165 plot(lambda(4,:),'b');
166 subplot(8,1,7);
167 title('Lambda 7');
168 plot(lambda(7,:),'b');
169 subplot(8,1,8);
170 title('Lambda 8');
171 plot(lambda(8,:),'b');
172
173 figure('Name','Case 6: no beacon no disparity, ...
       World','NumberTitle','off');
174 hold on;
175 axis equal;
176 plot_camera([X(1,1:floor(size(X,2)/10):size(X,2))' ...
       X(2,1:floor(size(X,2)/10):size(X,2))' ...
       X(3,1:floor(size(X,2)/10):size(X,2))'],25);
177 plot_landmark(B,25);
178 plot_landmark(landmarks(1:l,:),25,[1 1 1]);
179 for k=1:floor(size(X,2)/10):size(X,2)
180     cov = [Pb(1,1,k), Pb(1,3,k);
181            Pb(3,1,k), Pb(3,3,k)];
182     lambda = eig(cov);
183     [S D] = eig(cov);
184     scale = 2.447;
185     [a a_i] = max(lambda); %find maximum eigenvalue
186     [b b_i] = min(lambda); %find minimum eigenvalue
187     a = scale*sqrt(a); %scale according to confidence interval
188     b = scale*sqrt(b);
189     hold on;
190     plot(Xb(1,k),Xb(3,k),'r*');
191     if cov(1,1)≥cov(2,2) %resolve tilt of the ellipse
```

```
192         theta = atan2(S(2,a_i),S(1,a_i));
193         x_axis = a;
194         y_axis = b;
195     else
196         theta = atan2(S(2,b_i),S(1,b_i));
197         x_axis = b;
198         y_axis = a;
199     end
200
201     r=disc(Xb(1,k),Xb(3,k),x_axis,y_axis,theta,[1 1 0]);
202     set(r,'FaceAlpha',0.5);
203 end
204 k = size(X,2);
205     %find and plot uncertainty ellipse for landmark
206     cov = [Pb(7,7,k), Pb(7,8,k);
207            Pb(8,7,k), Pb(8,8,k)];
208     lambda1 = eig(cov);
209     [S D1] = eig(cov);
210     scale = 2.447;
211     [a a_i] = max(lambda1); %find maximum eigenvalue
212     [b b_i] = min(lambda1); %find minimum eigenvalue
213     a = scale*sqrt(a); %scale according to confidence interval
214     b = scale*sqrt(b);
215     hold on;
216     if cov(1,1)≥cov(2,2) %resolve tilt of the ellipse
217         theta = atan2(S(2,a_i),S(1,a_i));
218         x_axis = a;
219         y_axis = b;
220     else
221         theta = atan2(S(2,b_i),S(1,b_i));
222         x_axis = b;
223         y_axis = a;
224     end
225     plot(Xb(7,k),Xb(8,k),'r*');
226     r=disc(Xb(7,k),Xb(8,k),x_axis,y_axis,theta,[0 1 1]);
227
```

```
figure;
mov = avifile('simulation.avi', 'fps', 15);
for k = 1:floor(size(X,2)/100):size(X,2)
    clf
    hold on;
    axis([-180 380 -180 200]);
    %plot world
    plot_landmark(B,25);
    plot_landmark(landmarks(1:l,:),25,[1 1 1]);
    plot_camera(X(1:3,k)',25);
    %find and plot uncertainty ellipse for camera
    cov = [Pb(1,1,k), Pb(1,3,k);
           Pb(3,1,k), Pb(3,3,k)];
    lambda1 = eig(cov);
    [S D] = eig(cov);
    scale = 2.447;
    [a a_i] = max(lambda1); %find maximum eigenvalue
    [b b_i] = min(lambda1); %find minimum eigenvalue
    a = scale*sqrt(a); %scale according to confidence interval
    b = scale*sqrt(b);
    hold on;
    plot(Xb(1,k),Xb(3,k),'r*');
    if cov(1,1)≥cov(2,2) %resolve tilt of the ellipse
        theta = atan2(S(2,a_i),S(1,a_i));
        x_axis = a;
        y_axis = b;
    else
        theta = atan2(S(2,b_i),S(1,b_i));
        x_axis = b;
        y_axis = a;
    end

    r=disc(Xb(1,k),Xb(3,k),x_axis,y_axis,theta,[1 1 0]);
    set(r,'FaceAlpha',0.5);
    %find and plot uncertainty ellipse for landmark
    cov = [Pb(7,7,k), Pb(7,8,k);
```

```
            Pb(8,7,k), Pb(8,8,k)];
    lambda1 = eig(cov);
    [S D1] = eig(cov);
    scale = 2.447;
    [a a_i] = max(lambda1); %find maximum eigenvalue
    [b b_i] = min(lambda1); %find minimum eigenvalue
    a = scale*sqrt(a); %scale according to confidence interval
    b = scale*sqrt(b);
    hold on;
    if cov(1,1)≥cov(2,2) %resolve tilt of the ellipse
        theta = atan2(S(2,a_i),S(1,a_i));
        x_axis = a;
        y_axis = b;
    else
        theta = atan2(S(2,b_i),S(1,b_i));
        x_axis = b;
        y_axis = a;
    end
    plot(Xb(7,k),Xb(8,k),'r*');
    r=disc(Xb(7,k),Xb(8,k),x_axis,y_axis,theta,[0 1 1]);
    plot(X(1,:),X(2,:),'b:'); %plot trajecetory
    set(r,'FaceAlpha',0.5);
    opengl('software');
    W_frame = getframe(gcf);
    mov = addframe(mov, W_frame);
end
mov = close(mov);
```