



UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Robot Localisation with Unknown Correspondence using Probabilistic Graphical Models

Damon Patrick McLoughlin
20778929

Report submitted in partial fulfilment of the requirements of the module
Project (E) 448 for the degree Baccalaureus in Engineering in the Department of
Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Dr C.E. van Daalen

November 2020



UNIVERSITEIT • STELLENBOSCH • UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.

Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.

I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.


I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelike aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.

Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.

I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

20778929	
D.P. McLoughlin	09/11/2020

Abstract

English

The aim of this project is to explore the application of probabilistic graphical models (PGMs) to the mobile robot localisation problem with unknown measurement correspondence. This problem states that a mobile robot must infer its position and orientation in relation to a map of its environment. The classical methods of solving this problem are limited.

PGMs are used to provide structure to a probabilistic problem by depicting random variable (RV) relationships, upon which principled reasoning can be applied to query certain distributions.

Firstly this report explains the theoretical concepts required to understand both the localisation problem and PGMs. Next, a PGM for a simplified version of localisation is designed, and then extended in a step-wise fashion to describe this project's problem. This study's results showed that the designed model provides a valuable and easily extendable foundation for applying PGMs to more complex mobile robot models and localisation problems.

Afrikaans

Die doel van hierdie projek is om die toepassing van waarskynlikheids grafiese modelle te ondersoek (WGMs) vir die probleem van lokalisering van mobiele robotte met onbekende metingkorrespondensie. Hierdie probleem stel dat 'n mobiele robot sy posisie en oriëntasie in verhouding tot 'n kaart van sy omgewing moet aflei. Die klassieke metodes om hierdie probleem op te los, is beperk.

WGM's word gebruik om struktuur aan 'n waarskynlikheids probleem te bied deur willekeurige veranderlike (WV) verwantskappe uit te beeld, waarop beginselredes toegepas kan word om sekere verspreidings te ondersoek.

Eerstens, word die teoretiese begrippe uiteengesit om beide die lokaliseringsprobleem en WGM's te verstaan. Daarna, word 'n WGM vir 'n vereenvoudigde weergawe van lokalisering ontwerp, en dan stapsgewys uitgebrei om die probleem beskryf. Die resultate het getoon dat die ontwerpte model 'n waardevolle en maklik uitbreidbare basis bied vir die toepassing van WGM's op meer komplekse mobiele robotmodelle en lokaliseringsprobleme.

Contents

Declaration	i
Abstract	ii
List of Figures	vi
List of Tables	viii
Nomenclature	ix
1. Introduction	1
1.1. Background and Motivation	1
1.2. Problem Statement	2
1.3. Objectives	3
1.4. Scope	3
1.5. Report Overview	3
2. Gaussian Probability Distributions	4
2.1. Gaussian Representations	4
2.1.1. The Covariance Form	4
2.1.2. The Canonical Form	6
2.1.3. The Square Root Canonical and Covariance Forms	7
2.2. Operations on Gaussian Distributions	7
2.2.1. Operations on the Canonical Form	7
2.2.2. Weak Marginalisation using the Covariance Form	8
2.3. Canonical Form of Conditional Probability Distributions	9
3. Discrete Probability Distributions	11
3.1. The Table Form	11
3.1.1. Operations on the Table Form	11
4. Designing the Mobile Robot Model for the Localisation Problem	13
4.1. Problem Definition	13
4.2. Linear Robot Motion Model	14
4.3. Linear Robot Measurement Model	15

5. The Kalman Filter	18
5.1. Technique Overview	18
5.2. Algorithm Explanation	18
6. Probabilistic Graphical Models	20
6.1. Technique Outline	20
6.2. Bayesian Networks	21
6.3. Cluster Graphs	22
6.4. Message Passing	23
6.5. Hybrid Networks	26
7. Solving Simpler Localisation Problems with Probabilistic Graphical Models	28
7.1. PGM Solution for the Direct Measurement Model	28
7.1.1. Modelling the Problem	28
7.1.2. Inference and Results	31
7.2. PGM Solution for the Single Landmark Measurement Model	33
7.2.1. Modelling the Problem	33
7.2.2. Inference and Results	33
7.3. PGM Solution for the Multiple Landmark Measurement Model with Known Correspondence	34
7.3.1. Modelling the Problem	34
7.3.2. Inference and Results	35
8. Solving the Unknown Correspondence Localisation Problem with a Probabilis- tic Graphical Model	36
8.1. Modelling the Problem	36
8.2. Inference	38
8.3. Results	41
9. Summary and Conclusion	44
9.1. Future Work	45
Bibliography	46
A. Project Planning Schedule	47
B. Outcomes Compliance	48
C. Additional Results	49
D. The Square Root Canonical Form for Numerical Stability	52

E. Localisation Cluster Potentials	54
E.1. Direct Measurement Model	54
E.2. Single Landmark Measurement Model	55
E.3. Multiple Landmark Measurement Model with Known Correspondence . . .	55
E.4. Multiple Landmark Measurement Model with Unknown Correspondence .	55
F. Prior Joint Association Variable Probability Distributions	56

List of Figures

2.1.	Confidence Ellipse with major and minor axes labelled.	5
2.2.	Two Gaussian distributions and their (a) sum and (b) unnormalised weak marginalisation.	9
5.1.	The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{6.4,8.3\}$), with the corresponding KF beliefs using the direct measurement model.	19
6.1.	Example of a Bayesian network (node CPDs indicated).	21
6.3.	Example of a tree cluster graph.	25
7.1.	Bayes network for the localisation problem using direct measurements (node CPDs indicated).	29
7.2.	Cluster graph for the initial localisation problem, showing cluster numbering and messages.	29
7.3.	The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{10,8\}$), with the corresponding PGM beliefs using the direct measurement model.	32
7.4.	Bayesian network for localisation using multiple measurements with known correspondence.	34
8.1.	Hybrid network for localisation using multiple measurements with unknown correspondence (node CPDs indicated).	36
8.2.	Cluster graph for multiple measurements with unknown correspondence.	38
8.3.	The separate (a) discrete and (b) continuous cluster graph structures.	38
8.4.	The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{9.5,8.5\}$), with the corresponding PGM beliefs for the (a) known and (b) unknown correspondence localisation problems, using an uncertain initial pose distribution.	41
8.5.	Beliefs computed by PGMs for the (a) known and (b) unknown correspondence localisation problems using an uncertain initial pose distribution and measurements of all landmarks in the map at all times	43

C.1. The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{6.2,9.2\}$), with the corresponding PGM beliefs using the single landmark measurement model.	49
C.2. The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{5.9,9.8\}$), with the corresponding PGM beliefs using the multiple landmark measurement model with known correspondence.	49
C.3. The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{9.5,8.5\}$), with the corresponding PGM beliefs for the (a) known and (b) unknown correspondence localisation problems, using a more certain initial pose distribution.	50
C.4. The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{9,3.6\}$), with the corresponding PGM beliefs for the (a) known and (b) unknown correspondence localisation problems, using a more certain initial pose distribution.	51

List of Tables

6.1. Example of one row in a canonical table.	27
8.1. A row in the unknown correspondence PGM’s measurement cluster canonical table.	37
A.1. Project Plan Schedule (2020).	47
B.1. ECSA Exit Level Outcomes (ELOs), how they were achieved and in which chapters.	48
F.1. The prior joint association variable probability distribution when three measurements are taken with a map of three landmarks.	56
F.2. The prior joint association variable probability distribution when two mea- surements are taken with a map of three landmarks.	56
F.3. The prior joint association variable probability distribution when one mea- surement is taken with a map of three landmarks.	57

Nomenclature

Notation

x	Scalar value.
\mathbf{x}	Vector of values.
X	Matrix of values.
X^T	Transpose of matrix of values.
X^{-1}	Inverse of matrix of values.
$p(x)$	Probability distribution for variable x .
$\boldsymbol{\mu}$	Statistical mean vector.
Σ	Statistical covariance matrix.
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	Covariance form of a Gaussian distribution.
$\mathcal{C}(\mathbf{x}; K, \mathbf{h}, g)$	Canonical form of a Gaussian distribution.
\mathbf{x}_t	Vector of values at time step ' t '.
X_t	Matrix of values at time step ' t '.
$val(\mathbf{x})$	Any value in the random variable x 's scope.
$\tilde{p}(x)$	Any unnormalised distribution over variable x , that is not the prior
$\phi_t(x)$	Factor at timestep t , with a scope of variable x .

Acronyms

RV	Random Variable
PDF	Probability Density Function
PGM	Probabilistic Graphical Model
SLAM	Simultaneous Localisation And Mapping
LBP	Loopy Belief Propagation
CPD	Conditional Probability Distribution
KF	Kalman Filter
EKF	Extended Kalman Filter
BN	Bayesian Network
RIP	Running Intersection Property
BP	Belief Propagation

Chapter 1

Introduction

1.1. Background and Motivation

Recent advancements in the field of robotics have led to the development of a wide range of robots– from humanoid companions to autonomous cars. A large subset of these robots must traverse their environments unassisted to perform useful tasks. To do this, they must be informed of their relative location accurately. Consider a humanoid robot that is designed to pick up a box and move it from one shelf to another inside a factory. If it cannot determine its location in the factory and in relation to the box and shelves, it cannot perform its function. Thus, it is vital for mobile robots to develop an accurate perception of their position and orientation in their environment. The process of building this perception, or the robot’s ‘belief’, is known as the mobile robot localisation problem.

Typically, a robot takes high-resolution observations of its surroundings, resulting in very high-dimension measurement data. This information can be dramatically reduced in volume by selecting the data pertaining to a small set of salient points, called ‘features’ or ‘landmarks’. This landmark-based measurement model is commonly adopted, and is employed throughout this study. In this model, the robot has access to a map of its environment’s landmarks. Each landmark is allocated an identity, determined, for example, by the robot’s imaging sensors. Each measurement is thus allocated a ‘source landmark’. This is referred to as the ‘known association’, or ‘known correspondence’ localisation problem, because the robot is informed of which landmarks are responsible for which measurements. However, many sensors, such as infrared distance sensors, can only measure distance. Consequently, measurements are obtained without information about their source landmarks. Localising a robot with these measurements is referred to as the ‘unknown association’ or ‘unknown correspondence’ localisation problem. This problem is also applicable when the measurement sensors cannot **reliably** identify the landmarks in its environment.

Mobile robots face a number of uncertainties in determining their belief. Firstly, the measurement data is never a perfect representation of the robot’s distances from landmarks. There are no practically realisable disturbance-free measurement sensors. The resulting measurement noise means the robot cannot determine its exact position from this data.

Secondly, no disturbance-free actuators exist either. The actual movement of the robot will therefore differ from its instructed movement. Probabilistic robotics incorporates the aforementioned uncertainties by generalising deterministic robot motion equations [1]. This is achieved by describing uncertain quantities with random variables (RVs) and associating them with probability density functions (PDFs). For this study, the robot’s motion and measurements will be described with continuous distributions.

According to Koller, probabilistic graphical models (PGMs) are defined as “a rich framework for encoding probability distributions over complex domains: joint (multivariate) distributions over large numbers of random variables that interact with each other” [2]. These graphical models can be used to provide structure to a probabilistic problem by depicting the nature of the interactions between the RVs. Various statistical outcomes can be determined by applying probabilistic reasoning algorithms to the models.

The definition of PGMs given above illustrates that these structures can certainly be used to model the localisation problem. The robot’s belief distribution can then be determined by performing probabilistic reasoning, also known as ‘inference’, on the model. However, the application of PGMs to the unknown association problem specifically, is largely unexplored. There are three classical solutions. The first is to calculate the probability distributions jointly, in a centralised manner, which is computationally expensive. The second is to make ‘hard’ decisions about measurement allocations, which is not robust. The final option is to use multiple hypothesis tracking (MHT), which is a sophisticated combination of the previous two techniques. Unfortunately, MHT is still somewhat computationally intense and lacks some robustness [1].

Localisation using a PGM offers a much more computationally efficient solution, because inference can be performed on a distributed network. The solution is robust since it involves no approximations or ‘hard’ decisions. This is especially important when localisation is only part of a complex problem the robot must solve. An example of this is the simultaneous localisation and mapping (SLAM) problem. A well-known problem in robotics, the SLAM problem involves localisation without a map. As the robot moves, it must both localise itself and generate its own map of the environment. It is anticipated that this study’s results could provide a foundation upon which the application of PGMs to the SLAM problem can be explored.

1.2. Problem Statement

The aim of this study is to explore the use of a PGM to model the mobile robot localisation problem with unknown correspondence, and to solve it by applying the loopy belief propagation (LBP) inference algorithm. Additionally, the effects of unknown measurement correspondence on localisation using a PGM is explored.

1.3. Objectives

This study aims to achieve the following objectives, building towards the full solution:

1. Model a simple localisation problem that obtains direct measurements of the robot's position with a PGM and perform inference to compute its beliefs. Verify these beliefs using the Kalman filter (KF).
2. Extend this PGM to model the localisation problem with known correspondence.
3. Extend the previous PGM to model the localisation problem with unknown correspondence. Perform loopy belief propagation on the PGM to compute the robot's beliefs for a simulated robot's motion.
4. Identify the differences in the results of PGMs that model the known and unknown correspondence problems, for different cases.

1.4. Scope

In order to focus on the unknown correspondence problem, the simulated robot model was kept to a simple, linear model. This enabled a more valuable comparison between the beliefs of the known and unknown correspondence PGMs, since differences could be assigned to the measurement association knowledge, rather than to non-linear model behaviour. This enabled a clearer design process for the final PGM which focused on the modelling of the unknown measurement association aspect specifically.

1.5. Report Overview

To achieve this study's aims, we studied the literature on Gaussian RVs, the localisation problem in probabilistic robotics, and PGMs. We define this study's representations of – and operations on – both continuous and discrete PDF's in Chapters 2 and 3. The localisation problem is defined and the simulated robot's model is designed in Chapter 4. In Chapter 5, the KF's algorithm is explained and applied to a simple localisation problem. Chapter 6 explains the basic PGM concepts and inference algorithms required to understand the content of the following two chapters. Three preliminary PGMs were designed in Chapter 7, the last of which describes the known correspondence problem. Performing inference on these models is detailed. The intuition shown in the design of these PGMs was used to design the final PGM in Chapter 8. We explain the complex process of performing inference on this graph, and the resulting beliefs were compared to those of the known correspondence PGM.

Chapter 2

Gaussian Probability Distributions

A continuous random variable (RV) can hold one of infinitely many continuous values. A probability density function (PDF) describes the likelihoods of these values. The Gaussian distribution is a PDF that presumes the likelihood of a continuous RV's values decrease exponentially with respect to the squared distance to its mean. This squared distance is called the variance. Gaussian distributions are used in this study due to their mathematically convenient provision of probabilistic approximations of reality.

In this chapter we introduce the covariance and canonical forms of Gaussian distributions that were useful for this project. We briefly discuss the familiar covariance form to describe how it was used in this study. The less commonly used canonical form was crucial for the PGM solution to the localisation problem and is therefore discussed in more detail. We also explain the operations on these representations that are relevant to this study. This discussion is largely based on Koller and Friedman's work [3].

2.1. Gaussian Representations

2.1.1. The Covariance Form

The covariance form is the standard and most intuitive representation of Gaussian probability distributions. A univariate Gaussian PDF for variable x , $p(x)$, is described by

$$p(x) = \mathcal{N}(\mu, \sigma) \triangleq \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[\frac{-(x - \mu)^2}{2\sigma^2} \right], \quad (2.1)$$

where μ is the mean and σ^2 the variance. Since there are many random variables involved, the univariate distribution is not useful for localisation. Extending Equation 2.1 to the multivariate case defines the PDF of an N -dimensional vector \mathbf{x} as

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) \triangleq \frac{1}{(2\pi|\Sigma|)^{N/2}} \exp \left[-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right], \quad (2.2)$$

where $\boldsymbol{\mu}$ represents the mean vector of the distribution and Σ represents the covariance matrix. The mean is the most likely combination of values for the distribution's variables. The covariance matrix categorises the spread of the distribution over its variables.

This study primarily uses the covariance form’s parameters to plot the robot’s uncertain knowledge of its position. This is done by visualising a contour of the Gaussian distribution, commonly known as a confidence ellipse. We used a 95% confidence ellipse, that contains 95% of the samples that can be extracted from its source Gaussian distribution. Spruyt [4] calculates this ellipse’s parameters as

$$a = 2\sqrt{5.991\lambda_1}, \quad (2.3)$$

$$b = 2\sqrt{5.991\lambda_2}, \text{ and} \quad (2.4)$$

$$\alpha = \arctan\left(\frac{\mathbf{v}_1(y)}{\mathbf{v}_1(x)}\right), \quad (2.5)$$

where a and b are the major and minor axis lengths respectively, and α is the rotation of the major axis to the horizontal, as seen in Figure 2.1. The eigendecomposition of the covariance will yield two eigenvectors. The larger of the two is represented by \mathbf{v}_1 .

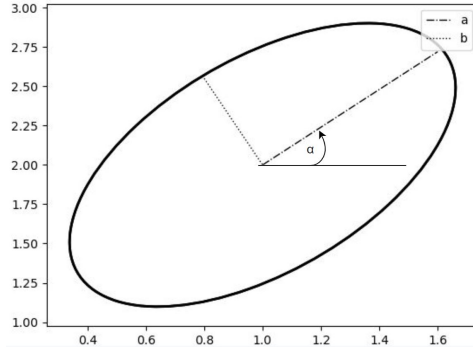


Figure 2.1: Confidence Ellipse with major and minor axes labelled.

In this study, the robot’s position is described by the x and y coordinates, resulting in a two-dimensional state vector. The lengths of the ellipse’s axes correspond to the eigenvalues λ_k of the resulting 2×2 covariance matrix, while the vectors of the axes correspond to the eigenvectors \mathbf{v}_k of the covariance matrix. These quantities are determined by solving a standard eigendecomposition problem: $\Sigma \mathbf{v}_k = \lambda_k \mathbf{v}_k$.

The centre of the ellipse represents the centre of the Gaussian distribution, or the mean. A smaller confidence ellipse is a result of smaller entries in the covariance matrix. This represents a narrower distribution, where it is less likely that the RV vector is far from the mean.

Representing probability distributions in the covariance form requires that the distributions integrate to one, to ensure a valid covariance matrix. However, many distributions do not satisfy this requirement. As a result, relying on the covariance form to represent distributions in a PGM would vastly restrict which problems the PGM could describe. To circumvent this problem, we used the more general canonical form of Gaussian distribution in this study’s PGMs.

2.1.2. The Canonical Form

The canonical form is defined as

$$\mathcal{C}(\mathbf{x}; K, \mathbf{h}, g) = \exp \left\{ -\frac{1}{2} \mathbf{x}^T K \mathbf{x} + \mathbf{h}^T \mathbf{x} + g \right\}, \quad (2.6)$$

where K is the information matrix, with larger entries reflecting that the distribution possesses more certainty. The potential vector \mathbf{h} represents the mean, scaled by this certainty, and g is a scaling constant. These parameters are found using the multivariate Gaussian distribution in Equation 2.2, as

$$p(\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \exp \left\{ -\frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \boldsymbol{\mu}^T \Sigma^{-1} \boldsymbol{\mu} - \log \left[(2\pi|\Sigma|)^{N/2} \right] \right\} \quad (2.7)$$

$$\Rightarrow \mathcal{N}(\boldsymbol{\mu}, \Sigma) = \mathcal{C}(\mathbf{x}; K, \mathbf{h}, g), \text{ where} \quad (2.8)$$

$$K = \Sigma^{-1} \quad (2.9)$$

$$\mathbf{h} = \Sigma^{-1} \boldsymbol{\mu} \quad (2.10)$$

$$g = -\frac{1}{2} \boldsymbol{\mu}^T \Sigma^{-1} \boldsymbol{\mu} - \log \left[(2\pi|\Sigma|)^{N/2} \right]. \quad (2.11)$$

Note that the generality of the canonical form lies in the fact that K does not **have** to be the inverse of a valid covariance matrix. The only requirement for the canonical form to represent a valid Gaussian PDF is that K is invertible. This is only true if the matrix is positive definite. In other words, this requires that

$$\mathbf{x}^T \Sigma \mathbf{x} > 0, \text{ for any value } \mathbf{x} \neq \mathbf{0}. \quad (2.12)$$

The covariance form's parameters are found in terms of the canonical form's parameters by reversing the subject of Equations 2.9 and 2.10:

$$\Sigma = K^{-1} \quad (2.13)$$

$$\boldsymbol{\mu} = K^{-1} \mathbf{h}. \quad (2.14)$$

For the message passing algorithm explained in Section 6.4, it is necessary to describe a vacuous canonical representation. The term 'vacuous' means that the distribution contains no information about its RVs. A vacuous canonical form is described by:

$$K = 0, \mathbf{h} = \mathbf{0} \text{ and } g = 0. \quad (2.15)$$

It can easily be shown that this is equivalent to unity by substituting Equation 2.1.2 into the canonical form's definition in Equation 2.6.

2.1.3. The Square Root Canonical and Covariance Forms

Using the square root canonical form instead of the standard canonical form substantially improves the computational efficiency and numerical stability of this study's PGM solution. The square root form is given by $\mathcal{C}(\mathbf{x}; L, \mathbf{h}, g)$, where L is obtained from the K matrix via the Cholesky decomposition. The Cholesky decomposition is defined as the factorisation of a positive-definite Hermitian matrix into the product of a lower triangular matrix and its transpose. This factorisation is mathematically given by $K = LL^T$. The square root covariance form is found in a similar way, by replacing the covariance matrix with the upper triangular matrix $R = L^{-T}$. The application of the square root form in this study is further detailed in Appendix D.

This section detailed the two main representations of Gaussian distributions that were used. We explained the confidence ellipse visualisation of Gaussian distributions, and the square root forms. Next, the operations required for the use of these distributions in PGMs are described.

2.2. Operations on Gaussian Distributions

2.2.1. Operations on the Canonical Form

Operations on the canonical form are mathematically convenient. The operations needed for the PGMs in Chapters 7 and 8 are explained below.

1. Broadening scope: The scope of a canonical form is expanded to include more variables by adding zero entries to K and \mathbf{h} as illustrated below:

$$\mathcal{C}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \end{bmatrix}, g\right) = \mathcal{C}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}; \begin{bmatrix} K_{xx} & K_{xy} & 0 \\ K_{yx} & K_{yy} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \\ \mathbf{0} \end{bmatrix}, g\right). \quad (2.16)$$

2. Rearranging scope: The scope of a canonical form is re-ordered by switching appropriate elements in K and \mathbf{h} :

$$\mathcal{C}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}; \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} \\ K_{yx} & K_{yy} & K_{yz} \\ K_{zx} & K_{zy} & K_{zz} \end{bmatrix}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \\ \mathbf{h}_z \end{bmatrix}\right) = \mathcal{C}\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{z} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} K_{xx} & K_{xz} & K_{xy} \\ K_{zx} & K_{zz} & K_{zy} \\ K_{yx} & K_{zy} & K_{yy} \end{bmatrix}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_z \\ \mathbf{h}_y \end{bmatrix}\right), \quad (2.17)$$

where g is omitted for conciseness, as it is unchanged.

3. Multiplication: The product of two canonical forms with scope \mathbf{x} is calculated as:

$$\mathcal{C}(\mathbf{x}; K_1, \mathbf{h}_1, g_1) \mathcal{C}(\mathbf{x}; K_2, \mathbf{h}_2, g_2) = \mathcal{C}(\mathbf{x}; K_1 + K_2, \mathbf{h}_1 + \mathbf{h}_2, g_1 + g_2). \quad (2.18)$$

The scope of two distributions must be extended to be identical before their product can be found.

4. Division: The quotient of two canonical forms with scope \mathbf{x} is defined as:

$$\mathcal{C}(\mathbf{x}; K_1, \mathbf{h}_1, g_1) \div \mathcal{C}(\mathbf{x}; K_2, \mathbf{h}_2, g_2) = \mathcal{C}(\mathbf{x}; K_1 - K_2, \mathbf{h}_1 - \mathbf{h}_2, g_1 - g_2). \quad (2.19)$$

The scopes of the two distributions must also be identical before dividing them.

5. Marginalisation: Using the canonical form

$$\mathcal{C} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \end{bmatrix}, g \right), \quad (2.20)$$

the marginal distribution over \mathbf{x} is obtained by integrating over \mathbf{y} :

$$\mathcal{C}(\mathbf{x}; K', \mathbf{h}', g') = \int_{\mathbf{y}} \mathcal{C} \left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}; \begin{bmatrix} K_{xx} & K_{xy} \\ K_{yx} & K_{yy} \end{bmatrix}, \begin{bmatrix} \mathbf{h}_x \\ \mathbf{h}_y \end{bmatrix}, g \right) d\mathbf{y}. \quad (2.21)$$

This integral is only finite if K_{yy} is the inverse of a legal covariance matrix. The result of this integral is then:

$$\begin{aligned} K' &= K_{xx} - K_{xy}K_{yy}^{-1}K_{yx}, \quad \mathbf{h}' = \mathbf{h}_x - K_{xy}K_{yy}^{-1}\mathbf{h}_y \quad \text{and} \\ g' &= g + \frac{1}{2} \left(\log \left\{ |2\pi K_{yy}^{-1}| \right\} + \mathbf{h}_y^T K_{yy}^{-1} \mathbf{h}_y \right). \end{aligned} \quad (2.22)$$

6. Reduction: A canonical form can be reduced with information, or **evidence**, about a variable in its scope. For example, the canonical representation in Equation 2.20 is reduced by observing $\mathbf{y} = \hat{\mathbf{y}}$, giving

$$\mathcal{C} \left(\mathbf{x}; K_{xx}, (\mathbf{h}_x - K_{xy}\hat{\mathbf{y}}), \left(g + \mathbf{h}_y^T - \frac{1}{2}\hat{\mathbf{y}}^T K_{yy}\hat{\mathbf{y}} \right) \right). \quad (2.23)$$

These operations are required to use the canonical form for Gaussian distributions in PGMs. Next, weak marginalisation using the covariance form is described.

2.2.2. Weak Marginalisation using the Covariance Form

The necessity of Gaussian mixture distributions in the PGM for unknown correspondence is discussed in Section 6.5. A Gaussian mixture distribution is a weighted sum of k Gaussian distributions, defined as $\{[w_i, \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)]\}_{i=1}^k$ with $\sum_{i=1}^k w_i = 1$. To maintain computational tractability, each mixture in the PGM is approximated by a Gaussian distribution, called the weak marginalisation. The weak marginalisation's mean is the weighted average of the means of the distributions in the mixture. The covariance is

the weighted sum of the distributions' covariances, added to the sum of the weighted covariances between the marginalisation's mean and the distributions' means. It is found for the mixture defined above as $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where

$$\boldsymbol{\mu} = \sum_{i=1}^k w_i \boldsymbol{\mu}_i \quad \text{and} \quad (2.24)$$

$$\Sigma = \sum_{i=1}^k w_i \Sigma_i + \sum_{i=1}^k w_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T. \quad (2.25)$$

It is clear from Figure 2.2 that a sum of Gaussian distributions is vastly different to their weak marginalisation. This distinction is important for Section 8.2.

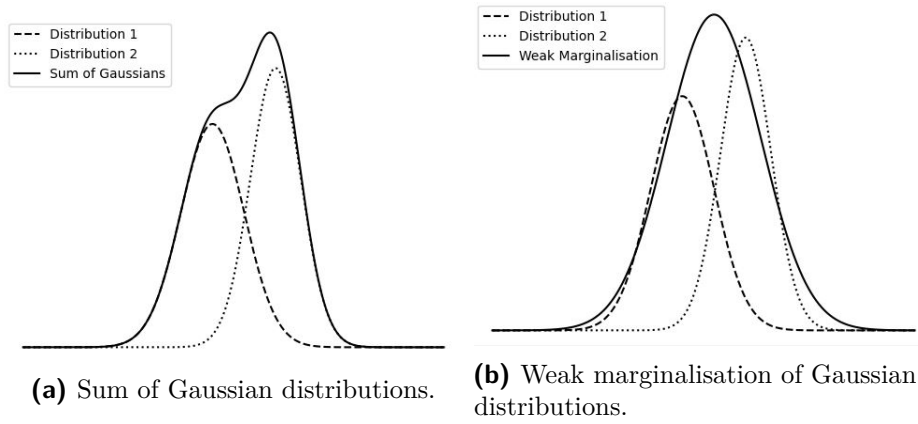


Figure 2.2: Two Gaussian distributions and their (a) sum and (b) unnormalised weak marginalisation.

The weak marginalisation has a major limitation, in that it can only be found for a mixture of Gaussian distributions with **valid** covariance matrices. Section 8.2 explains how the model used avoids this limitation. The weak marginalisation plays a vital part of the solution to the unknown correspondence localisation problem.

Crucial aspects of the localisation problem are described with conditional probability distributions (CPDs). Often, these distributions do not have an invertible covariance matrix. We therefore derive their canonical representation, for use in a PGM.

2.3. Canonical Form of Conditional Probability Distributions

A CPD describes the distribution of a set of variables \mathbf{x} given knowledge of the values of another set of variables \mathbf{y} . This can be written as

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{y})}. \quad (2.26)$$

Often, conditional PDFs will not have an invertible covariance matrix. They must therefore be represented in the canonical form to be used in a PGM. It is helpful to derive the general canonical representation of a conditional probability distribution $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$. This will result in a canonical distribution with a scope of $\{\mathbf{x}, \mathbf{y}, \mathbf{z}\}$. This general derivation is largely based on the work done by Schoeman [5].

Consider a linear function of two Gaussian vectors \mathbf{y} and \mathbf{z} with added Gaussian noise $\boldsymbol{\epsilon}$, resulting in a random variable vector $\mathbf{x} = A\mathbf{y} + B\mathbf{z} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is by $p(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \Sigma)$. The covariance form of the CPD $p(\mathbf{x}|\mathbf{y}, \mathbf{z})$ can easily be found as $\mathcal{N}(A\mathbf{y} + B\mathbf{z}, \Sigma)$. Expanding this with the multivariate Gaussian definition in Equation 2.2 gives

$$p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = \frac{1}{(2\pi|\Sigma|)^{N/2}} \exp \left\{ -\frac{1}{2}(\mathbf{x} - (A\mathbf{y} + B\mathbf{z}))^T \Sigma^{-1}(\mathbf{x} - (A\mathbf{y} + B\mathbf{z})) \right\}. \quad (2.27)$$

The expression $(\mathbf{x} - (A\mathbf{y} + B\mathbf{z}))$ can be rewritten as

$$\mathbf{x} - (A\mathbf{y} + B\mathbf{z}) = \begin{bmatrix} I & -A & -B \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix} = \begin{bmatrix} I \\ -A^T \\ -B^T \end{bmatrix}^T \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{z} \end{bmatrix}. \quad (2.28)$$

Using this result and defining $\mathbf{w} = [\mathbf{x} \ \mathbf{y} \ \mathbf{z}]^T$ allows Equation 2.27 to be simplified to:

$$p(\mathbf{x}|\mathbf{y}, \mathbf{z}) = \exp \left\{ -\frac{1}{2} \mathbf{w}^T \begin{bmatrix} I \\ -A^T \\ -B^T \end{bmatrix} \Sigma^{-1} \begin{bmatrix} I \\ -A^T \\ -B^T \end{bmatrix}^T \mathbf{w} - \ln \left[(2\pi|\Sigma|)^{N/2} \right] \right\}. \quad (2.29)$$

Using the canonical form's definition in Equation 2.6, the parameters are identified as

$$K_c = \begin{bmatrix} I \\ -A^T \\ -B^T \end{bmatrix} \Sigma^{-1} \begin{bmatrix} I \\ -A^T \\ -B^T \end{bmatrix}^T, \quad \mathbf{h}_c = \mathbf{0} \text{ and } g_c = -\ln \left[(2\pi|\Sigma|)^{N/2} \right]. \quad (2.30)$$

The canonical form of the general CPD is thus $\mathcal{C}(\mathbf{w}; K_c, \mathbf{h}_c, g_c)$. To obtain the canonical form over variables \mathbf{y} and \mathbf{z} requires re-arranging the scope and reducing it with the evidence. Note that this canonical form describes the CPD of \mathbf{x} , so it is not a realisable Gaussian distribution over \mathbf{w} . The information matrix K' is not invertible, so no covariance form exists. The canonical form of conditional distributions is central to the work discussed in Chapters 7 and 8.

This chapter described the Gaussian distribution representations and operations required for our PGMs and explained the visualisation of these distributions. Since many probabilistic applications require discrete random variables, the following chapter describes discrete distribution concepts and operations necessary for this study

Chapter 3

Discrete Probability Distributions

Discrete RVs are required to describe the measurement model for unknown correspondence, in Section 4.3. It is therefore necessary to describe a representation for discrete RVs and their associated discrete probability distributions. This chapter introduces the standard table form for these distributions and defines the operations discussed in Section 2.1.2 for this representation. The content is mainly derived from Koller and Friedman’s work [3].

3.1. The Table Form

The table form is used to represent discrete probability distributions, with each row relating a unique combination of discrete variable values to a discrete probability. For example, consider two discrete prior distributions $p(a, b)$ and $p(c)$, where a, b and c are binary variables:

a	b	$p(a, b)$
0	0	0.4
0	1	0.2
1	0	0.3
1	1	0.1

c	$p(c)$
0	0.1
1	0.9

A vacuous table form has a probability of one in each row. In this study, an unnormalised probability distribution for a variable x is denoted by $\tilde{p}(x)$. Next, the above discrete distributions will be used to detail the required operations for use in this study’s PGMs.

3.1.1. Operations on the Table Form

The operations are as follows:

1. Scope Extension: A distribution has no reason to prefer certain values not in its scope. Extending $p(a, b)$ with the variable c results in a table with a new row for unique combinations of $\{a, b, c\}$, with equal probabilities for different values of c .
2. Multiplication: The product of two discrete distributions in table form is found by multiplying the rows in one table with the other table’s rows. The product of $p(a, b)$ and $p(c)$ is

a	b	c	$p(a, b) \times p(c)$
0	0	0	0.04
0	0	1	0.36
0	1	0	0.02
0	1	1	0.18
1	0	0	0.03
1	0	1	0.27
1	1	0	0.01
1	1	1	0.09

3. Division: The quotient of two discrete distributions requires that the denominator's variables are a subset of the numerator's and that none of the denominator's probabilities are zero if the numerator's is nonzero. The prior distribution $p(b)$ and the quotient $p(a, b) \div p(b)$ are shown below.

b	$p(b)$
0	0.7
1	0.3

a	b	$p(a, b) \div p(b)$
0	0	0.571
0	1	0.667
1	0	0.429
1	1	0.333

4. Marginalisation: The marginal of $p(a, b)$ over variable b is found by summing the rows with the same a and different b values:

a	$\tilde{p}(a)$
0	0.6
1	0.4

5. Reduction: A discrete table is reduced with evidence by selecting the rows corresponding to the observation. Reducing $p(a, b)$ with the observation $b = 0$ yields:

a	b	$\tilde{p}(a)$
0	0	0.4
1	0	0.3

Discrete probability distributions have a wide application in PGMs. The operations on the table form were defined explicitly, since the complex discussions in Section 6.5 build on them.

Chapter 4

Designing the Mobile Robot Model for the Localisation Problem

This chapter introduces the localisation problem and explains the robot model designed for this study. The model was kept simple to keep the study's focus on the unknown correspondence problem. Many of the definitions are based on the work by Thrun et al. [1].

4.1. Problem Definition

Mobile robot localisation is defined as the process of inferring a mobile robot's pose, relative to a map of its environment. The pose of the robot is its position and orientation. Typically, the robot uses landmark-based measurements and information about its control inputs to do this. As explained in Section 1.1, a robot faces many uncertainties in determining its pose. In probabilistic robotics, a distribution is used to represent the robot's **uncertain** knowledge of its true pose. The robot's belief at time t is represented by $\tilde{p}(\mathbf{x}_t) = p(\mathbf{x}_T | U_T, Z_T)$, where \mathbf{x}_t is the state vector for the robot's pose. The controls and measurements up to termination time T are given by $U_T = \{\mathbf{u}_0, \dots, \mathbf{u}_T\}$ and $\{Z_T = \mathbf{z}_0, \dots, \mathbf{z}_T\}$ respectively. Section 1.1 explains that the accuracy of this belief is vital for a robot to perform its function.

The type of localisation problem is defined by the information available to the robot initially and during motion. If the robot's initial pose is known exactly, or with little uncertainty, it is referred to as local localisation. The uncertainty is 'local' in that it is restricted to the region near the robot's true pose. Here, the robot's belief is usually modelled by a unimodal distribution, like a Gaussian distribution. The more complex global localisation where the initial pose of the robot is completely unknown was not in the scope of this study. We model the scenario where an operator places the robot in a certain known area of the environment, without attempting to place it accurately or measuring its initial pose. The robot is thus given limited knowledge of its starting position. We implemented this by specifying $p(\mathbf{x}_0)$ with a mean nearby the true initial pose and a large covariance matrix.

Robot models are typically defined as discrete-time models because, usually, mea-

measurements are taken and commands are issued in discrete intervals. Also, describing the model's state variables in the continuous-time makes state estimation far more complex and provides minimal advantages. As a result, we used a discrete-time model, where a variable with a t subscript refers to its value at this discrete timestep. The time between subsequent timesteps is referred to as the sampling time.

There are two major sources of uncertainty that cause the true pose of a robot to be hidden from itself. Firstly, the robot's instructed movement from one timestep to the next is not exact. The cause of this uncertainty is due to disturbances or imperfect actuators. Secondly, sensor noise causes inaccurate measurements of the environment. It is typical to choose Gaussian distributions to model these uncertainties for the reasons discussed in Chapter 2. The unknown correspondence problem requires discrete distributions, to model the uncertainty of each measurement's source landmark. These distributions will be modelled in table form, as described in Section 3.1.

We found it helpful to explore the application of PGMs to the unknown correspondence problem in a stepwise fashion. Each step involved a more complicated measurement model, to solve this complex problem modularly. The four progressions of the measurement model are discussed in depth in Section 4.3. We used the motion model described in the following section throughout this study.

4.2. Linear Robot Motion Model

A robot's motion model describes the expected movement of the robot based on the control input and its previous position. This is an essential component of the localisation problem because it defines the state transition PDF $p(\mathbf{x}_t|\mathbf{u}_t, \mathbf{x}_{t-1})$. This CPD describes the distribution of the robot pose \mathbf{x}_t , when the robot executes the motion command \mathbf{u}_t from pose \mathbf{x}_{t-1} . For this study, the robot motion was chosen to be two-dimensional, with no rotational component. The lack of bearing means the robot's pose and position are referred to interchangeably in this study. There were two reasons for this choice. Firstly, the focus of this study is not on modelling realistic robot behaviour. Secondly, using PGMs for the localisation problem allows for easy extension of this motion model to, for example, incorporate bearing. A change in motion model would simply require a change in the state transition PDF, which is easy to implement in PGMs. This can be seen from the ease with which the PGMs are changed to describe different measurement models in Chapter 7. For this study, the robot's pose at timestep t is thus fully described by the vector

$$\mathbf{x}_t = \begin{bmatrix} x_t \\ y_t \end{bmatrix}, \quad (4.1)$$

where x_t and y_t are the current x and y coordinates of the robot, respectively.

The odometry and velocity motion models are the most commonly used planar robot

motion models in industry. Odometry measures the motion of the robot, and then uses these measurements as the mathematical equivalent of control signals. The actual robot input is either unavailable or is ignored. The control vector \mathbf{u}_t is a measurement of the robot’s movement, so the inaccuracies are limited to the measurement noise in the odometry sensors. The velocity motion model receives a velocity instruction as an input to the robot, and uses this as its control vector. In many robots, this instruction is provided by an operator. The velocity model’s inaccuracies are generally much larger, due to its inability to predict the effects of the commands. For example, it is impossible to accurately predict the robot’s pitch, roll and height from its commands and pose if it moves over an uneven surface. In contrast, odometry sensors can measure all modes of the robot’s motion. The velocity motion model can be used for motion planning, but since that is not required in this study, we used the odometry motion model. The control vector is thus

$$\mathbf{u}_t \triangleq \mathbf{x}_t - \mathbf{x}_{t-1} + \boldsymbol{\epsilon}_t = \begin{bmatrix} \bar{x} \\ \bar{y} \end{bmatrix}, \quad (4.2)$$

where \bar{x} and \bar{y} represent the motion that was measured by the odometry sensors over the sampling period. The odometry sensor measurement noise is represented by $\boldsymbol{\epsilon}_t$, a zero-mean Gaussian distributed RV that is quantified by its covariance matrix R_t . It is common practice to model noise in this manner, since there is no prior reason to assume it would be biased in any way. Since odometry uses measurements as controls, this noise introduces the state transition uncertainty. The state transition equation is found by rearranging Equation 4.2 into the general form of

$$\mathbf{x}_t = A_t \mathbf{x}_{t-1} + B_t \mathbf{u}_t + \boldsymbol{\epsilon}_t, \quad (4.3)$$

where A_t and B_t are 2×2 robot model matrices. Linear equations with an added Gaussian RV are referred to as ‘linear Gaussian’ equations. Consequently, the state transition CPD is found by substituting it into the definition of the multivariate normal distribution (Equation 2.2):

$$p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) = \frac{1}{(2\pi|R_t|)^{N/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x}_t - A_t \mathbf{x}_{t-1} - B_t \mathbf{u}_t)^T R_t^{-1} (\mathbf{x}_t - A_t \mathbf{x}_{t-1} - B_t \mathbf{u}_t) \right\}. \quad (4.4)$$

This section discussed the design decisions for this study’s robot motion model. Next, the robot’s measurement model is described.

4.3. Linear Robot Measurement Model

A robot’s measurement model defines the generation of its actual, physical measurements. More formally, it describes $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$, where \mathbf{z}_t is the measurement vector and \mathbf{m} is the

map of the robot's environment. For brevity's sake, the \mathbf{m} argument is omitted from hereon in, since the robot always has access to the map in this study. This CPD provides the likelihood of a measurement, given knowledge of the robot's position and environment.

This section defines four measurement models, each more complex than its predecessor. The first three models were designed purely to provide localisation problems that required sequentially more complicated PGMs to solve. The fourth and final model describes the measurement model for the unknown correspondence problem.

The simplest measurement model takes measurements of the robot's pose directly, without making use of the aforementioned map. This type of measurement is usually obtained with a GPS. We refer to this model as the direct measurement model, used for the 'simple localisation problem'. It is described by:

$$\mathbf{z}_t = C_t \mathbf{x}_t + \boldsymbol{\delta}_t, \quad (4.5)$$

where C_t is the model's 2×2 measurement matrix. $\boldsymbol{\delta}_t$ represents added zero-mean Gaussian-distributed measurement noise. This models the uncertainty of the measurement and is quantified with its covariance matrix Q_t . The measurement PDF can also be found by applying Equation 2.2, this time to the linear Gaussian equation in Equation 4.5:

$$p(\mathbf{z}_t | \mathbf{x}_t) = \frac{1}{(2\pi|Q_t|)^{N/2}} \exp \left\{ -\frac{1}{2} (\mathbf{z}_t - C_t \mathbf{x}_t)^T Q_t^{-1} (\mathbf{z}_t - C_t \mathbf{x}_t) \right\} \quad (4.6)$$

In most situations, the robot cannot sense or measure its pose directly. Fortunately, the robot's environment typically consists of several salient points or 'landmarks' that the robot can use to localise itself. For the purposes of this study, the robot has access to a landmark-based 'map' \mathbf{m} that contains the coordinates of arbitrarily located, static landmarks. The following measurement models all make use of this map.

The first map-based model uses the offset between the robot's position and a single, fixed landmark as the measurement at each timestep. This type of measurement can be obtained by, for example, an infrared distance sensor. This model will be referred to as the single landmark measurement model, and is described by

$$\mathbf{z}_t = C_t(\mathbf{l} - \mathbf{x}_t) + \boldsymbol{\delta}_t = C_t \mathbf{l} - C_t \mathbf{x}_t + \boldsymbol{\delta}_t, \quad (4.7)$$

where \mathbf{l} contains the coordinates of the solitary landmark provided by the map \mathbf{m} . The PDF of this distribution is defined in the same way as before:

$$p(\mathbf{z}_t | \mathbf{x}_t) = \frac{1}{(2\pi|Q_t|)^{N/2}} \exp \left\{ -\frac{1}{2} (\mathbf{z}_t - (C_t \mathbf{l} - C_t \mathbf{x}_t))^T Q_t^{-1} (\mathbf{z}_t - (C_t \mathbf{l} - C_t \mathbf{x}_t)) \right\}. \quad (4.8)$$

The following change in the measurement model does not involve a different measurement equation. The map was updated to include the coordinates of a number of

fixed landmarks and their unique identifiers. Measurements are modelled by evaluating Equation 4.7, using the different coordinates for each landmark as l . Using special imaging sensors, these measurements can be associated with their source landmarks. However, observing all landmarks at all times is an unrealistic expectation. Practical sensors used for robot measurements have a restricted range within which they can detect landmarks. To more accurately depict a practical robot, only landmarks within a limited radius of its position were measured. Handling a number of measurements at each timestep requires the introduction of a RV $\mathbf{z}_{t,k}$ for each measurement, where k is the measurement number. As a result, we have $k \in [0, N - 1]$ for an environment with N mapped landmarks. Since each measurement is known to **correspond** with a certain landmark, this model creates the known correspondence localisation problem.

The problem of unknown correspondence extends the known correspondence problem by introducing an additional uncertainty. We modelled unknown correspondence with a discrete RV for each measurement, representing the identity of its ‘source landmark’. These identities are unique natural numbers. This discrete RV is referred to as a measurement’s association variable. The association variable for a measurement $\mathbf{z}_{t,k}$ is defined as $a_{t,k} \in [1, N]$, for an environment with N mapped landmarks. This study implemented the PGM solution for $N = 3$. This ensured our solution’s computations remained tractable, and was sufficient to achieve our exploratory goal. In principle, there is no limit to the number of landmarks a PGM can tractably handle. This is discussed further in Section 9.1.

Both multiple landmark measurement models were built on two simplifying assumptions. The first is that each landmark can only be measured once at each timestep. The second is that all measurements were true depictions of the robot’s distance to a landmark. These assumptions align with our aim to design a simple PGM for unknown correspondence, that can be extended later.

Three design decisions were made regarding the model matrices. Firstly, the model matrices were chosen to be static in terms of time. The t subscript of the A, B, C, Q and R matrices are thus omitted from this point onward. Secondly, the A, B and C matrices were chosen as identity matrices. This A matrix defines the commonly used constant position model, which assumes the robot maintains its pose before a control vector is applied. Finally, the R and Q matrices were defined as diagonal matrices, modelling uncorrelated noise. These decisions do not affect this study’s PGM designs.

Throughout this chapter, several design decisions and assumptions were made to simplify the robot’s model. They can easily be replaced by other models for a specific robot. The linear nature of the designed models means that there are no approximations involved. Thus, it is easier to attribute non-ideal results of inference to specific causes, rather than the robot model. However, this robot model was sufficient for the purposes of this project, and was utilised for the Kalman filter in the next chapter, and for the PGMs in Chapters 7 and 8.

Chapter 5

The Kalman Filter

This chapter explains the Kalman Filter (KF) algorithm, applies it to the localisation problem with the direct measurement model and briefly discusses the results. The algorithm explanation is based on the work of Thrun et al. [1], while the rest of the content is unique.

5.1. Technique Overview

A Bayes filter is a general probabilistic, recursive algorithm for estimating an unknown or uncertain PDF using predictions from a mathematical model, adjusted by their corresponding measurements. The Kalman filter (KF) is a simple Bayes filter for estimating continuous states in linear Gaussian systems. In this study, the general KF algorithm was applied to the simplest localisation problem to estimate the pose vector \mathbf{x}_t . The motion and measurement models used in this KF algorithm are given by $\mathbf{x}_t = A_t \mathbf{x}_{t-1} + B_t \mathbf{u}_t + \boldsymbol{\epsilon}_t$ and $\mathbf{z}_t = C \mathbf{x}_t + \boldsymbol{\delta}_t$, respectively. We implemented the KF to verify the PGM in Section 7.1.2.

The KF can only provide state estimations for linear Gaussian systems. Sections 4.2 and 4.3 show that both the model equations shown above are linear Gaussian. This requirement is thus satisfied. Additionally, the prior belief $p(\mathbf{x}_0)$ must be a Gaussian distribution. The covariance form of the prior belief is defined in Section 4.1. The belief computed by the KF is $\tilde{p}(\mathbf{x}_t) = p(\mathbf{x}_t | U_t, Z_t)$, where the controls and measurements up to time t are given by $U_t = \{\mathbf{u}_0, \dots, \mathbf{u}_t\}$ and $\{Z_t = \mathbf{z}_0, \dots, \mathbf{z}_t\}$, respectively. The algorithm for this belief, Algorithm 5.1, is explained in the next section. Note that the typical t subscripts of the model matrices are omitted, since they are static.

5.2. Algorithm Explanation

KFs use the current and past control vectors U_t and measurements Z_t , and the belief at timestep $t - 1$ to calculate the belief at timestep t . Lines 1 and 2 compute the belief at t before the measurement is incorporated. This is called the prediction step, resulting in the predicted belief $p(\mathbf{x}_t | U_t, Z_{t-1}) = \mathcal{N}(\bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$. To obtain $\bar{\boldsymbol{\mu}}_t$, the old mean is shifted using a deterministic version of the state transition equation. Next, $\bar{\boldsymbol{\Sigma}}_t$ is calculated

Algorithm 5.1: Kalman filter algorithm for robot localisation using linear Gaussian state transition and measurement equations, as given by S. Thrun et al. [1].

1. $\bar{\boldsymbol{\mu}}_t = A\boldsymbol{\mu}_{t-1} + B\mathbf{u}_t$
 2. $\bar{\Sigma}_t = A\Sigma_{t-1}A^T + R$
 3. $K_t = \bar{\Sigma}_t C^T (C\bar{\Sigma}_t C^T + Q)^{-1}$
 4. $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + K_t(\mathbf{z}_t - C\bar{\boldsymbol{\mu}}_t)$
 5. $\Sigma_t = (I - K_t C)\bar{\Sigma}_t$
 6. return $\boldsymbol{\mu}_t, \Sigma_t$
-

by pre- and post-multiplying the old covariance matrix with the state transition matrix A and adding state transition noise with R . The updated belief is then calculated in the measurement update step in lines 3 to 5. Line 3 calculates the Kalman gain K_t , representing the measurement's weight in determining the updated belief. The predicted mean and covariance is modified by the measurement using this gain, in lines 4 and 5. The updated belief is therefore $p(\mathbf{x}_t|U_t, Z_t) = \mathcal{N}(\boldsymbol{\mu}_t, \Sigma_t)$. Figure 5.1 shows the correct operation of the KF for localisation. The robot's movement was simulated by implementing the motion model in Python, starting at position $\{1,1\}$ and moving upwards and to the right. Measurements were generated at every timestep $t \in [0, 4]$, using the direct measurement model. The robot's belief is shown with confidence ellipses as described in Section 2.1.1.

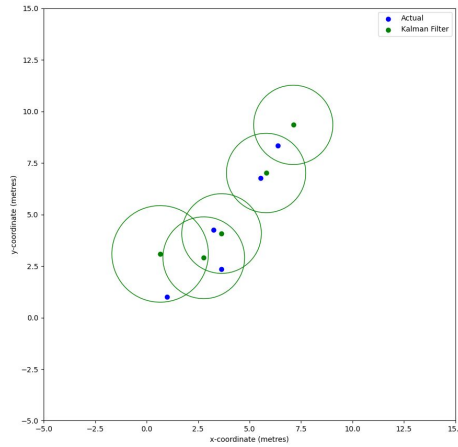


Figure 5.1: The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{6.4, 8.3\}$), with the corresponding KF beliefs using the direct measurement model.

The KF's initial belief has substantial uncertainty due to the prior belief's large covariance matrix. The size of the confidence ellipses reduce as the robot collects more measurements and control vectors, stabilising at a point due to the static model matrices.

Using the KF with more complex measurement models requires its adaptation to the extended Kalman filter (EKF). Since the focus of this study was on the application of PGMs to the localisation problem, this was not explored. This chapter's KF computes a rational pose belief for the simple localisation problem. It is used to verify the first localisation PGM so the extensions of this PGM are known to come from a correct base.

Chapter 6

Probabilistic Graphical Models

This chapter introduces this study’s primary mechanism of solving the localisation problem – probabilistic graphical models (PGMs). We give an overview of how PGMs are used for probabilistic inference and define the basic equations and algorithms necessary for Chapters 7 and 8. The theory of this chapter is based on the work by Barber [6] and Koller and Friedman [3], and its structure was adapted from Schoeman’s work [5].

6.1. Technique Outline

Probability theory principles and operations like Bayes’ theorem and marginalisation can be used to calculate a belief distribution. However, this is not feasible for complex problems. A graphical model uses a graph’s structure to represent the relationships between – and uncertainties about – RVs. This enables us to solve problems with probability theory more easily. A graph details how events are linked using a collection of nodes and connecting edges. Edges in a graph can be directed or undirected. A directed edge is shown with an arrow. A directed graph contains only directed edges. A Bayesian network (BN) is a directed graph. An undirected graph contains only undirected edges. An example of a type of undirected graph is a Markov network (MN). No undirected graphs were required, since the robot localisation problem is specified with causal relationships that are more naturally modelled with BNs. Since the operations for inference remain the same for these undirected links, Markov networks will not be discussed in detail.

Using a PGM involves a two-step procedure. First, a probabilistic problem is modelled with a network of its RVs. Then, given some evidence, inference can be performed using an inference algorithm. This computes the model’s probabilistic knowledge about certain RVs, or its ‘belief’. The next section details the process of modelling a problem with a BN.

6.2. Bayesian Networks

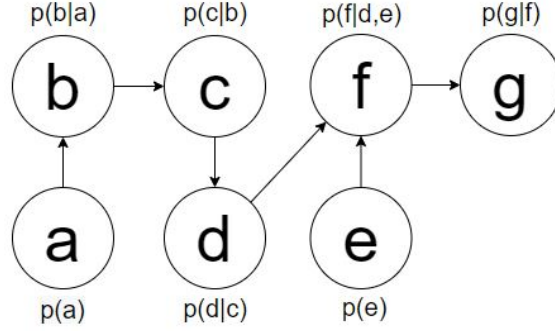


Figure 6.1: Example of a Bayesian network (node CPDs indicated).

Bayesian networks (BNs) are a subset of directed graphical models. In these networks, each node represents a RV. It is useful to model causal relationships between RVs with these directed edges. A node is termed a ‘parent’ of another node if there is a directed edge from the former to the latter. Figure 6.1 shows an example of a BN. Nodes in a BN each possess a conditional probability distribution (CPD) based on their edges. An RV x ’s CPD is defined as $p(x|\text{parents}(x))$, where ‘parents(x)’ denotes the nodes that are parents to x ’s node.

A ‘factor’ is a key concept for BNs and PGMs as a whole. A factor is defined as a function of random variables, termed its ‘scope’. A factor i will be denoted by ϕ_i throughout this report. The BN node CPDs are typically described as factors, such that:

$$\phi_i(x, \text{par}(x)) \triangleq p(x|\text{parents}(x)) , \quad (6.1)$$

where its scope is

$$\text{Scope}[\phi_i] \triangleq \{x, \text{parents}(x)\} . \quad (6.2)$$

The BN chain rule defines that the joint distribution of the random variables in the network is the product of each nodes’ CPD. For the example in Figure 6.1, the joint probability distribution is

$$p(a, b, c, d, e, f, g) = p(a) p(b|a) p(c|b) p(d|c) p(e) p(f|d, e) p(g|f) , \quad (6.3)$$

assuming that all the variables are continuous. Integrating over some of the variables in the joint distribution yields the marginal distribution. For example, $\tilde{p}(d, e, f, g)$ is calculated as:

$$\tilde{p}(d, e, f, g) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(a, b, c, d, e, f, g) da db dc . \quad (6.4)$$

If the variables are discrete, the integration is replaced with summation.

Once a system has been modelled with a BN, it must be converted to a cluster graph. The construction of cluster graphs will be discussed next.

6.3. Cluster Graphs

A cluster graph is an undirected graph where each node represents a subset of the variables in the source network. These subsets are called clusters. If a BN contains the variables x_1, \dots, x_N , a cluster C_i is defined as:

$$C_i \subseteq \{x_1, \dots, x_N\}. \quad (6.5)$$

The subset of shared variables between clusters C_i and C_j is defined as the sepset $S_{i,j}$:

$$S_{i,j} \subseteq C_i \cap C_j. \quad (6.6)$$

The sepset of two clusters is indicated on the edge between these clusters, indicating the sharing of information about the contained variables.

According to Koller and Friedman [3], a cluster graph must satisfy the following two properties:

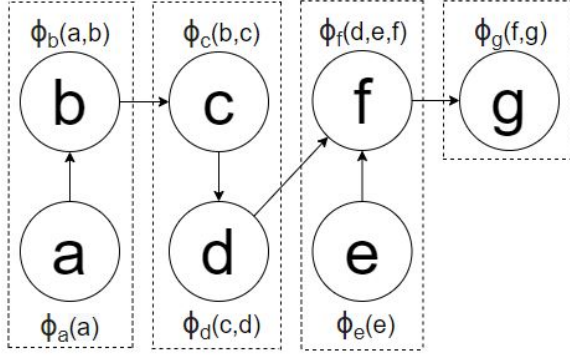
1. The family preservation property: All factors in the BN ϕ_n must be assigned to a cluster C_i with $\text{Scope}[\phi_n] \subseteq C_i$. This means a factor can only be assigned to a cluster if the factor's scope is entirely encapsulated by the cluster's scope.
2. The running intersection property (RIP): There must be a unique path between C_i and C_j in which all clusters and sepsets contain x , for each pair of clusters C_i, C_j where $x \in C_i \cap C_j$.

These properties are not strict enough to create a unique solution for each BN. It is thus possible to define multiple 'legal' cluster graphs for the same BN.

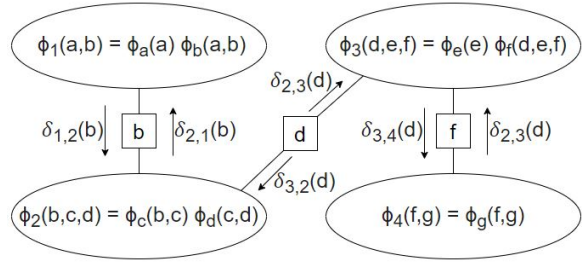
The cluster C_i 's potential $\phi_i(C_i)$ is defined as the product of the factors ϕ_k that it consists of, giving

$$\phi_i(C_i) \triangleq \prod_k \phi_k(x_k, \text{par}(x_k)). \quad (6.7)$$

The BN in Figure 6.1 is shown in Figure 6.2a, with factors instead of CPDs and with the factors grouped into clusters. As previously mentioned, this is one of many possible correct cluster groupings. This can easily be converted to a cluster graph, as shown in Figure 6.2b. Each cluster only contains factors whose scope it encapsulates. Each sepset has been mapped with a single connection between the clusters sharing that variable. For instance, factor $\phi_a(a)$ is contained by cluster $C_1(a, b)$, and cluster $C_3(d, e, f)$ is connected to $C_2(b, c, d)$ to share information about d . The next section explains the process of sharing information between these clusters, also known as message passing.



(a) Bayesian network in factor form showing cluster boundaries



(b) Cluster graph showing cluster potentials, messages and message sepsets

6.4. Message Passing

As explained earlier, a cluster graph is used to facilitate the process of estimating the values of the RVs, a procedure called inference. The graph does this by enabling clusters to share their beliefs about common variables. For example, in Figure 6.2b, C_1 and C_2 share their beliefs about their sepset, variable b . This information communication is called message passing. For this study, the belief propagation message passing algorithm was used.

As shown in Figure 6.2b, each sepset $S_{i,j}$ is contained in two messages between C_i and C_j . There is a message from C_i to C_j , $\delta_{i,j}$, and a message in the opposite direction, $\delta_{j,i}$. To understand how messages are calculated, it is helpful to consider the concept of variable elimination. Variable elimination (VE) is an inference algorithm. It obtains a marginal distribution of a subset of variables from a joint probability distribution. It is calculated by integrating out the unwanted variables, as in Equation 6.4. Consider the following marginal, for Figure 6.2b's graph:

$$p(f, g) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(a, b, c, d, e, f, g) da db dc dd de. \quad (6.8)$$

This expression is simplified using factor notation, the chain rule for BNs (Equation 6.3), and by omitting the range of integration. Rewriting the factors as per their definition in Figure 6.2b, the simplified marginal is:

$$p(f, g) = \int \int \int \int \int \phi_1(a, b) \phi_2(b, c, d) \phi_3(d, e, f) \phi_4(f, g) da db dc dd de. \quad (6.9)$$

However, VE is a standalone inference algorithm. The belief propagation (BP) algorithm is used to calculate the messages in a cluster graph. Using a specific VE ordering and defining certain quantities, we can define the BP algorithm for tree-structured graphs.

A message is calculated by finding the marginal over the message's sepset. To do this, one must integrate out the unwanted variables in the combined scope of the message's

source factor and its incoming messages. The messages from C_1 through to C_4 require the order of variable elimination to move through the cluster graph in the direction of the messages. Therefore, the order is first a , then b and c , and finally d and e . To define the messages, the joint marginal of the variables at the end of the BN chain is found:

$$\begin{aligned}
p(f, g) &= \int \int \int \int \phi_2(b, c, d) \phi_3(d, e, f) \phi_4(f, g) \left[\int \phi_1(a, b) da \right] db dc dd de \\
&= \int \int \int \int \phi_2(b, c, d) \phi_3(d, e, f) \phi_4(f, g) \delta_{1,2}(b) db dc dd de \\
&= \int \int \phi_3(d, e, f) \phi_4(f, g) \left[\int \int \phi_2(b, c, d) \delta_{1,2}(b) db dc \right] dd de \\
&= \int \int \phi_3(d, e, f) \phi_4(f, g) \delta_{2,3}(d) dd de \\
&= \phi_4(f, g) \int_d \int_e \phi_3(d, e, f) \delta_{2,3}(d) dd de = \phi_4(f, g) \delta_{3,4}(f) .
\end{aligned} \tag{6.10}$$

Finding the marginal $p(a)$ starting from C_4 defines the remaining messages as

$$\begin{aligned}
\delta_{1,2}(b) &= \int \phi_1(a, b) da \\
\delta_{2,3}(d) &= \int \int_c \phi_2(b, c, d) \delta_{1,2}(b) db dc \\
\delta_{3,4}(f) &= \int \int \phi_3(d, e, f) \delta_{2,3}(d) dd de \\
\delta_{4,3}(f) &= \int \phi_4(f, g) dg \\
\delta_{3,2}(d) &= \int \int \phi_3(d, e, f) \delta_{4,3}(f) de df \\
\delta_{2,1}(b) &= \int \int \phi_2(b, c, d) \delta_{3,2}(d) dc dd .
\end{aligned} \tag{6.11}$$

The BP algorithm is defined as performing the VE algorithm in this manner. From this example, the general equation for a message $\delta_{i,j}$ can be described in two steps. First, the product of the cluster potential $\phi_i(C_i)$ and all messages into the cluster $\delta_{k,i}$, where $k \neq j$, must be computed. This quantity should then be integrated over the variables that are not in the sepset $S_{i,j}$. It is described formally by Koller and Friedman [3] as:

$$\delta_{i,j}(S_{i,j}) = \int_{C_i - S_{i,j}} \phi_i(C_i) \prod_{k \neq j} \delta_{k,i}(S_{k,i}) , \tag{6.12}$$

where differential elements of the integration variables were omitted for conciseness.

The formulae in this chapter have been described for continuous RVs, but can easily be appropriated for discrete variables. Replacing integrations with summations defines a discrete factor C_i 's outgoing messages as

$$\delta_{i,j}(S_{i,j}) = \sum_{C_i - S_{i,j}} \phi_i(C_i) \prod_{k \neq j} \delta_{k,i}(S_{k,i}) . \tag{6.13}$$

This chapter's formulae do not necessarily produce normalised discrete probability distributions – only their factor representations. This does not matter though, since it is the relative values of the factors at different RV values that affect the results.

As stated above, the message $\delta_{j,i}$ is excluded in the calculation of $\delta_{i,j}$. This can be understood if one views the message $\delta_{i,j}$ as the belief that the clusters on the side of C_i of the graph have about the sepset $S_{i,j}$. Incorporating $\delta_{j,i}$ would mean that either side of the graph would confirm the other's belief about the sepset continually, causing an over-certain belief.

Calculating the joint distribution of a cluster reveals the PGM's belief about that cluster's scope. This distribution is calculated by normalising the product of the incoming messages and the cluster's potential. If a graph is given evidence Z , a cluster C_i 's belief is defined as

$$\tilde{p}(C_i|Z) \triangleq \phi_i(C_i) \prod_k \delta_{k,i}(S_{k,i}) . \quad (6.14)$$

Often, cluster graphs do not form a chain of clusters as in Figure 6.2b. A more generic form of cluster graphs are tree graphs, where each cluster can be connected to many other clusters. An example is shown in Figure 6.3.

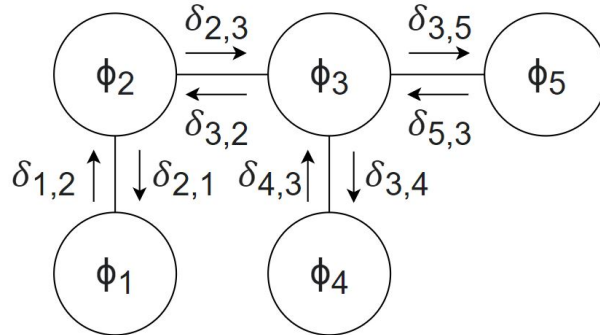


Figure 6.3: Example of a tree cluster graph.

The function that calculates messages in a tree is called the sum-product algorithm. This name can be explained by inspecting Equation 6.12. The function performs an integration (sum) over a product of factors and messages.

A leaf cluster C_i is a cluster with only one connected cluster C_j . For the example in Figure 6.3, the leaf clusters are ϕ_1 , ϕ_4 and ϕ_5 , with outgoing leaf messages $\delta_{1,2}$, $\delta_{4,3}$ and $\delta_{5,3}$. For these messages, the product operator in Equation 6.12 does not include any elements, since the only messages to these clusters are where $k = j$. The outgoing leaf messages are therefore only dependent on their respective leaf cluster potentials ϕ_i . This is an important observation for two reasons. Firstly, messages leaving leaf clusters are constant. Secondly, these messages can be computed directly. The result is that messages leaving leaf clusters provide a concrete place to begin message passing in tree graphs. Calculating the outgoing leaf messages first guarantees that each message need only be

computed once in a tree graph.

In many cluster graphs, there is insufficient information about incoming messages to start the message passing process. The standard approach is to initialise these incoming messages to a vacuous factor representation for the first iteration of the sum-product algorithm. Then, the message values are updated as they are computed with the sum-product formula.

Once all incoming messages $\delta_{k,i}$ to a cluster C_i are known, the messages leaving that cluster $\delta_{i,j}$ (for all j) can be computed. Therefore, calculating $\delta_{1,2}$ means $\delta_{2,3}$ can be calculated, computing $\delta_{4,3}$ and $\delta_{5,3}$ means $\delta_{3,2}$ can be computed, and so on. Repeating this process will evaluate all the messages in the tree graph. The sum-product algorithm approach can be summarised as: compute the cluster potentials, then calculate the outgoing leaf cluster messages, and finally compute the other messages once their required messages have been calculated. As mentioned earlier, this approach ensures each message in a tree graph only needs to be calculated once before the graph's belief is computed. The belief computed by BP in a tree graph is exact. However, many cluster graphs contain loops that require the loopy belief propagation (LBP) for inference. This algorithm involves the same calculations as for BP, but it requires a number of iterations before the PGM's beliefs converge. Also, when this distribution converges, it is not exact.

As was alluded to in Section 2.1.2, PGMs often have information, or 'evidence', about some of their variables. It is therefore vital to describe the effect this has on message passing. Consider a variable c that has been observed as \hat{c} . All clusters in the graph have access to this information. Consequently, no message should include the variable c in its scope. A message $\delta_{i,j}(a, c)$ from cluster C_i , with a scope of $\{a, b, c\}$, reduced with $c = \hat{c}$ is:

$$\delta_{i,j}(a) = \int_b \phi_i(a, b, c = \hat{c}) \prod_{k \neq j} \delta_{k,i}(S_{k,i}) . \quad (6.15)$$

This shows that evidence is accounted for in message passing by equating the variables to their observed values instead of integrating over them. The discrete message in Equation 6.13 is reduced with evidence in the same way. Many useful PGMs contain both discrete and continuous variables, resulting in both discrete and continuous messages in the same graph. Instead of creating these PGMs from either discrete or continuous networks, they are constructed from what are known as hybrid networks.

6.5. Hybrid Networks

This study's discussion of hybrid networks is limited to conditional linear Gaussian (CLG) networks, as these were sufficient for this application of PGMs. In these networks, all continuous variables are modelled by CLG CPDs. Additionally, no discrete variables have continuous parents.

As implied above, variable elimination algorithms do not conceptually differ between discrete and continuous factors. Discrete factors are represented in table form, and continuous factors in their canonical form. When both discrete and continuous factors exist in the same network, an interface between these factors is required to facilitate their belief communication. An additional representation, the canonical table, is defined for this purpose. The canonical table possesses a row for every unique combination of its discrete variable \mathbf{d} , each with an associated canonical form over its continuous variables \mathbf{x} . More formally, each row in a canonical table has the form of Table 6.1.

$\hat{\mathbf{d}}$	$\mathcal{C}(\mathbf{x}; K_{\hat{\mathbf{d}}}, \mathbf{h}_{\hat{\mathbf{d}}}, g_{\hat{\mathbf{d}}})$
--------------------	--

Table 6.1: Example of one row in a canonical table.

To perform inference on hybrid networks, it is necessary to define the factor operations for canonical tables. The product and quotient of canonical tables are executed on corresponding table rows, as table form multiplication. The multiplication and division of the associated canonical forms are calculated as shown in Equations 2.18 and 2.19. Canonical tables can be reduced with evidence of its discrete or continuous variables. When a discrete variable is observed, the rows inconsistent with the observation are removed. If a continuous RV is observed, every row's canonical form is reduced with this evidence, as in Equation 2.23. The discrete table operations mentioned are given in Section 3.1.1.

The final operation to discuss is marginalisation. For the marginalisation over a continuous variable, the marginal for each canonical form is computed, as in Equation 2.21. At first glance, the marginal over a discrete variable would be computed by summing rows, as in Section 3.1.1. However, this requires the addition of Gaussian distributions, which does not produce another Gaussian distribution, as shown in Figure 2.2a. Consequently, the result cannot be represented in a canonical table. To overcome this hurdle, the weak marginalisation of the sum of Gaussian distributions, described in Section 2.2.2, is calculated. The result is a Gaussian distribution, which is used in the canonical table as usual. Due to the weak marginalisation's limitation discussed previously, discrete variables can only be marginalised out when the associated canonical forms characterise legal Gaussian distributions.

PGMs are used by modelling a problem with a network, creating a cluster graph and then performing inference to compute the model's belief. We use this approach to solve the localisation problem for different measurement models in the next chapter.

Chapter 7

Solving Simpler Localisation Problems with Probabilistic Graphical Models

This chapter explains the design of the solutions to three preliminary localisation problems. Each problem involves an increasingly complex measurement model. The models used are the direct measurement, single landmark, and multiple landmark with known correspondence models described in Section 4.3. The purpose of this chapter is to apply PGMs to specifically solve basic localisation problems, so that the PGM approach to the unknown correspondence problem of Chapter 8 is clear. The new content presented in this chapter is completely original. We implemented the PGMs in this study using the EMDW c++ library. References to components of this library are in *italics*.

7.1. PGM Solution for the Direct Measurement Model

7.1.1. Modelling the Problem

The simplest localisation problem uses the direct measurement model, defined by Equation 4.5. First, a Bayesian network (BN) of the problem's variables and their relationships is constructed. The RVs of interest in this localisation problem are the pose \mathbf{x}_t , control signal \mathbf{u}_t and measurement \mathbf{z}_t , at every timestep. In the software, each timestep's x and y position was assigned an integer identifier, referred to as an *RVID*. The pose \mathbf{x}_t is dependent on the control signal \mathbf{u}_t and the previous timesteps pose \mathbf{x}_{t-1} . Consequently, \mathbf{u}_t and \mathbf{x}_{t-1} are modelled as the parents of \mathbf{x}_t in the network. Additionally, \mathbf{x}_t is the parent of \mathbf{z}_t since each measurement is dependent on the robot's pose at that time.

The above dependencies are the same for all timesteps, except $t = 0$. The initial pose does not have a preceding pose or an associated control vector. However, the prior pose distribution $p(\mathbf{x}_0)$ is known by the robot, as specified in Section 4.1.

Using the principles given in Section 6.2, the BN for all timesteps $t \in [0, T]$ is constructed as shown in Figure 7.1. The variables whose values have been observed are shaded. These observations will be used as evidence to simplify the problem.

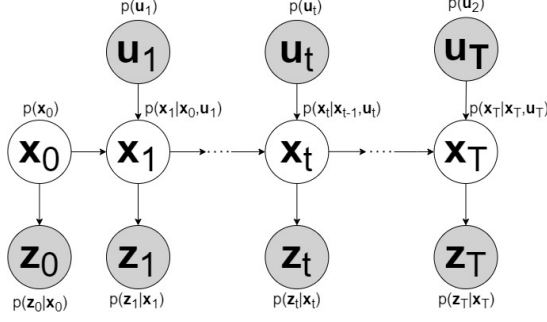


Figure 7.1: Bayes network for the localisation problem using direct measurements (node CPDs indicated).

The second step is to convert this BN to a legal cluster graph, according to the two properties provided in Section 6.3. It is logical to group the variables \mathbf{x}_t , \mathbf{x}_{t-1} and \mathbf{u}_t into a cluster for each t . This groups the variables contained in each state transition PDF, $p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t)$, into the same clusters. These clusters will be referred to as the ‘motion clusters’ since they contain direct information about the robot’s movement. Their potentials are denoted by Φ_{2t+2} , and are calculated with Equation E.1. We did not need to compute any of this study’s cluster potentials by hand, as this was handled by the *loopyBP-CG* method of the EMDW library. They can thus be found in Appendix E.

The measurement variables must still be grouped into clusters. Each measurement \mathbf{z}_t contains information about its associated pose \mathbf{x}_t , so each measurement is placed in its own cluster, with its potential denoted by Ψ instead of Φ to explicitly show the different cluster types. These potentials are calculated with Equation E.4. The above groupings creates the graph shown in Figure 7.2, which obeys the family preservation and running intersection properties. The use of PGMs for probabilistic reasoning is a general technique.

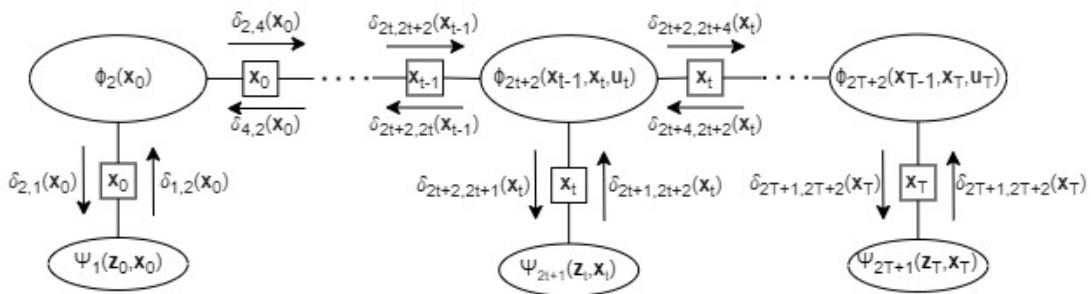


Figure 7.2: Cluster graph for the initial localisation problem, showing cluster numbering and messages.

It is thus crucial that the factors provided to the PGM are in a standard format. The continuous distributions must therefore be represented in their canonical form. The first step to obtaining the state transition’s canonical form is computing the covariance form using the state transition equation. The mean is calculated using the expectation operator

from general probability theory:

$$\begin{aligned}\mathbb{E}\{\mathbf{x}_t\} &= \mathbb{E}\{A\mathbf{x}_{t-1} + B\mathbf{u}_t + \boldsymbol{\epsilon}_t\} = \mathbb{E}\{A\mathbf{x}_{t-1} + B\mathbf{u}_t\} \quad (\text{since } \mathbb{E}\{\boldsymbol{\epsilon}_t\} = \mathbf{0}) \\ &= A\mathbf{x}_{t-1} + B\mathbf{u}_t \quad (\text{since } \mathbf{x}_{t-1} \text{ and } \mathbf{u}_t \text{ are known}).\end{aligned}\tag{7.1}$$

The covariance matrix is obtained using its statistical definition. Using the result of Equation 7.1, the covariance is calculated as follows:

$$\begin{aligned}\text{cov}\{\mathbf{x}_t\} &= \mathbb{E}\{(\mathbf{x}_t - \mathbb{E}\{\mathbf{x}_t\})(\mathbf{x}_t - \mathbb{E}\{\mathbf{x}_t\})^T\} \\ &= \mathbb{E}\{(A\mathbf{x}_{t-1} + B\mathbf{u}_t + \boldsymbol{\epsilon}_t - A\mathbf{x}_{t-1} - B\mathbf{u}_t)(A\mathbf{x}_{t-1} + B\mathbf{u}_t + \boldsymbol{\epsilon}_t - A\mathbf{x}_{t-1} - B\mathbf{u}_t)^T\} \\ &= \mathbb{E}\{\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t^T\} = R \quad (\text{since } \mathbb{E}\{\boldsymbol{\epsilon}_t\} = \mathbf{0}).\end{aligned}\tag{7.2}$$

The state transition CPD's covariance form is thus $\mathcal{N}(\mathbf{x}_t; A\mathbf{x}_{t-1} + B\mathbf{u}_t, R)$. The canonical representation of this distribution is obtained by following the procedure in Section 2.3. This result is simplified by using the known control vectors. Providing the evidence that $\mathbf{u}_t = \hat{\mathbf{u}}_t$, and using Equation 2.23, the factor is reduced to

$$\begin{aligned}p(\mathbf{x}_t | \mathbf{x}_{t-1}, \hat{\mathbf{u}}_t) &= \mathcal{C}\left(\begin{bmatrix} \mathbf{x}_t & \mathbf{x}_{t-1} \end{bmatrix}^T; K'_s, \mathbf{h}'_s, g'_s\right), \text{ where} \\ K'_s &= \begin{bmatrix} R^{-1} & -R^{-1}A \\ -A^T R^{-1} & A^T R^{-1} \end{bmatrix}, \mathbf{h}'_s = \begin{bmatrix} R^{-1}B \\ -A^T R^{-1}B \end{bmatrix}, \text{ and} \\ g'_s &= -\ln[(2\pi|R|)^{N/2}] - \frac{1}{2}\hat{\mathbf{u}}_t^T B^T R^{-1} B \hat{\mathbf{u}}_t.\end{aligned}\tag{7.3}$$

The incorporation of $\hat{\mathbf{u}}_t$ as evidence changes the motion cluster potentials, as shown in Equation E.2.

Next, the canonical form of the measurement factors must be determined. Following the procedure shown in Equations 7.1 and 7.2, it is found that $p(\mathbf{z}_t | \mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t; C\mathbf{x}_t, Q)$. Using the process in Section 2.3 again, the canonical representation of this distribution can be derived. The square root form of this representation is derived in Appendix D to demonstrate how the numerical stability of its implementation can be improved. Instead, we reduced the canonical form, and implemented **its** square root form. Given $\mathbf{z}_t = \hat{\mathbf{z}}_t$, the reduced representation is found by applying Equation 2.23. The result is that

$$\begin{aligned}p(\mathbf{z}_t = \hat{\mathbf{z}}_t | \mathbf{x}_t) &= \mathcal{C}(\mathbf{x}_t; K'_m, \mathbf{h}'_m, g'_m), \text{ where} \\ K'_m &= C^T Q^{-1} C, \mathbf{h}'_m = Q^{-1} C \hat{\mathbf{z}}_t, \text{ and } g'_m = -\ln[(2\pi|Q|)^{N/2}] - \frac{1}{2}\hat{\mathbf{z}}_t^T Q^{-1} \hat{\mathbf{z}}_t.\end{aligned}\tag{7.4}$$

The redefined measurement cluster potentials are given in Equation E.5. We implemented the above reduced canonical forms in our code using their square root form, as described in Section 2.1.3, with *SqrtMVG factor* pointers.

This section described the direct measurement localisation problem with a cluster

graph, and defined the cluster potentials. This cluster graph was created in our software by passing a vector of *SqrtMVG factor* pointers to a *ClusterGraph* object constructor. Next, message passing for inference is performed and the results are compared to those of the KF for the same model.

7.1.2. Inference and Results

The graph in Figure 7.2 clearly possesses a tree structure, where the measurement clusters Ψ_{2t+1} form the leaf clusters. Consequently, messages in this cluster graph only need to be computed once for exact inference. The outgoing leaf cluster messages $\delta_{2t+1,2t+2}$ and $\delta_{2,4}$ are calculated first, since they do not incorporate any incoming messages. Next, the messages from left to right between the motion clusters, $\delta_{2t+2,2t+4}$, can be computed. The evidence regarding the control vectors and measurements is used to reduce these messages into the form given by Equation 6.15. This equation is used to calculate all the messages in a process summarised as follows:

1. Compute the outgoing measurement cluster messages, using the measurements $\hat{\mathbf{z}}_t$. Since \mathbf{z}_t is observed and \mathbf{x}_t is the sepset, no marginalisation need be performed.
2. Compute the message from the initial pose cluster C_1 to the first motion cluster C_2
3. Starting at C_2 , compute the messages from left to right, using each control vector $\hat{\mathbf{u}}_t$.

The graph's messages that are directed from bottom to top and left to right have been computed. This passes information about the robot's pose 'forward' in time. Thus far, the PGM approach is equivalent to the Kalman filter described in Section 5.2, since they use the same model. However, only half of the inference on this PGM has been completed. The remaining half of the graph's messages can still be computed. These messages, going from right to left and top to bottom, carry information 'backwards' in time. The message from the termination timestep T is calculated first, omitting $\delta_{2T+4,2T+2}$ since it does not exist. Thereafter, each message $\delta_{2t+2,2t}$ is computed once its associated $\delta_{2t+4,2t+2}$ has been calculated.

It is worthwhile to provide an intuitive explanation for the usefulness of passing information 'backwards' in time. Consider a mobile robot that takes a measurement \mathbf{z}_0 of its position \mathbf{x}_0 , and then moves to position \mathbf{x}_1 . It then measures its position again, generating \mathbf{z}_1 . The control vector \mathbf{u}_1 applied to move the robot from the first timestep to the second is known. From this it is obvious that the measurement \mathbf{z}_1 is used as additional information about \mathbf{x}_0 . The ability to pass information 'backwards' in time is therefore extremely useful. We calculated the messages in code by calling the *loopyBP-CG()* method on the *ClusterGraph* object.

For this problem, the messages from the motion clusters down to their measurement clusters, $\delta_{2t+2,2t+1}$, do not need to be evaluated. The RV \mathbf{z}_t is observed, so computing this

message will not reveal any new information about this variable and \mathbf{x}_t is included in the motion clusters. Once message passing between the motion clusters has been completed, the PGM has already established its beliefs about this variable. Passing these messages into the measurement clusters reveals the same beliefs for \mathbf{x}_t .

The final step in the inference process is extracting the robot's belief $\tilde{p}(\mathbf{x}_t)$ from each motion cluster. This is done by multiplying each motion cluster potential by its incoming messages and performing marginalisation over \mathbf{x}_{t-1} , giving

$$\tilde{p}(\mathbf{x}_t) = \int_{\mathbf{x}_{t-1}} \Phi_{2t+2}(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}_t = \hat{\mathbf{u}}_t) \delta_{2t, 2t+2}(\mathbf{x}_{t-1}) \delta_{2t+1, 2t+2}(\mathbf{x}_t) \delta_{2t+4, 2t+2}(\mathbf{x}_t) d\mathbf{x}_{t-1}. \quad (7.5)$$

This yields a Gaussian distribution in its canonical form for each pose. To calculate the confidence ellipses, the beliefs must be in covariance form. Equations 2.13 through 2.14 is used for this purpose. We computed the PGM's beliefs of all the robot's positions by calling the *queryLBP_CG()* method on the *ClusterGraph* object for each pose component's *RVID*. The covariance form is retrieved from the returned *factor* pointer by typecasting it to a *SqrtMVG* pointer. The same procedure is used for the other PGMs.

To visualise the advantage of using PGMs for localisation, the beliefs computed by the Kalman filter of Chapter 5 are compared with this PGM's beliefs in Figure 7.3. The robot's motion was simulated over five timesteps, collecting direct measurements while moving towards the top right of the environment. Initially, there is little information

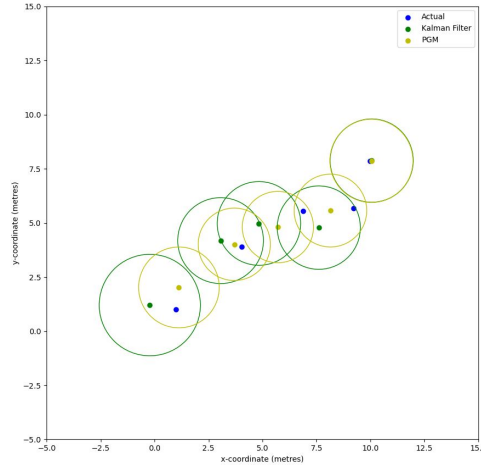


Figure 7.3: The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{10,8\}$), with the corresponding PGM beliefs using the direct measurement model.

about the robot's pose, resulting in an uncertain belief. This is due to the large covariance matrix of the $p(\mathbf{x}_0)$ distribution, and the lack of a prior timestep. Over time, the robot gains more information about its position through the measurements, generally gaining certainty in its beliefs. Note that the PGM's beliefs are more certain and more accurate on average. This is because the PGM makes use of the full controls and measurements for all execution time $t \in [0, T]$, as defined in Section 4.1. Therefore, the PGM makes

better use of the information available to the robot. In the final timestep, there is no information that can be passed ‘backwards’ in time. The PGM’s belief matches the KF’s here – verifying this PGM’s correctness.

This section explained the PGM solution to the localisation problem for the direct measurement model. Although this is a simple model, it provides a good starting point for describing the single landmark localisation problem discussed in the next section.

7.2. PGM Solution for the Single Landmark Measurement Model

This localisation problem uses the single landmark measurement model. The PGM for this problem was a simple extension of the previous solution’s model, involving only a change in the measurement CPD.

7.2.1. Modelling the Problem

The structure of this problem’s BN is the same as Figure 7.1, since the same variables and relationships hold. However, the measurement CPD has been modified. Using Equation 4.7, the new covariance form for the measurement distribution is derived in the same fashion as in Equations 7.1 and 7.2, giving $p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t; C\mathbf{l} - C\mathbf{x}_t, Q)$. The reduced factor’s canonical form is determined in the same way as before, giving

$$p(\mathbf{z}_t = \hat{\mathbf{z}}_t|\mathbf{x}_t) = \mathcal{C}(\mathbf{x}_t, K'_l, \mathbf{h}'_l, g'_l), \text{ where} \quad (7.6)$$

$$K'_l = C^T Q^{-1} C, \mathbf{h}'_l = Q^{-1} C\mathbf{l} - Q^{-1} C\hat{\mathbf{z}}_t, \text{ and} \quad (7.7)$$

$$g'_l = -\ln[(2\pi|Q|)^{N/2}] - \frac{1}{2}\hat{\mathbf{z}}_t^T Q^{-1} \hat{\mathbf{z}}_t - \frac{1}{2}C\mathbf{l}^T Q^{-1} C\mathbf{l}. \quad (7.8)$$

The resulting cluster graph is the same as for the previous problem, shown in Figure 7.2. The changed measurement CPD resulted in a different cluster potential, as shown in Equation E.6. To implement this model change in our software, we simply provided the *ClusterGraph* object with a *SqrtMVG* pointer representing the new measurement CPD.

7.2.2. Inference and Results

The message passing and belief extraction for this PGM is equivalent to the procedure discussed in Section 7.1.2. These beliefs were very similar in nature to those of the previous PGM. A visualisation can be seen in Figure C.1 in Appendix C.

The development of this problem was valuable because it generalised the PGM to make use of landmark measurements. The measurement CPD’s canonical form specified

in Equation 7.6 is used for all further PGMs of this study. The final problem addressed in this chapter provides the robot with a map of multiple landmarks.

7.3. PGM Solution for the Multiple Landmark Measurement Model with Known Correspondence

For this problem, landmarks within a small range of the robot's true position were measured at every timestep, with known correspondence. This problem makes use of the same measurement CPD as the previous problem, but introduces the complexity of having more than one measurement at each timestep.

7.3.1. Modelling the Problem

This problem's BN extends the first problem's network to include more than one measurement variable per timestep. The new network is shown in Figure 7.4. To demonstrate the network's structure explicitly, this figure shows the resulting nodes if three landmarks are measured at timestep zero. Each measurement node for $z_{t,k}$ results in a separate

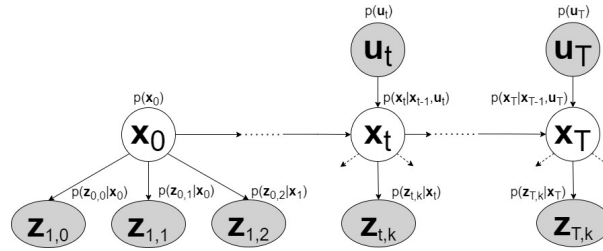


Figure 7.4: Bayesian network for localisation using multiple measurements with known correspondence.

measurement factor. The CPD of these factors is still defined by Equation 7.6. Each factor's reduced scope is just x_t , so we grouped the factors for the same timestep into a measurement cluster C_{2t+1} . This is logical since these factors have identical sepsets with the same motion cluster C_{2t+2} .

The variable k_t represents the total number of measurements at time t . In this measurement model, it is possible that there are no landmarks within the robot's measurement range. This will result in no measurement variables $z_{t,k}$ and consequently no measurement cluster C_{2t+1} . For any t where $k_t > 0$, the structure of both the BN and the cluster graph will mirror the network in 7.1 and the graph in Figure 7.2 respectively. We implemented this model the same way as before, apart from the addition of more measurement *SqrtMVG* factor pointers to the *ClusterGraph* object.

7.3.2. Inference and Results

Performing inference on this graph is mostly the same as for the previous graph. The only new development is that cluster C_{2t+1} may not be defined. The messages to and from this cluster are simply omitted in this case. Our implementation's inference process was also the same as prior, apart from the above exception.

The robot's motion was simulated over five timesteps, collecting multiple landmark measurements while moving towards the top right of the environment. In general, the PGM's beliefs are very accurate and certain for the known correspondence problem. This is due to the large amount of information available to the model. Numerous measurements enable the PGM to 'average out' the measurement noise, reducing its effect on the beliefs. The ellipses follow the same general trend as before. However, the measurement range implemented causes the uncertainty and error to vary depending on how many landmarks can be measured. In general, this measurement model results in more accurate and more certain belief computation than the previous models, depending on the number of measurements taken. This simulation's output can be seen in Figure C.2, which is not shown here due to its similarity to the results shown in Section 8.3.

In general, this measurement model results in more accurate and more certain belief computation than the previous models, depending on the number of measurements taken.

In conclusion, the PGM approach is clearly a valuable technique for the mobile robot localisation problem. This approach is summarised as follows. First, create a BN to model the specific problem and use it to construct a cluster graph. Then, collect the measurements with the measurement model and use these measurements and the control vectors to compute the cluster potentials. Finally, perform inference on the model and extract its beliefs about the robot's position from the motion clusters. By using PGMs to solve three preliminary localisation problems, this chapter has laid the foundation for using graphical models to solve the mobile robot localisation problem with unknown correspondence, in the next chapter.

Chapter 8

Solving the Unknown Correspondence Localisation Problem with a Probabilistic Graphical Model

This chapter describes the design of the solution to this study’s main problem – mobile robot localisation using measurements with unknown landmark correspondence. It is largely dependent on the content of the previous two chapters. The work presented is entirely unique.

8.1. Modelling the Problem

The definition of this problem in Section 4.3 states that no landmark can generate more than one measurement at a single timestep. This means the association variables at the same timestep cannot share a value, so they are dependent on each other. Since the same timestep’s association variables influence each other equally, it is natural to model their relationships with undirected edges. Each association variable $a_{t,k}$ affects its corresponding measurement $z_{t,k}$ directly, so the association variables are modelled as parents of their respective measurements. The remainder of the RV network is the same as the BN in Figure 7.4. The resultant network for this problem is shown in Figure 8.1. Since the structure of the network changes based on the number of measurements k_t , the network is shown for the case of three measurements at $t = 0$. The inclusion of both continuous and

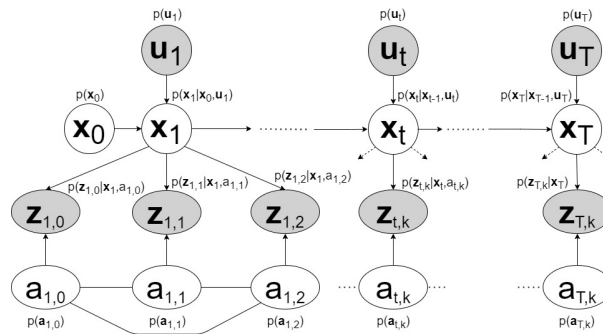


Figure 8.1: Hybrid network for localisation using multiple measurements with unknown correspondence (node CPDs indicated).

$a_{t,k} = f$	$p(\mathbf{z}_{t,k} = \hat{\mathbf{z}}_{t,k} \mathbf{x}_t, a_{t,k} = f)$
---------------	--

Table 8.1: A row in the unknown correspondence PGM’s measurement cluster canonical table.

discrete RVs in this network means this is a hybrid network. Although it is not strictly a BN, this network can still be used to construct a cluster graph in a similar way.

Inspecting the structure of the hybrid network in Figure 8.1 reveals that the association variables form a clique. A clique is defined as a subset of fully connected nodes, where all the edges are undirected. Due to this, the association variables were grouped into an ‘association cluster’, C_{5t+1} , for each t . This is beneficial for two reasons. Firstly, it simplifies the message passing process by vastly reducing the number of messages to be computed. Secondly, no knowledge about each variables prior distribution $p(a_{t,k})$ is known. Using this clustering allows for easy application of the information the PGM does have at hand – these variables’ joint prior PDF. However, this limits the computational tractability of our solution, which is further discussed in Section 9.1. The prior distribution $p'(a_{t,0}, \dots, a_{t,k_t-1})$ has equal likelihoods for each unique combination of association variable values, since no landmark should be favoured prior to the model receiving evidence. Different joint distributions are used for each timestep, based on the number of measurements. Since we used three landmarks, there are three possible joints, which are shown in Appendix F. We implemented this in code by creating a *DiscreteTable* factor pointer. The resulting association variable cluster potentials are given in Appendix E.

In actuality, a number of the measurement clusters and their related messages are not defined due to the varying value number of measurements k_t . To simplify the discussion of the various clusters and messages, the numbering system allocates a number to every **possible** cluster and message. In all the following calculations, the undefined quantities for a specific situation should simply be omitted.

The measurement variables $\mathbf{z}_{t,k}$ still need to be grouped to create the cluster graph for this problem. Before, the measurement factors for each t were grouped into one cluster. Due to the individual association variables, these factors no longer have an overlapping scope after observing measurement $\hat{\mathbf{z}}_{t,k}$. To avoid defining measurement clusters and with very broad scopes, we placed each measurement factor in its own cluster C_{5t+2+k} .

This cluster interfaces between canonical form and discrete table factors, so it must be represented by a canonical table. Each row is of the form shown in Table 8.1, where f is a landmark identifier. The above grouping results in the loopy cluster graph shown in Figure 8.2. The software resources available for this study did not include functionality for hybrid networks. We created separate cluster graphs for the continuous and discrete RVs, and handled the hybrid measurement clusters separately. The graph containing the continuous RVs and continuous messages for all timesteps will be referred to as the ‘continuous cluster graph’. All factors used this graph are in their canonical form. A

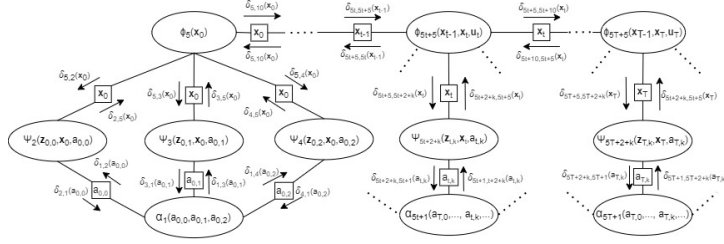


Figure 8.2: Cluster graph for multiple measurements with unknown correspondence.

separate ‘discrete cluster graph’ is created for each timestep, incorporating the association variables and corresponding discrete messages. The factors are in table form. Both the ‘continuous’ and the collection of ‘discrete graphs’ are tree graphs, so the usual belief propagation algorithm is used to perform inference on them.

The hybrid measurement clusters will be used to interface between these networks for information sharing. We implemented this by creating the ‘continuous graph’ with the upwards, continuous messages from the canonical tables included as leaf clusters. The same is done for each ‘discrete graph’, where the corresponding downwards, discrete messages are incorporated as leaf clusters. The broad definition of factors allows for this, as long as the messages are in the appropriate factor form for the graph. The resulting outgoing leaf cluster messages are equal to their cluster potentials, which are the original messages. We used separate *ClusterGraph* objects for the two ‘discrete graphs’. *DiscreteTable* and *SqrtMVG* factor pointers were used for the measurement cluster’s outgoing discrete and continuous messages, respectively. The separated discrete and continuous cluster graphs are shown in Figure 8.3.

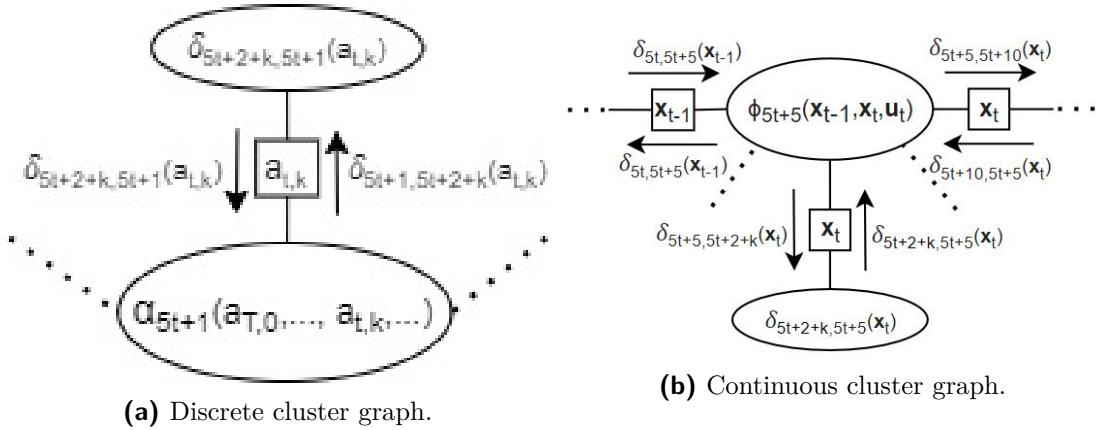


Figure 8.3: The separate (a) discrete and (b) continuous cluster graph structures.

8.2. Inference

The full cluster graph for this problem, shown in Figure 8.2, is a loopy cluster graph. Loopy belief propagation (LBP) must be performed to arrive at a final, non-exact belief, as stated

in Section 6.4. The first step in implementing LBP for this problem is creating the ‘discrete graphs’ for each timestep. The downward, discrete messages in this graph are initialised as vacuous discrete factors in table form. Using these messages, each ‘discrete graph’'s outgoing messages $\delta_{5t+1,5t+2+k}$ is computed. This provides the information necessary to calculate the upwards messages $\delta_{5t+2+k,5t+5}$. These continuous messages are the weak marginalisation of the distributions contained in each hybrid measurement cluster C_{5t+2+k} . After evaluating these messages for all timesteps, the continuous cluster graph is created. The procedure discussed thus far is detailed as follows:

1. Construct the discrete cluster graphs for every timestep where $k_t > 0$. Each graph must incorporate the appropriate association cluster C_{5t+1} and the downwards, discrete messages $\delta_{5t+2+k,5t+1}$.
2. Compute the discrete messages leaving these graphs. Using Equation 6.13, the outgoing messages are given as

$$\delta_{5t+1,5t+2+k}(a_{t,k}) = \sum_{\mathbf{a}_t} \alpha_{5t+1}(a_{t,0}, \dots, a_{t,k_t-1}) \prod_{n \neq k} \delta_{5t+2+n,5t+1}(a_{t,n}), \quad (8.1)$$

where \mathbf{a}_t represents the set of association variables in the association cluster's scope, excluding $a_{t,k}$. The variable n is of the same range as k , namely $n \in [0, k_t - 1]$.

3. Compute the upwards messages leaving the measurement clusters by performing discrete variable marginalisation over each respective $a_{t,k}$. This results in a sum of Gaussian distributions. As discussed in Section 6.5, the weak marginalisation of this sum must be calculated to use it in the PGM. Each weight w_i is equal to the probability of observing $a_{t,k} = i$, which is contained in the message $\delta_{5t+1,5t+2+k}$. This is written as $w_i = \delta_{5t+1,5t+2+k}(a_{t,k} = i)$. The mean $\boldsymbol{\mu}_i$ and covariance matrix Σ_i are obtained from the corresponding row in the measurement cluster's canonical table. Equations 2.24 and 2.25 is used to obtain the weak marginalisation's mean $\boldsymbol{\mu}_w$ and covariance Σ_w . Converting from the the covariance to canonical form with Equations 2.9 to 2.11 gives these messages as $\delta_{5t+2+k,5t+5}(\mathbf{x}_t) = \mathcal{C}(\mathbf{x}_t; K_w, \mathbf{h}_w, g_w)$.
4. Construct the continuous cluster graph with the messages $\delta_{5t+2+k,5t+5}$ and motion clusters C_{5t+5} for all t .
5. Compute the messages within the cluster graph. It can be seen from Figure 8.3b that this graph forms a tree, and is identical in structure to Figure 7.2. The only difference between these graphs is that the measurement cluster potentials of before are replaced with the continuous messages $\delta_{5t+2+k,5t+5}$. Consequently, the message passing within this graph is identical to the procedure described in Section 7.1.2.

The messages from the ‘discrete graph’, $\delta_{5t+1,5t+2+k}$, represent the discrete probability distribution of the association variable $a_{t,k}$. The messages leaving the measurement clusters,

all $\delta_{5t+2+k,5t+5}$, are each measurement cluster's approximate PDF for the pose \mathbf{x}_t . Finally, the messages within the 'continuous graph', $\delta_{5t,5t+5}$ and $\delta_{5t+5,5t+10}$ represent the information sharing between the motion clusters about the robot's pose.

As mentioned briefly in Section 6.5, discrete variable marginalisation in canonical tables can only be performed on valid Gaussian distributions. In this study, the canonical form in each row of the C_{5t+2+k} canonical tables is equivalent to the CPD $p(\mathbf{z}_t = \hat{\mathbf{z}}_t | \mathbf{x}_t)$. If \mathbf{z}_t is observed, Equation 4.8 can be rearranged into a valid Gaussian distribution in terms of \mathbf{x}_t . The weak marginalisation can thus be computed. The beliefs of the 'continuous graph' are extracted in the same way as before, with

$$\tilde{p}(\mathbf{x}_t) \triangleq \int_{\mathbf{x}_{t-1}} \Phi_{5t+5}(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}_t = \hat{\mathbf{u}}_t) \delta_{5t,5t+5}(\mathbf{x}_{t-1}) \delta_{5t+10,5t+5}(\mathbf{x}_t) \prod_{k=0}^{k_t-1} \delta_{5t+2+k,5t+5}(\mathbf{x}_t) d\mathbf{x}_{t-1}. \quad (8.2)$$

However, these are not the final pose beliefs of the PGM. The 'discrete graph' must be updated with these beliefs. The downwards message passing is continued as follows:

6. Compute the messages downwards from the motion clusters, C_{5t+5} , to the measurement clusters C_{5t+2+k} . This involves multiplying the incoming messages from the neighbouring motion and measurement clusters and performing marginalisation over \mathbf{x}_{t-1} on the product.
7. Compute the messages from the measurement clusters downwards. These messages $\delta_{5t+2+k,5t+1}$ are the result of performing the continuous marginalisation of each row's canonical form over \mathbf{x}_t . The resultant discrete table for $a_{t,k}$ has a row for each possible value f that contains the probability that $a_{t,k} = f$.
8. Construct a new 'discrete graph' for each timestep, using the appropriate prior joint distribution $p'(a_{t,0}, \dots, a_{t,k_t-1})$, and the updated downward messages $\delta_{5t+2+k,5t+1}$.

All downward messages $\delta_{5t+5,5t+2+k}$ represent the continuous cluster graph's updated belief about each pose \mathbf{x}_t . The downward messages leaving the measurement clusters, $\delta_{5t+2+k,5t+1}$, contain the updated association variable PDFs. The invalid combinations of association variable values are zeroed by their corresponding joint distribution.

Each 'discrete graph's updated joint belief is extracted by applying Equation 6.14:

$$\tilde{p}(a_{t,0}, \dots, a_{t,k_t-1}) = p(a_{t,0}, \dots, a_{t,k_t-1}) \prod_{k=0}^{k_t-1} \delta_{5t+k+2,5t+1}. \quad (8.3)$$

Again, these are not yet the final beliefs of the PGM.

The definition of messages in Equation 6.12 and the belief in Equation 6.14 only differ by the exclusion of one message. It is thus possible to use the beliefs from Equation 8.3 to

calculate the messages leaving the ‘discrete graph’, $\delta_{5t+1,5t+2+k}$ as

$$\delta_{5t+1,5t+2+k}(a_{t,k}) = \tilde{p}(a_{t,0}, \dots, a_{t,k_{t-1}}) \div \delta_{5t+2+k,5t+1}(a_{t,k}). \quad (8.4)$$

The messages from the ‘continuous graph’, $\delta_{5t+5,5t+2+k}$, can be computed in the same way, using Equation 8.2. We used this method for computing these messages, as it simplified the implementation.

After constructing the updated ‘discrete graphs’, the process outlined in steps 1 through 8 is repeated. The **original** motion cluster potentials and prior joint distributions must be used to create the ‘continuous’ and ‘discrete graphs’ respectively for **iteration** iteration.

After a number of repetitions, the beliefs will converge to the final result of approximate inference using LBP. The next section analyses these beliefs and compares them to the known correspondence PGM’s beliefs.

8.3. Results

The PGM for the known correspondence problem will be referred to as PGM1 for brevity’s sake. The PGM of this chapter will be indicated by PGM2 in this discussion. For all comparisons, both PGMs were provided with the same measurements, but the measurement-landmark correspondence was hidden for PGM2. The simulation was set up so that the robot took measurements at every timestep $t \in [0, 4]$, starting at position $(1, 1)$ and moving towards the top right corner of the environment.

The PGMs were both given the same map, control vectors and initial pose distributions, producing the confidence ellipses shown in Figure 8.4.

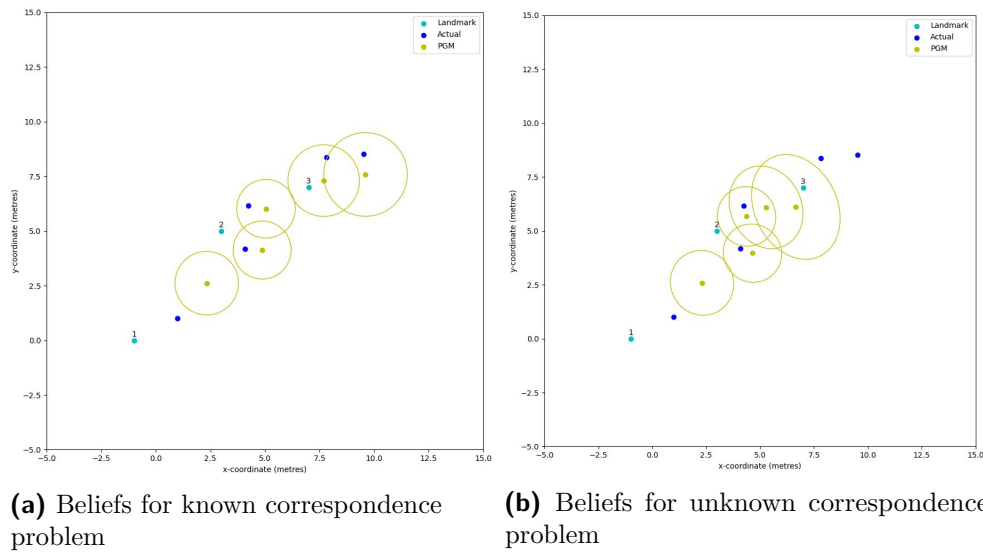


Figure 8.4: The robot’s actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{9.5,8.5\}$), with the corresponding PGM beliefs for the (a) known and (b) unknown correspondence localisation problems, using an uncertain initial pose distribution.

Both PGMs’ beliefs about \mathbf{x}_0 were quite far from the actual pose in this simulation. At this timestep, two measurements were taken, of landmark 1 and 2. However, both of these measurements happened to incorporate a large amount of noise, which skewed the beliefs for this pose. It should be noted that in most simulations, both PGMs computed much more accurate initial pose beliefs. An example is shown in Figure C.4. This specific plot was shown to demonstrate the effect of large measurement noise. For the following two timesteps, both sets of beliefs were very similar in terms of their distance from the true pose and their certainty. At these timesteps, landmarks 1 and 2 were measured. These landmarks are far apart from each other, and landmark 3 is much further from the robot than 1 and 2. Consequently, PGM2 was very certain of its measurement source allocation, resulting in a belief certainty that mirrors PGM1’s. The low likelihood of either of the measurements being produced by landmark 3 is the main reason for this. The main differences between the two sets of beliefs are present in the final two timesteps. The robot has moved too far from landmarks 1 and 2 to measure them, so only landmark 3 is measured. This reduction in information creates much larger belief uncertainties for both PGMs. The movement to the position at $t = 3$ was quite large in comparison to the other robot’s movements. This was due to a large, positive state transition noise at this timestep. This does not affect the beliefs of PGM1, since the measurement to landmark 3 can be used to directly determine its approximate position. However, PGM2’s beliefs are largely affected. Due to their ellipsoid shape, it was concluded that the PGM’s beliefs for \mathbf{x}_3 and \mathbf{x}_4 are the weak marginalisation of two distributions, where one is above and to the left of the other. The approximation of these two Gaussian distributions would spread the confidence ellipses in this manner. The confidence ellipses for PGM1’s beliefs are circular, since the horizontal and vertical dimensions remain statistically independent. Additionally, both the motion and measurement noise are symmetrical, according to the model’s definition.

To explore the performance of these PGMs further, they were provided with a more certain initial position distribution by decreasing the entries in the covariance matrix of the PDF $p(\mathbf{x}_0)$. However, the results were very similar to those shown in Figure 8.4. Only the PGM’s belief for \mathbf{x}_0 moved slightly closer to the robot’s true position. Obviously, the large measurement noise previously mentioned has a more powerful effect on the belief computation. The visualisation (Figure C.3) and a short discussion can be found in Appendix C.

For a final comparison, the measurement range limitation was removed to determine the effect of using a sensor with a larger reach. This provided both PGMs with much more information, as three new measurements were taken at every timestep. In this simulation, the robot started at position (1,1) again, moving upwards and to the right. The prior distribution of the initial position was changed back to the more realistic PDF with a larger uncertainty. Moving the robot and generating measurements over five timesteps

resulted in the plots shown in Figure 8.5.

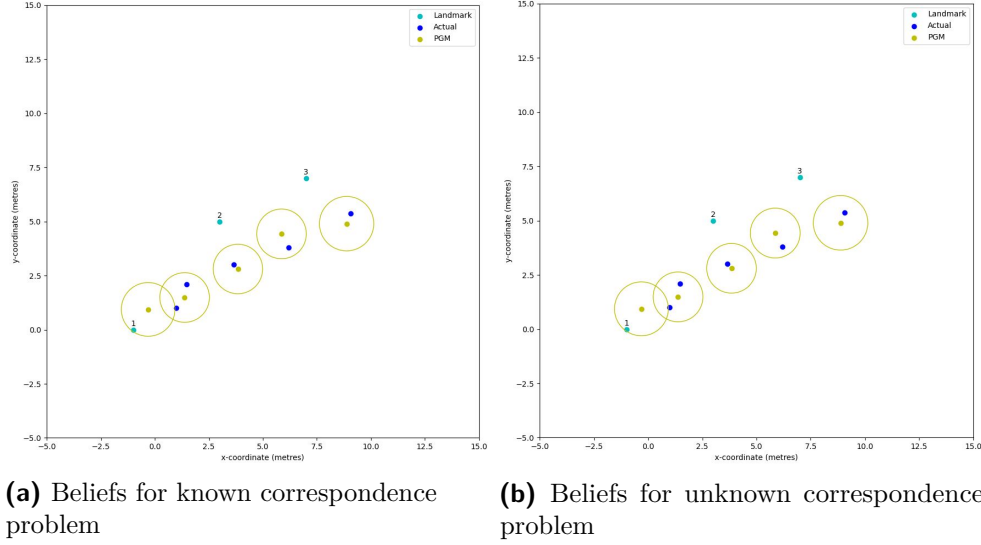


Figure 8.5: Beliefs computed by PGMs for the (a) known and (b) unknown correspondence localisation problems using an uncertain initial pose distribution and measurements of all landmarks in the map at all times

Providing both PGMs with more measurements meant they had access to more information, improving the accuracy and certainty of their beliefs. The performance improvement of PGM2 was very obvious. The PGM's access to measurements of all landmarks at every timestep meant it could allocate measurements to landmarks with very high accuracy and certainty. As a result, PGM2's beliefs are almost identical to PGM1's. The comparison of Figures 8.5 and C.3 to Figure 8.4 revealed an important observation – PGMs are more useful for localisation the more information they have access to. This can be practically implemented by providing the robot with a long range measurement sensor, so that it can measure as many mapped landmarks as possible, at each timestep.

Reducing the uncertainty for $p(\mathbf{x}_0)$ for this simulation resulted in a minuscule reduction in the first confidence ellipse's size. The beliefs were almost identical to those shown in Figure 8.5, so they are not shown.

In conclusion, PGMs can be used to compute pose beliefs for a robot with unknown measurement association. These beliefs are sufficiently accurate and certain under most circumstances. However, its performance differs vastly depending on the number of measurements provided, the measurement noise, and the number and spread of the mapped landmarks. Further commentary on how this model could be improved is provided in Section 9.1.

Chapter 9

Summary and Conclusion

We met the first and second objectives, by implementing and verifying the PGM for localisation with direct measurements in Chapter 7. This PGM performed better than the KF, due to its ability to use the given information more thoroughly. We extended this PGM twice to design the PGM for known correspondence. Performing inference on this PGM computed beliefs that, in general, were closer to the actual position, with more certainty. This was due to the increase in measurement information available to the PGM, allowing it to ‘average out’ more of the measurement noise. Objectives three and four were also achieved through the extension of the known correspondence PGM to model the unknown correspondence problem in Chapter 8. The computed beliefs were thoroughly compared to those of the PGM for known correspondence. The main conclusion obtained from this comparison was that PGMs can solve the unknown association problem with high accuracy and certainty, provided it is given measurements to all the landmarks in its environment. Providing fewer measurements diminishes the ability to solve the unknown association problem as it becomes less and less certain of their correspondence. This comparison completed the final objective of this study. An additional observation was that the PGM beliefs were not significantly affected by the changes in uncertainty of the prior position distribution.

In conclusion, the usefulness of probabilistic graphical models in mobile robot localisation is clear. The ease with which PGMs can be modified for different robot systems and problems makes this study’s PGM solution for localisation with unknown correspondence even more valuable. The computational benefits due to their distributed nature, and their ability to model complex probabilistic problems should provide experts in the field of robotics with significant reason to explore their further use. The success of this study provides a valuable basis for those who wish to use PGMs to solve more complex robot localisation problems, like SLAM. Additionally, this study showed that increasing the number of measurements the PGM has access to, will significantly improve its ability to compute beliefs that are closer to the true pose, with greater certainty.

9.1. Future Work

The PGM designed for the aim of this study has a broad range of possible applications and extensions. However, this study was limited to the exploration of using a PGM to solve the unknown association problem. Although it is known to be computationally efficient, this is not the only factor in determining which technique best solves the localisation problem. To determine its true performance, the accuracy and certainty of the PGMs beliefs should be compared to those calculated by the classical methods mentioned in Chapter 1. If its performance is verified as at least similar to the classical methods, it is worthwhile exploring the extension of this model.

Extending the robot’s measurement model could lead to some interesting insights into the application of PGMs to the localisation problem. For example, it is practically possible that sensors take more than one measurement of the same landmark at one timestep. It should consequently be possible for the PGM to believe that two association variables at the same timestep hold the same landmark identifier. It is also possible that a received measurement does not correlate to any of the landmarks in the robot’s environment. The potential extension of this PGM to handle ‘false measurements’ as well as the aforementioned double measurements would certainly improve the robustness of this model.

It may be worthwhile experimenting with the number of landmarks in the robot’s environment. Perhaps increasing the number of landmarks will not always improve the accuracy of the beliefs, since there will be more possible combinations of measurement associations. Researching the optimal map’s landmark density and spread for accurate belief computation with PGMs could reveal which environments are best suited for localisation using a PGM.

The clustering of the association variables at each timestep vastly increases the computational intensity of performing inference on the unknown correspondence PGM. Extending the model to use separate clusters for each association variable would create a more distributed network. This would ensure that the belief computations remain tractable for a much higher number of landmarks. In contrast to this, it may be possible to improve the accuracy of this PGM’s beliefs at the cost of some computational efficiency. Currently, each timestep’s association variables are in their own ‘discrete graphs’. By connecting these graphs, these variables could share information between timesteps, potentially improving the beliefs. However, this would create an even more ‘loopy’ cluster graph, resulting in worsened computational efficiency.

Our successful belief computation in the final PGM provides an avenue to the application of graphical models to the SLAM problem with unknown correspondence. Since computational efficiency is of great importance in robotics, successfully using PGMs to solve this problem could be of great value to the field of mobile robotics.

Bibliography

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
- [2] Daphne Koller, “Probabilistic graphical models specialization,” 2020. [Online]. Available: <https://www.coursera.org/specializations/probabilistic-graphical-models>
- [3] D. Koller and N. Friedman, *Probabilistic graphical models : principles and techniques*. The MIT Press, 2016.
- [4] Vincent Spruyt, “How to draw a covariance error ellipse,” 2014. [Online]. Available: <https://www.visiondummy.com/2014/04/draw-error-ellipse-representing-covariance-matrix/>
- [5] J. Schoeman and C. van Daalen, “Simultaneous localisation and mapping of a robotic vehicle using a probabilistic graphical model,” pp. 28–33, 2016.
- [6] D. Barber, *Bayesian reasoning and machine learning*. Cambridge University Press, 2020.

Appendix A

Project Planning Schedule

Week	Date	Description
1	27/07	Extensive study of Gaussian filters and robot motion modelling
2	3/08	Extensive study of robot perception modelling and mobile localisation
3	10/08	Extensive study of factors and Bayes networks
4	17/08	Extensive study of Markov random fields and belief propagation
5	24/08	Implement the basic Kalman filter solution in Python
6	31/08	Extensive study of the emdw library
7	07/09	Implement the basic PGM solution in c++, using the emdw library
8	14/09	Implement the basic PGM solution in c++, using the emdw library
9	21/09	Extend the PGM solution to single landmark measurement case
10	28/09	Extend the PGM solution to multiple landmark measurement case
11	5/10	Extend the PGM solution to solve the unknown correspondence problem
12	12/10	Extend the PGM solution to solve the unknown correspondence problem
13	19/10	Extend the PGM solution to solve the unknown correspondence problem
14	26/10	Debug code and solve problems
15	2/11	Write the study's report with feedback from the study leader
16	9/11	Submit Report

Table A.1: Project Plan Schedule (2020).

Appendix B

Outcomes Compliance

ELO Description	How it was achieved	Chapter
1. Problem Solving	Identifying the localisation problem, formulating the modular problems, solving them with PGMs, solving the unknown correspondence problem with a PGM.	4, 6, 7
2. Application of Scientific and Engineering Knowledge	Describing Gaussian and discrete distributions, developing a robot model, modelling localisation with PGMs.	2, 3, 7
3. Engineering Design	Solving localisation with the KF, modelling and solving both the simpler localisation problems and the unknown correspondence problem with PGMs.	5, 7, 8
4. Investigations, Experiments and Data Analysis	Comparing a PGM and a KF for localisation, analysing PGM's beliefs for different measurement models, comparing beliefs of PGMs for known and unknown correspondence problem.	7, 8
5. Engineering Methods, Skills and Tools, Including Information Technology	Independent development of software for PGMs using c++ and Python for simulations.	5, 7, 8
6. Professional and Technical Communication	Professional report style and language, use of appropriate diagrams, explanations of technical terms.	All
8. Individual Work	Refining unknown correspondence problem into sub-problems and solving them individually to build final solution	All
9. Independent Learning Ability	Understanding information about robot models, the localisation problem and PGMs. Using this knowledge to design the solutions.	All

Table B.1: ECSA Exit Level Outcomes (ELOs), how they were achieved and in which chapters.

Appendix C

Additional Results

This appendix contains additional PGM belief simulation visualisations that are referred to in this report. All the simulations were for five timesteps, with the robot starting at $\{1,1\}$ and moving towards the top right. The differing models and information used is indicated in the captions.

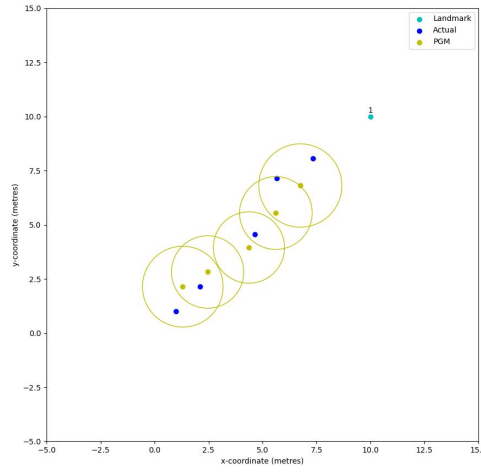


Figure C.1: The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{6.2,9.2\}$), with the corresponding PGM beliefs using the single landmark measurement model.

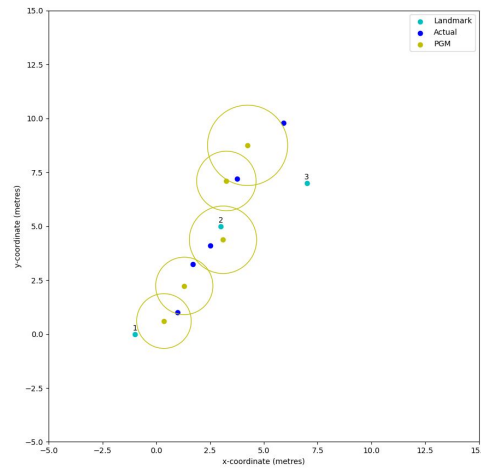


Figure C.2: The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{5.9,9.8\}$), with the corresponding PGM beliefs using the multiple landmark measurement model with known correspondence.

Using the same motion and measurements utilised for Figure 8.4, Figure C.3 was created by giving the robot a more certain initial pose.

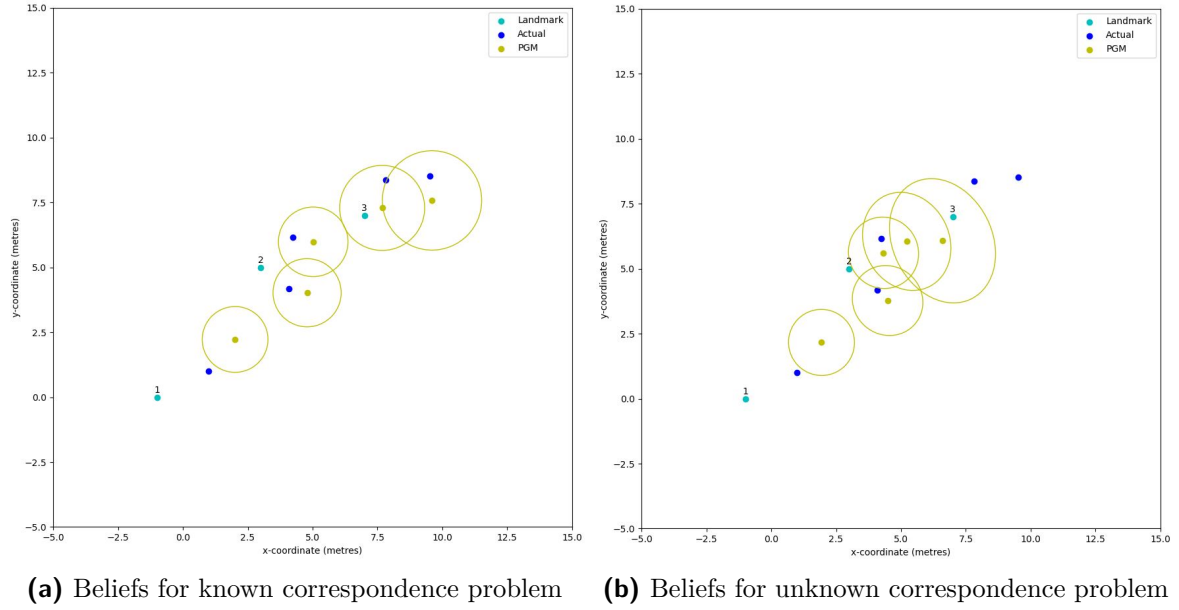
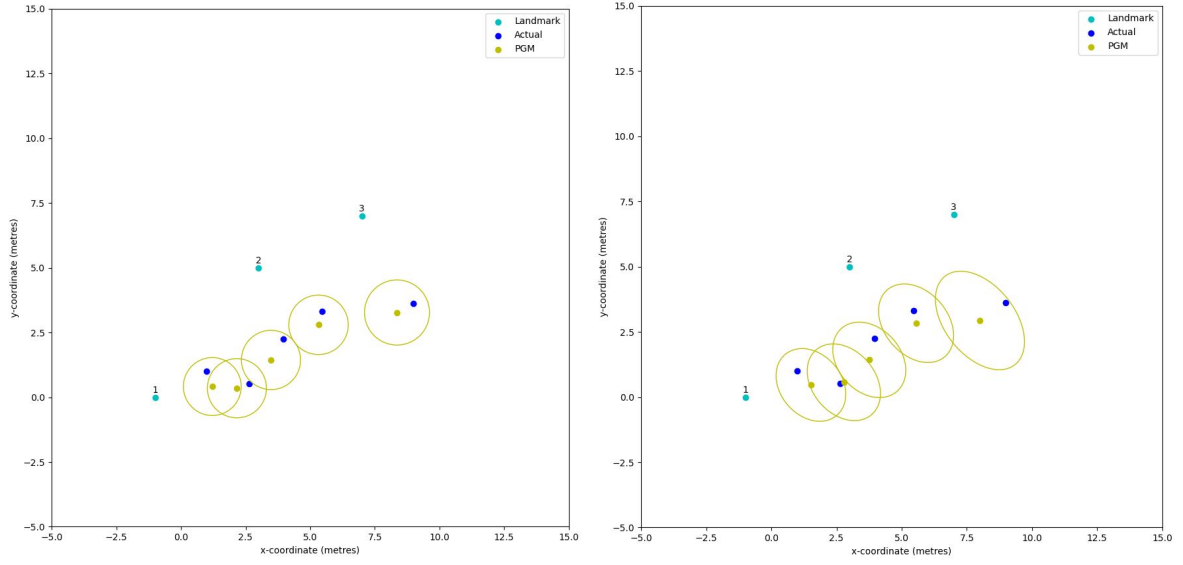


Figure C.3: The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{9.5,8.5\}$), with the corresponding PGM beliefs for the (a) known and (b) unknown correspondence localisation problems, using a more certain initial pose distribution.

Increasing the certainty of the initial position only had a noticeable effect on the PGM's belief for \mathbf{x}_0 . The belief's accuracy was improved, although the true position still lies outside the 95 % confidence region for both PGMs. Obviously, the large measurement noise previously mentioned has a more powerful effect on the belief computation. The accuracy improvement propagated forward in time, but it diminished in effect with every timestep. By $t = 3$, both PGMs' confidence ellipses are identical to those shown in Figure 8.4. Therefore, the accuracy of PGM beliefs for localisation problems will be more effectively increased by providing less noisy measurement information, rather than a more precise starting position.

Figure C.4 shows another simulation where both models were provided with big initial position uncertainties, and they had no range limitation. This simulation shows that the inaccuracies in Figures 8.4 and C.3 were in fact due to the measurement noise.



(a) Beliefs for known correspondence problem (b) Beliefs for unknown correspondence problem

Figure C.4: The robot's actual positions (starting at $\{1,1\}$ and moving upwards and to the right to $\{9,3.6\}$), with the corresponding PGM beliefs for the (a) known and (b) unknown correspondence localisation problems, using a more certain initial pose distribution.

Appendix D

The Square Root Canonical Form for Numerical Stability

This chapter shows the derivation for the square root canonical form for the direct measurement model's measurement factor, before it is reduced. It was included to show the inner workings of the *EMDW* software that computes the square root form via the *Cholesky()* method.

Following the procedure shown in Equations 7.1 and 7.2, it is found that $p(\mathbf{z}_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{z}_t; C\mathbf{x}_t, Q)$. Using the process in Section 2.3 again, the canonical representation of this distribution is derived as

$$\mathcal{N}(\mathbf{z}_t; C\mathbf{x}_t, Q) = \mathcal{C} \left(\begin{bmatrix} \mathbf{z}_t & \mathbf{x}_t \end{bmatrix}^T; K_m, \mathbf{h}_m, g_m \right), \quad (\text{D.1})$$

where

$$K_m = \begin{bmatrix} Q^{-1} & -Q^{-1}C \\ -C^T Q^{-1} & C^T Q^{-1}C \end{bmatrix}, \mathbf{h}_m = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \text{ and } g_m = -\ln [(2\pi|Q|)^{N/2}].$$

The Cholesky decomposition will be applied to the K_m matrix of Equation D.1. Prior to this calculation, it is useful to show the form of the measurement noise matrix Q , and its inverse Q^{-1} . Representing the standard deviation of the noise with σ^2 , the matrices are

$$Q = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \Rightarrow Q^{-1} = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{\sigma^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma} & 0 \\ 0 & \frac{1}{\sigma} \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma} & 0 \\ 0 & \frac{1}{\sigma} \end{bmatrix}. \quad (\text{D.2})$$

The K_m matrix is divided into blocks, using the Cholesky decomposition definition, as follows:

$$\begin{aligned} K_m &= \begin{bmatrix} L_{0,0} & 0 \\ L_{1,0} & L_{1,1} \end{bmatrix} \begin{bmatrix} L_{0,0}^T & L_{1,0}^T \\ 0 & L_{1,1}^T \end{bmatrix} \\ &= \begin{bmatrix} L_{0,0}L_{0,0}^T & L_{0,0}L_{1,0}^T \\ L_{1,0}L_{0,0}^T & L_{1,0}L_{1,0}^T + L_{1,1}L_{1,1}^T \end{bmatrix}. \end{aligned} \quad (\text{D.3})$$

Each block of the factorisation in Equation D.3 is found using Q^{-1} shown in Equation D.2:

$$Q^{-1} = L_{0,0}L_{0,0}^T = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{\sigma^2} \end{bmatrix} \Rightarrow L_{0,0} = \begin{bmatrix} \frac{1}{\sigma} & 0 \\ 0 & \frac{1}{\sigma} \end{bmatrix}, \quad (\text{D.4})$$

$$-C^T Q^{-1} = L_{1,0}L_{0,0}^T = \begin{bmatrix} -\frac{1}{\sigma^2} & 0 \\ 0 & -\frac{1}{\sigma^2} \end{bmatrix} \Rightarrow L_{1,0} = \begin{bmatrix} -\frac{1}{\sigma} & 0 \\ 0 & -\frac{1}{\sigma} \end{bmatrix}, \quad (\text{D.5})$$

$$\text{and } C^T Q^{-1} C = \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{\sigma^2} \end{bmatrix} = L_{1,0}L_{1,0}^T + L_{1,1}L_{1,1}^T \quad (\text{D.6})$$

$$= \begin{bmatrix} \frac{1}{\sigma^2} & 0 \\ 0 & \frac{1}{\sigma^2} \end{bmatrix} + L_{1,1}L_{1,1}^T \Rightarrow L_{1,1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}. \quad (\text{D.7})$$

Combining the results of Equations D.4 through D.7 produces the lower triangular matrix:

$$L_m = \begin{bmatrix} \frac{1}{\sigma^2} & 0 & 0 & 0 \\ 0 & \frac{1}{\sigma^2} & 0 & 0 \\ -\frac{1}{\sigma^2} & 0 & 0 & 0 \\ 0 & -\frac{1}{\sigma^2} & 0 & 0 \end{bmatrix}. \quad (\text{D.8})$$

It is simple to verify that this is the correct solution for L using the definition of the Cholesky decomposition in Section 2.1.3. The square root canonical form is thus given as

$$\mathcal{C} \left(\begin{bmatrix} \mathbf{z}_t & \mathbf{x}_t \end{bmatrix}^T; L_m, \mathbf{h}_m, g_m \right) \quad (\text{D.9})$$

Appendix E

Localisation Cluster Potentials

Various cluster potentials are defined for the different localisation problems discussed in this study. These cluster potentials are theoretically important, but we did not handle them manually. The EMDW library computes them in the background when the *loopyBP-CG()* method is called.

E.1. Direct Measurement Model

The direct measurement model's motion cluster potentials ϕ_{2t+2} are defined as

$$\Phi_{2t+2}(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}_t) \triangleq p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t) , \quad (\text{E.1})$$

for $t \neq 0$. The control vector \mathbf{u}_t is known at all times, so no $p(\mathbf{u}_t)$ distribution contributes any information to the model. The incorporation of $\hat{\mathbf{u}}_t$ as evidence of \mathbf{u}_t reduces the motion cluster potentials. Using the reduced canonical form in Equation 7.3, Equation E.1 becomes

$$\Phi_{2t+2}(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}_t = \hat{\mathbf{u}}_t) \triangleq \mathcal{C} \left(\begin{bmatrix} \mathbf{x}_t & \mathbf{x}_{t-1} \end{bmatrix}^T ; K'_s, \mathbf{h}'_s, g'_s \right) . \quad (\text{E.2})$$

The initial pose's cluster potential is defined by

$$\Phi_2(\mathbf{x}_0) \triangleq p(\mathbf{x}_0) . \quad (\text{E.3})$$

The direct measurement model's cluster potentials Ψ_{2t+1} are

$$\Psi_{2t+1}(\mathbf{z}_t, \mathbf{x}_t) \triangleq p(\mathbf{z}_t | \mathbf{x}_t) , \quad (\text{E.4})$$

where Ψ is used instead of Φ to make the different cluster types explicit. Reducing the measurement cluster's with evidence causes Equation E.4 to be redefined as

$$\Psi_{2t+1}(\mathbf{z}_t = \hat{\mathbf{z}}_t, \mathbf{x}_t) \triangleq \mathcal{C}(\mathbf{x}_t; K'_m, \mathbf{h}'_m, g'_m) . \quad (\text{E.5})$$

E.2. Single Landmark Measurement Model

The measurement clusters' potentials in Equation E.5 can be redefined using the reduced canonical form, as

$$\Psi_{2t+1}(z_t = \hat{z}_t, \mathbf{x}_t) \triangleq \mathcal{C}(\mathbf{x}_t; K'_l, \mathbf{h}'_l, g'_l) . \quad (\text{E.6})$$

The remaining cluster potentials are calculated in the same manner as before.

E.3. Multiple Landmark Measurement Model with Known Correspondence

The measurement cluster potential is given by

$$\Psi_{2t+1}(z_{t,k} = \hat{z}_{t,k} \forall k \in [0, k_t - 1], \mathbf{x}_t) \triangleq \prod_{k=0}^{k_t-1} \mathcal{C}(\mathbf{x}_t; K'_{l,k}, \mathbf{h}'_{l,k}, g'_{l,k}) , \quad (\text{E.7})$$

where k_t is the total number of measurements at timestep t . The arguments $K'_{l,k}$, $\mathbf{h}'_{l,k}$ and $g'_{l,k}$ represent the canonical components of Equation 7.6 associated with measurement $\hat{z}_{t,k}$.

E.4. Multiple Landmark Measurement Model with Unknown Correspondence

The motion cluster potentials, represented by Φ_{5t+5} are defined as

$$\Phi_{5t+5}(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{u}_t = \hat{\mathbf{u}}_t) \triangleq \mathcal{C}\left(\begin{bmatrix} \mathbf{x}_t & \mathbf{x}_{t-1} \end{bmatrix}^T; K'_s, \mathbf{h}'_s, g'_s\right) , \quad (\text{E.8})$$

where the canonical form's components are still defined by Equation 7.3.

The association variable cluster potentials are given by

$$\alpha_{5t+1}(a_{t,0}, \dots, a_{t,k_t-1}) \triangleq p(a_{t,0}, \dots, a_{t,k_t-1}) . \quad (\text{E.9})$$

Reducing each measurement with its respective $\hat{z}_{t,k}$, the measurement cluster potentials are defined as

$$\Psi_{5t+2+k}(z_{t,k} = \hat{z}_{t,k}, \mathbf{x}_t, a_{t,k}) \triangleq p(z_{t,k} = \hat{z}_{t,k} | \mathbf{x}_t, a_{t,k}) . \quad (\text{E.10})$$

Appendix F

Prior Joint Association Variable Probability Distributions

This chapter explicitly defines the table form of the prior joint association variable probability distributions. The zero probability entries have been omitted for conciseness. If three measurements are taken, $k_t = 3$ and the unnormalised prior distribution is defined by Table F.1. The prior distributions for two measurements ($k_t = 2$) and ($k_t = 1$) are given by Tables F.2 and F.3 respectively.

$a_{t,0}$	$a_{t,1}$	$a_{t,2}$	$p'(a_{t,0}, a_{t,1}, a_{t,2})$
0	1	2	1
0	2	1	1
1	2	0	1
2	1	0	1
1	0	2	1
2	0	1	1

Table F.1: The prior joint association variable probability distribution when three measurements are taken with a map of three landmarks.

$a_{t,0}$	$a_{t,1}$	$p'(a_{t,0}, a_{t,1})$
0	1	1
0	2	1
1	0	1
2	0	1
2	0	1
2	1	1

Table F.2: The prior joint association variable probability distribution when two measurements are taken with a map of three landmarks.

$a_{t,0}$	$p'(a_{t,0})$
0	1
1	1
2	1

Table F.3: The prior joint association variable probability distribution when one measurement is taken with a map of three landmarks.