Parametric POMDPs for Planning in Continuous State Spaces

Alex M. Brooks

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

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

January 2007
Declaration

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

Alex M. Brooks

January 10, 2007
Abstract

Alex M. Brooks
The University of Sydney

Parametric POMDPs for Planning in Continuous State Spaces

This thesis is concerned with planning and acting under uncertainty in partially-observable continuous domains. In particular, it focusses on the problem of mobile robot navigation given a known map. The dominant paradigm for robot localisation is to use Bayesian estimation to maintain a probability distribution over possible robot poses. In contrast, control algorithms often base their decisions on the assumption that a single state, such as the mode of this distribution, is correct. In scenarios involving significant uncertainty, this can lead to serious control errors. It is generally agreed that the reliability of navigation in uncertain environments would be greatly improved by the ability to consider the entire distribution when acting, rather than the single most likely state.

The framework adopted in this thesis for modelling navigation problems mathematically is the Partially Observable Markov Decision Process (POMDP). An exact solution to a POMDP problem provides the optimal balance between reward-seeking behaviour and information-seeking behaviour, in the presence of sensor and actuation noise. Unfortunately, previous exact and approximate solution methods have had difficulty scaling to real applications.

The contribution of this thesis is the formulation of an approach to planning in the space of continuous parameterised approximations to probability distributions. Theoretical and practical results are presented which show that, when compared with similar methods from the literature, this approach is capable of scaling to larger and more realistic problems.

In order to apply the solution algorithm to real-world problems, a number of novel improvements are proposed. Specifically, Monte Carlo methods are employed to estimate distributions over future parameterised beliefs, improving planning accuracy without a loss of efficiency. Conditional independence assumptions are exploited to simplify the problem, reducing computational requirements. Scalability is further increased by focussing computation on likely beliefs, using metric indexing structures for efficient function approximation. Local online planning is incorporated to assist global offline planning, allowing the precision of the latter to be decreased without adversely affecting solution quality.

Finally, the algorithm is implemented and demonstrated during real-time control of a mobile robot in a challenging navigation task. We argue that this task is substantially more challenging and realistic than previous problems to which POMDP solution methods have been applied. Results show that POMDP planning, which considers the evolution of the entire probability distribution over robot poses, produces significantly more robust behaviour when compared with a heuristic planner which considers only the most likely states and outcomes.
Acknowledgements

This work was made possible by many humans other than myself, and I am extremely grateful for their support.

I would like to begin by thanking my supervisors. Stef, thank you for the guidance, the willingness to talk over ideas, and for always expecting a high standard. Alex, thanks for your invaluable academic support, and especially for always staying for one more drink.

Thanks to everyone at the ACFR who has helped make things entertaining and stimulating. Particularly, thank you Toby, for the fake identity and for never getting offended with the German jokes. Thanks to my cubicle mates Fabio, for the mermaids and the chubby seagulls, and Ian, for always setting me straight and keeping my feet on the ground. Thank you George, for maintaining the uninterrupted beer supply. Thank you Ben, for the coffee breaks and for letting me squat in your office. Thank you Hugh, for being an inspiration and for providing a lab environment where I can code barefoot on the couch.

I also want to thank all my mates from outside Sydney, for a willingness to travel and for taking my mind off robotics.

Thank you to Rosemary and Simon for many years of encouragement, support, and travel in strange places.

I particularly want to thank my proof-readers, Rosemary and Ben, for giving up so much of their time to make this document legible. It’s a big job and I appreciate it.

I’d like to reserve my final thank you for my wuzzer, Shiona, for her support throughout this process, and for putting up with my thesis-writin’ mood swings.
To my parents, for making anything possible.
Contents

Declaration i
Abstract ii
Acknowledgements iii
Contents v
Nomenclature x

1 Introduction 1
1.1 Markov Decision Processes 2
1.2 Partial Observability 3
1.3 Solving POMDPs 4
1.4 Application Domain 7
1.5 Contributions 9
1.6 Thesis Structure 10

2 Sequential Decision Making 13
2.1 Markov Decision Processes 13
2.1.1 Solving MDPs Using Value Iteration 16
2.2 From MDPs to POMDPs 17
2.2.1 POMDPs as Information-State MDPs 18
2.2.2 \( I_{\text{hist}} \) and Derived Information Spaces 21
2.3 Belief States and \( I_{\text{prob}} \) 25
2.4 Discretised State Spaces: \( \kappa_{\text{probdisc}} \) 28
## Contents

2.4.1 Gradient-Based Solution Methods ............................................. 29
2.4.2 Approximate Gradient-Based Solutions: Point-Based Methods ...... 31

2.5 Beyond \( \mathcal{I}_{\text{probdisc}} \) ................................................. 33
  2.5.1 Fitted Value Iteration ......................................................... 33
  2.5.2 Converting a Continuous MDP to a Discrete MDP of Size \(|G|\) ... 35
  2.5.3 Application to Various I-Spaces .......................................... 37

2.6 Belief Set Selection ..................................................................... 39

2.7 Non-Value-Iteration-Based Approaches ....................................... 40
  2.7.1 Heuristic Approaches ........................................................... 40
  2.7.2 Policy Iteration .................................................................... 42
  2.7.3 Forward Search .................................................................... 43
  2.7.4 Hierarchical Approaches ...................................................... 45
  2.7.5 POMDPs as Graphical Models ............................................. 46

2.8 Summary ..................................................................................... 48

3 Parametric Information Spaces ......................................................... 49
  3.1 Choosing an Information Space .................................................. 51
  3.2 Gaussian Information Spaces .................................................... 52
    3.2.1 Quality of a Gaussian Approximation .................................. 52
    3.2.2 Dimensionality of the Belief-Space ..................................... 55
  3.3 Function Approximation and Belief Set Selection ......................... 56
    3.3.1 Multilinear Interpolation .................................................... 56
    3.3.2 Freudenthal Triangulation ................................................ 57
  3.4 Solving Parametric POMDPs ...................................................... 59
    3.4.1 Deriving the I-State MDP .................................................... 61
    3.4.2 Computational Complexity ................................................ 63
  3.5 Plan Execution .......................................................................... 64
  3.6 BlockWorld: A Simple Continuous Navigation Problem ............... 65
    3.6.1 The Rules of BlockWorld ................................................... 65
    3.6.2 Discrete Solutions ............................................................. 67
    3.6.3 Parametric Solution .......................................................... 68
CONTENTS

3.6.4 Parameter Summary ................................................. 70
3.7 Basic BlockWorld Experiments .......................................... 70
  3.7.1 Comparison in the Standard World .............................. 71
  3.7.2 Comparison in a Tiled World .................................. 74
3.8 Summary ................................................................. 75

4 An Improved Belief Transition Function ............................... 77
  4.1 Experiments on Different Worlds ................................. 78
  4.2 The Requirement to Anticipate Future Observations .......... 80
  4.3 An Improved Belief Transition Function ......................... 82
    4.3.1 A Belief Transition Function Using Monte Carlo Methods . 82
    4.3.2 Stochastic Effects ........................................... 85
  4.4 Efficient Calculation of the Belief Transition Function ........ 89
    4.4.1 Re-use of Predictions ....................................... 89
    4.4.2 Reducing Likelihood Calculations ......................... 90
  4.5 Scalability .......................................................... 95
  4.6 Results .............................................................. 97
  4.7 Summary ............................................................. 97

5 Factoring Observations .................................................. 100
  5.1 A Different View of the FVI Approximation .................... 101
  5.2 Factoring Conditionally-Independent Observation Components .... 103
    5.2.1 Conditioning on the Belief after Acting and Partial Observation .. 104
  5.3 Application to the PPOMDP Solution Algorithm ................ 105
    5.3.1 Representing $p(I^+|I^n_o)$ ................................ 106
  5.4 A Modified PPOMDP Solution Algorithm ......................... 107
    5.4.1 Computational Complexity .................................. 107
  5.5 Experiments ........................................................ 110
  5.6 Results ............................................................ 111
  5.7 Summary ........................................................... 112
6 Using Arbitrary Belief Sets

6.1 Efficient Implementation of a Weighting Function ........................................ 115
6.2 A Review of Similarity Search Algorithms .................................................. 116
   6.2.1 Similarity Search in Vector Spaces: Spatial Access Methods .............. 117
   6.2.2 Similarity Search in Metric Spaces: Metric Indexing Structures ....... 119
6.3 Inter-Gaussian Distance Metrics ............................................................... 120
   6.3.1 Inter-Gaussian Distance Metrics ...................................................... 121
   6.3.2 The Triangle Inequality ...................................................................... 126
   6.3.3 Comparison of Metrics on a Simple Example .................................. 127
6.4 Search Efficiency ........................................................................................ 129
   6.4.1 Improving Efficiency Through Metric Repair ................................ 132
6.5 Evaluation of Expected Plan Quality ......................................................... 134
   6.5.1 Probabilistic Paths through Discrete Belief Sets ............................. 135
6.6 BlockWorld Experiments ........................................................................... 139
   6.6.1 Comparison Against Previously-Presented Algorithms .................. 140
   6.6.2 Experiments with the Distance Metric ............................................ 143
   6.6.3 Experiments with a Distinct Set $B_C$ ............................................. 146
   6.6.4 Experiments with Belief Set Generation ......................................... 147
6.7 Summary ....................................................................................................... 149

7 Plan Execution and Forward Planning ......................................................... 152

7.1 Forward Planning as Game-Tree Expansion .............................................. 153
7.2 Strategies for Related Tree-Expansion Problems ..................................... 155
   7.2.1 Common Strategies for MiniMax Tree Expansion ......................... 156
7.3 Forward Planning for a Particle-Based PPOMDP ..................................... 158
   7.3.1 Logical Tree Structure ................................................................. 158
   7.3.2 Observation-Nodes ....................................................................... 160
   7.3.3 Action-Nodes ............................................................................. 161
7.4 Experiments ................................................................................................. 162
7.5 Results .......................................................................................................... 163
7.6 Planning Further Ahead .............................................................................. 164
7.7 Summary ....................................................................................................... 167
8 Scaling to the Real World 168

8.1 Application Domain ........................................ 169
8.2 Localisation Approach .................................... 172
8.3 From BlockWorld to the Real World ....................... 173
  8.3.1 The PPOMDP Model ................................. 173
  8.3.2 Discussion of the Model .............................. 178
  8.3.3 PPOMDP Parameter Settings ......................... 179
8.4 Evaluation of Policies ..................................... 180
  8.4.1 The Component Framework for Online Evaluation of Policies .... 181
  8.4.2 The Process Noise Model Used in Realistic Simulation .......... 182
  8.4.3 The Competition: Non-Probabilistic Path Planning .......... 183
8.5 ToyWorld: A Toy Problem with Realistic Dynamics .......... 185
  8.5.1 Results ........................................... 186
8.6 The RealWorld Problem ................................... 189
  8.6.1 Problem Description ................................. 189
  8.6.2 Results ........................................... 191
8.7 Summary .................................................. 199

9 Conclusion 200

9.1 Summary .................................................. 200
9.2 Avenues for Future Research ............................... 202
  9.2.1 Dynamic or Unknown Environments ....................... 202
  9.2.2 Further Application of POMDP Solution Methods ............ 203
  9.2.3 Increased Belief Complexity ............................ 203
  9.2.4 Heuristics for Dealing with Non-Gaussian Beliefs .......... 204
  9.2.5 Operating in a More Expressive I-Space .................. 205
9.3 Conclusion ............................................... 206

A Derivation of $w_\mu$ for Repairing the Matusita Distance 207

Bibliography 209
Nomenclature

Notation

\((\cdot)_k\) \hspace{1cm} (\cdot) at discrete time \(k\)
\((\cdot)^+\) \hspace{1cm} (\cdot) at the next time interval

General Symbols

\(k\) \hspace{1cm} an index into discrete time
\(x\) \hspace{1cm} a state
\(X\) \hspace{1cm} the state space
\(u\) \hspace{1cm} an action
\(U\) \hspace{1cm} the action space
\(f\) \hspace{1cm} the state transition function
\(w\) \hspace{1cm} a disturbance affecting the state transition
\(T\) \hspace{1cm} a function describing the state transition probabilities
\(r\) \hspace{1cm} a state-based reward
\(R\) \hspace{1cm} a state-based reward function
\(\gamma\) \hspace{1cm} a discount factor
\(\pi\) \hspace{1cm} a policy
\(V\) \hspace{1cm} a value function
\(z\) \hspace{1cm} an observation
\(Z\) \hspace{1cm} the observation space
\(h\) \hspace{1cm} the observation function
\(v\) \hspace{1cm} a disturbance affecting the observation
\(O\) \hspace{1cm} the function describing the observation probabilities
\(c_0\) \hspace{1cm} the initial conditions available to a POMDP agent

\(I\) \hspace{1cm} an information state
\(\mathcal{I}\) \hspace{1cm} an information space
\(f_I\) \hspace{1cm} the information transition function
\[ r_I \] an I-state-based reward
\[ R_I \] the I-state-based reward function
\[ \pi_I \] an information-based policy
\[ I_{der} \] a derived information-space
\[ I_{der}^\ast \] a derived information-state
\[ \kappa_{der} \] a mapping function, mapping to a derived information space

\[ s \] a discrete state
\[ S \] a set of discrete states
\[ R_s \] a reward function defined for discrete states
\[ \alpha \] a value function hyperplane
\[ \Gamma \] a set of value function hyperplanes
\[ \Gamma^u,\alpha^+ \] a set of intermediate alpha-vectors

\[ \lambda \] a weighting function
\[ G \] a set of states
\[ x_{G,i} \] the \( i \)'th point in \( G \)
\[ \psi(x_{G,i}) \] the value of \( x_{G,i} \)
\[ \Psi \] a set of state-value pairs
\[ \phi \] a function approximator
\[ B \] a set of beliefs
\[ I_{B,i} \] the \( i \)'th point in \( B \)

\[ \Delta \] a set of posterior beliefs
\[ Q \] a set of particles
\[ q \] a particle, *i.e.* a tuple of the form \( <x,w> \)
\[ w \] a particle weight
\[ \hat{z}^+ \] an expected observation
\[ L \] a likelihood matrix
\[ W \] a weight matrix

\[ \delta \] the Dirac delta function
\[ z_{\alpha}^+ \] a component of the observation \( z^+ \)
\[ Z_\alpha \] the space of possible values of \( z_{\alpha}^+ \)
\[ I_\alpha^+ \] an intermediate I-state produced after incorporating \( z_{\alpha}^+ \)
\[ \mathcal{I}_\alpha \] the space of possible values of \( I_\alpha^+ \)
\[ f_{I_\alpha} \] the I-state transition function which incorporates \( z_{\alpha}^+ \)
\[ B_\alpha \] a set of I-states in \( \mathcal{I}_\alpha \)
\[ I_{B_\alpha,i}^+ \] the \( i \)'th I-state in \( B_\alpha \), at the next time interval
\[ T_{I_\alpha} \] a CPT governing transitions from \( B \) to \( B_\alpha \)
\[ z_C^+ \] an observation from a robot’s collision sensor
An observation from a robot’s range sensors

*an observation from a robot’s range sensors*

$D$ a distance measure

$P$ a database of points

$V$ the domain of $P$

$q$ a query into the database

$r$ the radius of a search query

$v$ a vector of parameters

$\tau$ the distance to the closest point found so far during a search

$\eta$ a kernel function

$\zeta$ a bandwidth

$F$ a set of features

$f_i$ the $i$'th feature in $F$

$Z_f$ a set of polar feature observations

$z_{f,i}$ the $i$'th feature observation in $Z_f$, of the form $<z_{r,i}, z_{b,i}>$

$z_{r,i}$ the range of $z_{f,i}$

$z_{b,i}$ the range of $z_{f,i}$

$\hat{Z}_f$ the expected set of feature observations

**Abbreviations**

- MDP Markov Decision Process
- POMDP Partially Observable Markov Decision Process
- FVI Fitted Value Iteration
- CPD Conditional Probability Distribution
- CPT Conditional Probability Table
- SLAM Simultaneous Localisation and Mapping
- PWLC Piecewise-Linear and Convex
Chapter 1

Introduction

This thesis is concerned with the problem of planning and acting in uncertain, partially observable, continuous domains. In particular, it focusses on the task of planning and acting for mobile robot navigation when a map of the environment is available. Robot navigation problems are particularly challenging for planners because they are inherently continuous, uncertain, and non-linear. However, the ability to make good plans despite these conditions is fundamental to an autonomous mobile robot’s ability to navigate reliably in real-world environments.

Classical Artificial Intelligence (AI) planning assumes that environments are fully-observable, deterministic, finite, static, and discrete [95]. The first major planning system for such environments was STRIPS [39], which represented the state of the environment with a set of symbols. A set of actions were posited, each of which had a set of pre-conditions and a set of deterministic effects on the symbolic state of the world. Given a start state, the definitions of actions, and a goal state, a STRIPS-style planner could autonomously map out a fixed sequence of actions which would lead to that goal.

Unfortunately, few of the assumptions of classical AI planning hold for realistic mobile robot applications. Fixed sequences of actions are inappropriate because actions’ outcomes are unpredictable. Real robots therefore have difficulty executing STRIPS-style plans [40]. Instead, feedback is required: an agent must observe the world and react accordingly. A number of extensions allow classical AI planning systems to incorporate feedback, for example by making plans conditional on the state of the world [95][12].
Another important omission of classical AI planning is that, rather than reaching one or more goal states, agents in realistic scenarios are usually required to satisfy various (possibly competing) objectives simultaneously. For example, robots should act so as to minimise the risk of encountering hazards which might cause them harm. One way of specifying objectives is through the use of a reward\(^1\) function \([105][59]\), which specifies the desirability of possible states of the world, and perhaps the desirability of particular actions in particular states. A more sophisticated model, which accounts for unpredictable actions and general reward functions, is a Markov Decision Process.

### 1.1 Markov Decision Processes

A Markov Decision Process, or MDP, provides a general mathematical model for the interaction between an agent and the world. Many classical AI planning algorithms can be formulated as special cases of MDPs \([16]\). An MDP assumes that the state of the world at any time can be described by a set of continuous or discrete variables. This state evolves in small discrete time-steps, affected by the agent’s actions. The agent chooses these actions based on its direct and infallible knowledge of the state.

To account for unpredictability in the world, the MDP model requires that the effects of agents’ actions can be described by stationary probability distributions. That is, from any given state and for any given action, an MDP specifies a probability distribution over subsequent states.

Matters are simplified considerably by the Markov assumption \([111]\). This asserts that the current state is a sufficient statistic for the past. In other words, if the agent knows the current state of the world, the details of how the world came to be in that state convey no extra information about what will happen in the future. This is usually a fairly accurate assumption for the real world, given a sufficiently descriptive state vector.

By framing a problem as an MDP, one gains access to a powerful arsenal of solution algorithms \([16][105][12]\). The output of many solution algorithms is a value function, which specifies a value for every possible state. Loosely speaking, this value is the sum of rewards which can be obtained in the future by acting optimally (with an infinite-horizon lookahead).

\(^{1}\)While reward is usually used in the computer science literature, cost (negative reward) is usually used in the economics and operations research literature \([10]\)
from that state. Armed with the value function, an agent need not plan ahead, since plans are implicitly encoded in the value function. An agent can act optimally simply by greedily choosing actions which will immediately lead to high-value states.

MDP solution algorithms have solved many challenging problems, particularly for board games such as backgammon [106]. Translating this success to real-world problems can be problematic however, because the MDP formulation assumes that the agent has perfect knowledge of the state. A more realistic formulation assumes only partial observability.

1.2 Partial Observability

While an MDP models uncertainty in an agent’s actions, it assumes that the agent is completely aware of the state of the world. This assumption is often invalid in real scenarios, particularly for the kinds of problems considered in this thesis. A more realistic model is a Partially Observable Markov Decision Process, or POMDP. A POMDP extends an MDP by assuming that, rather than sensing the state of the world directly, an agent can make observations which give it imperfect information about the state. The POMDP formulation assumes that the likelihood of observations given the hidden state can be described by stationary probability distributions.

While an MDP agent has the luxury of making decisions based on the state, a POMDP agent must make decisions based solely on the history of actions and observations. This history represents everything the agent knows about the world, and is often referred to as the information-state. This dependence on history complicates matters. For MDPs, the Markov property implies that an agent can safely ignore history. The POMDP formulation also assumes that the state obeys the Markov property, however this state is no longer directly observable. The entire history of actions and observations is therefore relevant as it potentially confers information about the hidden state.

The dominant approach to avoiding this history is to use all the available information to maintain a probability distribution, or belief, over possible states. The belief is a sufficient statistic for history. That is, if an agent knows the current probability distribution, the observations and actions which led to that distribution are irrelevant for predicting the future. It will be shown in Chapter 2 that a POMDP can be seen as a special kind of MDP. The unobservable state can be replaced by the observable information-state, which can be
CHAPTER 1. INTRODUCTION

Figure 1.1: POMDP solution methods span a continuum. On one extreme, the combination of a fine discretisation with exact value iteration will produce excellent plans but will be incapable of scaling to realistic problems. On the other extreme, heuristics require little computation but fail to take a principled approach to uncertainty. The most useful planner lies somewhere in the middle. The figure is adapted from [94].

summarised by a belief-state. Standard MDP solution algorithms can then be applied to the resultant MDP.

POMDPs are excellent models for many mobile robot navigation problems in which the state is the pose (position and heading) of the robot. A typical scenario involves the use of sensors such as cameras, laser range-finders, and wheel encoders to gather information. The information is imperfect because sensors are noisy, cannot look everywhere at once, and usually cannot sense the state of interest (i.e. the pose) directly. In this context, a POMDP solution represents a plan which allows the robot to gather the information it requires while simultaneously bringing it to its goal.

1.3 Solving POMDPs

Realistic mobile navigation problems are difficult to solve using POMDPs because their state, action, and observation spaces are large and continuous. It will be shown in Chapter 2 that a continuous state-space implies an infinite-dimensional continuous value function. Since this cannot be represented except in very special cases, approximations are clearly required for the general case. Solving POMDPs for robot navigation problems is therefore a game of approximations. One must strike the right balance between approximations which over-simplify the problem to the point where the robot is incapable of planning effectively, and approximations which do not simplify the problem enough, leaving it computationally intractable. This trade-off is illustrated in Figure 1.1.

The simplest approach is to choose actions based on the assumption that the most likely
pose is true, ignoring uncertainty. As an improvement, a number of heuristics for dealing with uncertainty have been devised. The details are discussed in Section 2.7.1, however suffice it to say that these approaches can only reason about the evolution of probability distributions over a very short planning horizon. In terms of Figure 1.1 they err on the side of over-simplification of the problem, producing plans which are not robust in the presence of uncertainty.

Another common approximation is to discretise the state, action and observation spaces. Having done so, a number of exact solution algorithms exist such that no further approximation is required [23]. Unfortunately, these exact algorithms are considered to be incapable of scaling to real-world problems in general, and will certainly not scale to the kinds of problems considered in this thesis. They lie on the opposite extreme of the spectrum depicted in Figure 1.1, representing an under-simplification of the problem.

After applying discretisation, a number of further approximations stem from the important insight that not all probability distributions are equally relevant. Figure 1.2 shows a hypothetical example of two probability distributions for a robot navigation problem. The belief shown in Figure 1.2(a) is certainly relevant, in that it is typical of the kinds of distributions that are expected to be encountered in practice. If the robot’s poor planning has not considered this belief, that poor planning is likely to be exposed. In contrast, Figure 1.2(b) shows an irrelevant belief. It is impossible, or at least highly unlikely, for this belief to occur. Hence, a robot which has specifically planned for this belief is unlikely to perform any differently from a robot which has ignored it.

This insight has been used in two ways:

1. to focus computation on a set of likely beliefs within a class, and
2. to restrict the class of beliefs which can be considered.

The first approach includes point-based methods which generate belief sets by model simulation [86][103][101]. Using some policy (e.g. random actions [103]), these methods simulate the repeated interaction of an agent with the POMDP model in order to generate a representative set of likely beliefs. Computation can then be focussed on these beliefs during planning. The hope is that the plan will generalise from this representative set of beliefs to all beliefs which are likely to be encountered in practice.
CHAPTER 1. INTRODUCTION

6

(a) A relevant belief

(b) An irrelevant belief

Figure 1.2: A relevant and an irrelevant belief. Beliefs are represented as particle sets, shown in blue. Particle density is proportional to probability density. The belief shown in (a) is relevant because similar beliefs are likely to occur during plan execution. Since it is impossible or at least highly unlikely that the belief in (b) will occur, a planner will only waste time by considering it. The beliefs are displayed on top of an occupancy grid [37]. Black denotes an occupied cell, white denotes an empty cell, and grey denotes unknown occupancy. Adapted from [94].

The second approach is to restrict the class of beliefs which can be considered. This includes approaches such as belief compression [94], which is based on dimensionality reduction. The space of all possible probability distributions over a discrete set of states is high-dimensional and continuous. Rather than allowing arbitrary beliefs, belief compression restricts itself to those which lie on a low-dimensional manifold embedded in that high-dimensional space. By choosing the manifold carefully, the set of beliefs which are likely to occur in practice will hopefully lie on or near that manifold.

The approach advocated in this thesis can be seen as a case of restricting the class of representable beliefs. The important difference from previous work is that we do not begin
CHAPTER 1. INTRODUCTION

with discretisation. Instead, we assume that beliefs can be approximated by continuous functions described by finite sets of parameters. The space of functions prescribes the class of beliefs to which the planner is restricted. If this space is chosen appropriately, it will hopefully be a good approximation to the kinds of beliefs which will occur in practice.

We will show that this difference has important ramifications, allowing us to scale to real-world problems. Most importantly, the use of parameterised continuous functions provides a compact representation of beliefs which does not rely on an underlying discretisation. For large problems, an underlying discrete representation is problematic. One must choose between a fine discretisation, which introduces scalability problems, and a coarse discretisation, with which one is unable to represent smooth gradients and small shifts in distributions.

In contrast, the use of continuous parameterised functions can handle smooth gradients and small shifts, but introduces a choice of function complexity. The use of overly complex functions may cause a problem to be intractable, whereas simple functions may constrain the shapes of beliefs too tightly, resulting in poor plans. We will show that for robot navigation problems, Gaussians represent a class of functions which are sufficiently simple to allow us to scale to large problems. At the same time they are sufficiently expressive to closely approximate the kinds of probability distributions which are usually, but not always, encountered during robot navigation. Section 3.1 will discuss the validity of this approximation in detail.

1.4 Application Domain

The particular application which this thesis works towards is the reliable operation of the autonomous mobile robot shown in Figure 1.3 in a large semi-structured outdoor environment. The robot senses the environment with a forward-looking laser range-finder, and senses its own motion with wheel encoders. We assume that the robot is given an a priori map of the environment.

The scenario considered here is particularly challenging for a number of reasons. Debris and holes in the asphalt make odometry particularly poor. Since the robots are statically unstable, they must pitch back and forth in order to balance, especially when accelerating or decelerating. This makes sensing with a fixed laser complicated. On the edge of the robots’ working area lies natural terrain for which an accurate geometric model would be
extremely complicated. In addition, the environment contains a number of hazards which could cause the robots to fall. These hazards lie below the plane of the laser, and are therefore essentially invisible to the robots.

Indeed, this is the most challenging robot navigation problem to which POMDP solution methods have been applied, by a significant margin. We would argue that Roy’s work represents the most challenging problem previously attempted [94]. This involved a simulated environment of a similar size. The problem was simplified by using an omni-directional sensor and ignoring the robot’s heading. The addition of heading is more realistic but much more challenging, adding an extra dimension to the problem. By excluding heading from the POMDP model, the robot’s actions must be specified in absolute terms, which is unrealistic unless the robot is somehow aware of its absolute heading during plan execution. Furthermore, the robot cannot be aware of the fact that sharp turns now increase uncertainty in the future. The simplification of using an omni-directional sensor means that the robot needn’t consider the fact that its ability to gather information depends on the direction in which it is travelling. POMDP solution methods which rely on this simplification are unable to
CHAPTER 1. INTRODUCTION

generalise to the arguably more common case of sensors which are not omni-directional. In contrast, our application includes heading and uses a forward-pointing sensor, implying that our robot must be able to account for all these details.

A solution to the decision-making problem relies on a robust solution to the localisation problem. That is, the robot must be able to use its uncertain sensor readings to work out where in the map it might be. While this is challenging, it is by no means unsolvable. Excellent progress has been made in this area over the last couple of decades by casting the problem as one of Bayesian estimation, and by applying approximations which make the problem computationally tractable in real-time [111]. Essentially, the robot maintains a probability distribution (or belief) over possible poses, and uses the actions and observations at each time interval to update this probability distribution.

Unfortunately, while the agent has access to these powerful methods of maintaining probability distributions, practical systems do not generally use the entirety of those distributions for decision-making. The standard approach is to assume that the most likely pose is in fact the true pose. To get from a start location to a destination, one can then apply any one of a number of deterministic path-planning algorithms which assume complete observability and deterministic actions [63][64]. This blind faith that the most likely pose is true can lead the robot to be overly confident, with potentially catastrophic results. POMDPs provide a framework for overcoming this problem by using the entire probability distribution, however the computational complexity of solving POMDPs has prohibited their widespread adoption. The following section outlines the contributions this thesis makes in order to address this issue.

1.5 Contributions

The contributions of this thesis are as follows:

- The presentation of a unifying view of several POMDP solution methods from the literature as specific instances of a more general solution method, namely the application of fitted value iteration in a particular information-space.

- The development and analysis of a novel planning algorithm, entitled PPOMDP, representing a specific instance of the general methodology defined above. PPOMDP
entails the application of fitted value iteration in the space of continuous parameterised functions.

- The formulation of an approach to estimating distributions over posterior parameterised beliefs using methods from the particle filtering literature. This novel approach is shown to scale independently from the size of the state-space, and hence is applicable to large, realistic planning problems.

- The formulation of a simplification of the planning problem using a factoring based on conditional independence assumptions. With certain approximations, this novel approach allows algorithms based on fitted value iteration to be broken into smaller components, reducing the total computational complexity.

- The presentation of a method for efficient function approximation for arbitrary sets of parameterised beliefs, using data structures from the similarity search literature.

- The efficient integration of local online forward planning into the PPOMDP framework, assisting offline global planning.

- The experimental evaluation of the PPOMDP approach in its various forms, and an experimental comparison against a state-of-the-art POMDP solution algorithm, in several simulated environments.

- The real-time implementation and experimental validation of the PPOMDP algorithm on a real robot navigating in a challenging environment. To the author’s knowledge, this work represents the most challenging robot navigation problem to which POMDP solution methods have successfully been applied to date.

### 1.6 Thesis Structure

Chapter 2 introduces basic MDP and POMDP terminology and concepts more formally, then reviews numerous solution algorithms which have been proposed in the literature. Chapter 3 introduces the basic concepts and solution algorithm behind the approach in this thesis, namely planning in the space of parameterised continuous functions. It argues for the use of Gaussian functions to approximate beliefs encountered in robot navigation problems. In order to compare approaches, Chapter 3 introduces BlockWorld: a simple continuous
CHAPTER 1. INTRODUCTION

navigation problem. We describe the details necessary to implement our algorithm for BlockWorld, then compare its performance against an MDP-based heuristic and a state-of-the-art point-based algorithm [103]. The results and algorithm presented in Chapter 3 are very similar to previously published work [19].

Chapter 3 serves as a foundation for the remaining chapters of this thesis. The algorithm presented in Chapter 3 has a number of deficiencies, in terms of both performance and scalability. Subsequent chapters maintain the same basic approach, but present improvements to both the quality of plans and the scalability of planning, to the point where the algorithm can operate competently in real environments. Each of Chapters 4 through 7 build on the algorithm as presented in the previous chapter by improving on a specific aspect, and present results on the BlockWorld problem to quantify that improvement.

Chapter 4 highlights some of the deficiencies of the Algorithm from Chapter 3. It suggests an improved and more general algorithm for projecting beliefs forward in time, and describes details of how it can be implemented efficiently. This improvement allows the algorithm to produce significantly better plans in approximately the same amount of time.

In Chapter 5, it will be shown how planning speed can be improved dramatically by pre-calculating the effects of observations. Essentially, the problem can be broken down into smaller components by a factoring based on conditional independence assumptions. This improvement produces similar results to the algorithm presented in the previous chapter, but in a fraction of the time.

Having already restricted the space of representable beliefs, Chapter 6 focusses computation on the important areas of that space. The algorithm requires a set of sample beliefs to plan over. Until this point, the algorithm has required that these lie on a regular grid over belief-space. In order to relax this requirement, a method is needed to efficiently retrieve the set of beliefs in the vicinity of a query belief. Chapter 6 reviews data structures from the similarity search literature and applies them to achieve this aim, resulting in a significant increase in scalability.

Chapter 7 introduces the final improvement. It shows how real-time online planning can be integrated with the offline planning algorithm described in previous chapters. Essentially, this allows an online agent to locally “fill in the gaps” of an occasionally coarse pre-computed global plan.
Chapter 8 moves beyond BlockWorld and applies the algorithm, with all its improvements, to a real problem. It describes the environment outlined in Section 1.4 in more detail, and explains how the algorithm presented in previous chapters can be applied to it. It presents results first on a toy simulated world with realistic dynamics, then on a simulated version of the real environment, and finally on a real robot operating in the real environment. Chapter 9 concludes and discusses future work.