Learning Search Strategies from Human Demonstrations

Dissertation (2016)

Submitted to the School of Engineering, Doctoral Program on Manufacturing Systems and Robotics

École Polytechnique Fédérale de Lausanne (EPFL)

in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by

Guillaume de Chambrier

Thesis Committee:
Prof. Aude Billard, thesis advisor

Lausanne, Switzerland
October, 2016

Chapter 1

Introduction

1.1 Motivation

Taking long term decisions or spontaneous reactive actions when presented with incomplete information or partial knowledge is paramount to the survival of any biological or synthetic entity. Reasoning given a state of uncertainty is a continuously occurring event throughout our livelihood. When considering long term decisions an abundance of examples come to mind. For instance, in economic investments uncertainty is to the best of efforts quantified and minimised in order to avoid unwarranted risks. Reactive actions are just as common; when looking for the snooze button of an alarm clock, early in the morning, our hand seems to autonomously search the surrounding space picking up sensory cues gradually acquiring information guiding us towards the button. All the above types of decision require the integration of evidence and an ability to predict the outcomes of the taken decisions in order to insure a favourable end state. Abilities close to these have met with mixed levels of success in Artificial Intelligence (AI) & robotics. There is a been noticeable success in artificial agents beating humans at board games (backgammon, chess and go) but having a robot successfully climb a staircase, open a door or pick up a glass are still ongoing open problems.

It is not yet fully understood how decisions are taken, yet alone under uncertainty. The difficulty is that two processes responsible for the synthesis of our actions and decisions, our beliefs and desires, are not directly or easily measurable. There is growing interest in Neuroscience to understand the mechanisms underlying perception and decision making under uncertainty Preuschoff et al. (2013); there is not yet a consensus on the biological mechanisms involved in decision making and efforts are ongoing[1] to construct plausible models of our decision processes. At a behavioural level, early efforts to model human decision making were made in mathematics & economics (Bernoulli (1954),Von Neumann and Morgenstern (1990)), in which gambles and investments were chiefly considered. There has been considerable effort in many fields (neuroscience, cognitive science, physiology, economics, etc..) to understand how decisions and actions are taken, starting with the role of our neurons to high level decisions

[1] the human brain project: https://www.humanbrainproject.eu/

Figure 1.1: Examples of the decision making under uncertainty in both robotics and everyday life situations. Images taken from the public domain.

like gambling, orientation and navigation problems to reflexes.

Artificial intelligence & robotics considered early on uncertainty in decision making, where the predominant domain of application was spatial navigation, Cassandra et al. (1996a). The problem has always been treated in two parts: the construction and representation of a world model (the map) and a planner which can reason with respect to this model in order to accomplish an objective. The world construction problem attracted a large amount of interest and has resulted in many successfully applications in a wide spectrum of robotic domains (AUV, UAV, etc..). The integration of planning with mapping in a single framework is still difficult to achieve and is based on either representing the decision problem as a Partially Observable Markov Decision Process (POMDP) which is notoriously difficult to solve for large scale problems, or through search heuristics. The mapping problem can generally be solved when assuming the uncertainty is Gaussian and thus quantifiable by a few parameters.

In summary there are still open problems in decision making when considering partial observability. The mapping problem has been studied and solved within a certain set of constraining assumptions. For the mapping problem we develop a Bayesian filter which is non-parametric and has no explicit representation of a joint distribution.

Currently, both humans and animals are far better at navigation than robots, especially when uncertainty is present, Stankiewicz et al. (2006). When addressing the decision making, we leverage human foresight and reasoning in a Learning from Demonstration (LfD) framework (Billard et al. (2008)), which is used to transfer skills from an expert teacher (usually a human) to a robot. Examples include the transfer of kinematic task constraints, stiffness and impedance

constraints and motion primitives, to name only a few.

In this thesis we address both problems under extreme levels of uncertainty.

1.2 Contribution

In this thesis we bring to light two main ideas. The first is the transfer of human behaviour to robots in tasks where a lot of uncertainty in present, making them difficult to solve using traditional techniques. The second is a non-parametric Bayesian state space filter which is efficient under sparse sensory information and high levels of uncertainty.

Throughout the work in this thesis we consider case studies in which vision is not available, leaving tactile and haptic information. This choice was made to induce a high level of uncertainty making it easier to study its effect on the decision making process. As a consequence the tasks we consider are by nature, haptic and tactile searches. The following three sections detail the contribution of this thesis to research decision making under sever uncertainty constraints.

1.2.1 Learning to reason with uncertainty as humans

A Markov Decision Process (MDP) allows the formulation of a decision problem in terms of states, actions, a discount factor and a cost function. Given this formulation and a suitable optimisation method (dynamic programming, temporal difference, etc..) a set of optimal decision rules are returned, known as a policy. The benefit of this approach is that the policy is non-myopic and sequences of complicated actions can be synthesised to achieve a goal which an opportunistic policy would fail to achieve. A Partially Observable Markov Decision Process (POMDP) is a generalisation of an MDP to a hidden state space and only observations are available relating to the state space. Finding an exact optimal solution to a POMDP problem is notoriously difficult due to the computational complexities involved. Sample based approaches to solve a POMDP rely heavily on a good trade-off between exploration and exploitation actions. Good explorative actions increase the chance of discovering a set of optimal decisions/actions.

In this thesis we propose a Learning from Demonstration approach to solving POMDP problems in haptic and tactile search tasks. Our hypothesis is that if we know the mental state of the human expert in terms of his believed location and observe his actions we can learn a statistical policy which mimics his behaviour. Since the human's beliefs are not directly observable we infer them by assuming that the way we integrate evidence is similar to a Bayesian filter. There is evidence both in cognitive and neuroscience that this is the case (Bake et al. (2011)). From observing the expert human performing a task we learn a cognitive model of the human's decision process by learning a generative joint

distribution over his beliefs and actions. The generative distribution is then used as a control policy. By this approach we are able to have a policy which can handle uncertainty similarly to humans.

1.2.2 Non-parametric Bayesian state space filter

Simultaneous Localisation and Mapping (SLAM) is concerned with the development of filters to accurately and efficiently infer the state parameters of an agent (position, orientation) and aspects of its environment, commonly referred to as the map. It is necessary for the agent to achieve situatedness which is a precondition to planning and reasoning. The predominant assumption in most applications of SLAM algorithms is that uncertainty is related to the noise in the sensor measurements. In our haptic search tasks there is no visual information and a very large amount of uncertainty. Most of the sensory feedback is negative information, a term used to denote the non event of a sensory response. In the absence of recurrent sightings or direct measurements of objects there are no correlations from the measurement errors which can be exploited.

In this thesis we propose a new SLAM filter, which we name Measurement Likelihood Memory Filter (MLMF), in which no assumptions are made with respect to the shape of the uncertainty (it can be Gaussian, multi-modal, uniform, etc..) and motion noise. We adopt a histogram parametrisation (this is considered non-parametric because a change in a parameter has a local effect). The conceptual difference between the MLMF and standard SLAM filters, such as the Extended Kalman Filter (EKF), is that we avoid representing the joint distribution since it would entail a shattering space and time complexity. This is achieved by keeping track of the history of measurement likelihood functions. We demonstrate that our approach gives the same filtered marginals as a histogram filter. In such a way we achieve a Bayes filter which has both linear space and time complexity. This filter is well suited to tasks where the landmarks are not directly observable.

1.2.3 Reinforcement learning in belief space

We propose a Reinforcement Learning framework for the task of searching and connecting a power plug to a socket, with only haptic. We previously addressed this setup by learning a generative model of the beliefs and actions with data provided by human demonstrations following the LfD approach. However, it is usually the requirement that the teacher is an expert, with few notable exceptions (Rai et al. (2013)). Since we were solely learning a statistical controller, both good and bad demonstrations will be mixed in together. By introducing a cost function representing the task we can explicitly have a quality metric of the provided demonstrations. In this way we can optimise the parameters

of our generative model to maximise the cost function. In this LfD Reinforcement Learning setup with a very simple cost function we can have a significant improvement of our a policy.

1.3 Thesis outline

The thesis is structured accordingly to the three main contributions outlined in the previous section, and all will have their individual chapter. We outline below the structure of the thesis.

Chapter 2 - Background

In this chapter we introduce and mathematically formalise the sequential decision making problem under uncertainty and we provide a detailed literature review of the related work in this domain. We provide a brief introduction to *Decision Theory* before focusing on the work in AI & robotics relevant to POMDPs whilst highlighting their relevance and contribution to our work.

Chapter 3 - Learning to reason with uncertainty as humans

In this chapter we present an approach for transferring human skills in a blind haptic search task to a robot. The belief of the human is represented by a particle filter and all subsequent beliefs are inferred from the human's motions acquired via a motion tracking system. A generative model of the joint belief and actions distribution is learned and used to reproduce the behaviour on a WAM and KUKA robot in two search tasks. Experimental evaluations showed the approach to be superior to greedy opportunistic policies and traditional path planning algorithms. The major parts of this chapter have been presented de Chambrier and Billard (2014). We also provide a review of work related to humans taking decisions under uncertainty in spatial navigation and haptic tasks with an emphasis on works which consider diminished or no visual information.

Chapter 4 - Non-parametric Bayesian state space filter

In this chapter we present an approach to perform a state space estimation of a map and agent given that there is no direct observation between the landmarks and the agent. We demonstrate that by not explicitly parametrizing the full joint distribution of the landmarks and agent but instead keeping track of the applied measurement functions we can fully reconstruct the optimal Bayesian state estimation. The advantage of our approach is that the space complexity is linear as oppose to exponential. We validate our approach in 2D search navigation tasks. This work is currently under review. We also give an overview of the literature of SLAM and emphasis the position of our filter within it.

Chapter 5 - Reinforcement learning in belief space

In this chapter we present an approach similar to the one presented in Chapter 3, "Learning to reason with uncertainty as humans", with the difference that we explicitly encode the task through the introduction of a binary objective function and we consider a peg-in-hole task under high levels of uncertainty. The task requires both high and low levels or precision to be able to accomplish it, which makes it particularly interesting. We learn a value function approximation of the belief space through locally weighted regression and approximate dynamical programming. By combining a LfD approach in this Actor-critic Reinforcement Learning framework, we demonstrate an improvement upon a purely statistical controller with nearly no additional cost. We additionally provide a review of RL methods in the context of POMDPs.

Chapter 6 - Conclusion

We conclude by providing a holistic summary of our work and achievements. We draw attention to the current open problems and directions for future work in field of uncertainty and reasoning in Artificial intelligence and robotics.

Chapter 2

Background

Planning and reasoning under uncertainty is central to AI and robotics and has been an active area of research for decades. Planning and reasoning under uncertainty is an umbrella term in which a wide spectrum of fields study its aspect: *economics*, *psychology*, *cognitive science*, *neuroscience*, *robotics* and *artificial intelligence*. The work in this thesis relies on results from all of the aforementioned fields. Cognitive and neuroscience bring justification and insight into the way we represent our beliefs and how we act accordingly. AI and robotics provide computational models and optimisation methods which take into account insights attained in the sciences. Because of the vast spectrum of topics we cannot do justice to all them and we will focus on works which are directly relevant to the problems we are addressing in this thesis.

In this chapter we cover the following topics in the presented order: Decision Theory (DT), Markov Decision Process (MDP), Partially Observable Markov Decision Process (POMDP), a literature review and the approach taken in this thesis. See Figure 2.1 for an illustration of the outline.

Figure 2.1: Chapter outline

- **Section** 2.1, introduces what is meant by taking decisions under uncertainty and what are the different sources of uncertainty. We take a historical look at Decision Theory since it is the root node of all subsequent research in reasoning and acting under uncertainty and provides for a good introduction to the topics which will follow.
- **Section** 2.2, mathematically formalises the sequential decision problem under uncertainty and make the link with Decision Theory. We derive from first principal the Bellman optimal equation which is probably the most important result to date in the field.
- **Section** 2.3, provides an in depth literature review with the latest results in AI & robotics in the subject of planning and acting under uncertainty. We draw attention to the different approaches to solving this problem whilst pointing out their advantages and weaknesses.

- **Section ??**, we provide a summary of what has been achieved so fare and how this thesis contributes and complements the field.

2.1 Decisions under uncertainty

The main objective of reasoning under uncertainty is to find an action or sequence of actions which will result in the most preferable outcome. There are two key attributes which can render this problem difficult: **stochastic actions** and **latent states**.

Stochastic actions when applied in the same state will not always result in the same outcome. This type of uncertainty can arise from many sources. For instance, the outcome of chaotic systems will always lead to different results when the same action is applied to the same initial conditions, such as the throwing of a dice or the flipping of a coin. In outdoor robotics the terrain might lead to slippage, causing the robot to skid, or in an underwater environment currents might drastically offset the position of an UAV. In articulated robots, the friction in the joints can result in an error in the end-effector position (especially true for cable driven robots).

The second source of uncertainty is when the state space cannot be determined. This arises when the sensors are not able to provide sufficient information to reliably estimate the state. In robotics this uncertainty can arise from inadequate or noisy sensors. In poor environmental conditions such as humidity, lack of light or smoke the robot can experience difficulties in ascertaining its position and thus in planning how to achieve a given objective.

Given these two types of uncertainty, the question is how to represent these uncertainties. The predominant approach is to quantify the uncertainty in terms of probabilities. For instance the application of a forward action to a wheeled robot will result in some probability in a new position further ahead and with a remaining probability distributed to adjacent regions which might have occurred due to slippage.

An observation made through the robot's sensors will result in a probability distribution over the robot's probable location. This quantification of the action and observation uncertainty, in terms of a probability distribution over the state, must be utilised by the agent to plan actions towards accomplishing its goal. In order to take a decision, the agent must assign a utility to each state weighted by the probability of its outcome and act so as to get the highest utility. The utility indicates a preference over the outcomes and when combined with probabilities leads to Decision Theory, which is the topic of the next section.

2.1.1 Decision theory

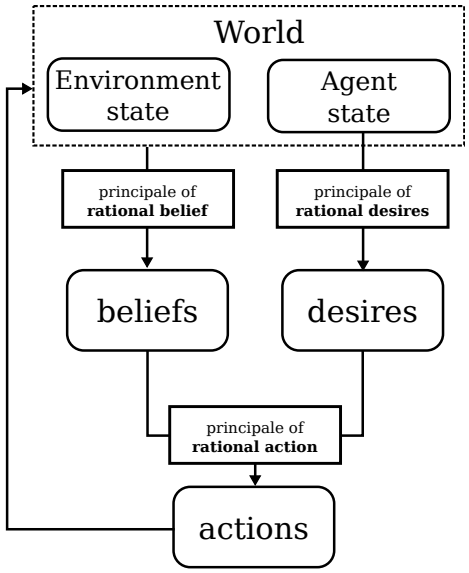


Figure 2.2: Relation between beliefs, desires and actions and are all considered to be rational.

The central question that Decision Theory asks is: *how do we take decisions when faced with uncertain outcomes ?* To answer such a question we need to identify the attributes which are involved when we take a decision, namely our **beliefs** and **desires**. Beliefs reflect a degree of knowledge we have about the world. This degree is ascertained by the amount of evidence we have in support of our beliefs. Epistemology studies in great detail the relationship between truth, beliefs and knowledge. We will not go into a philosophical discussion of their interplay, but make use of the following: if we have sufficient evidence in support of our beliefs and they represent the truth then we consider them to be **rational beliefs**. As for desires, they are linked to our disposition to take upon them. For example if I want to switch off my alarm clock I have to look for it in the last area I believed it to be in. These two attributes, beliefs and desires, are used to frame a decision problem. Early work in decision theory assumed that the problem was well grounded and focused on finding what **rational actions** need to be taken given our beliefs in order to achieve our desires.

Early interest in such questions were typically centred around economics which included deciding an appropriate investment or wager for a particular gamble. It was noted that the expected monetary outcome of a gamble as a means of basing a decision, would often lead to a course of action which contradicts common sense. A famous example of this contradiction is demonstrated in the St. Petersburg paradox. In this paradox a bookmaker proposes the following gamble. An initial pot starts with a content of £2. The bookmaker proceeds to flip a fair coin until the first appearance of a tails which ends the game. Until the occurrence of the first tails the money in the pot doubles after every toss. Once the game ends the player leaves with the contents of the pot. As an avid

gambler and **expected value** maximiser how much would one be willing to pay to enter this game ? To access, one would need to know the average payout. The amount of money increases by $£2^n$, where n is the number of non-final tosses and the probability of reaching n is $1/2^n$. In this case the expected monetary outcome is infinite:

$$\underset{p(£)}{\mathbb{E}}\{£\} = \underbrace{\frac{1}{2}£2}_{\text{first toss}} + \frac{1}{4}£4 + \cdots = \sum_{n=1}^{\infty} £\frac{2^n}{2^n} = £\infty$$

So the expected gain or return for paying to enter such game is an infinite amount of money. Thus in principal if a player was seeking to maximise his expected return value he would be willing to pay an amount close to infinity to enter the game. This does not seem a good decision rule; no person in the world would be willing to pay such high amounts to enter this game.

Nicolas Bernoulli proposed a solution to the problem which was later published by his brother Daniel (republished Bernoulli (1954)). He introduced the notion of a **utility function**, and he claimed that people should base their decision on the expected utility instead of solely on the monetary outcome.

> "...the value of an item must not be based on its price, but rather on the utility it yields."
>
> — Daniel Bernoulli

The introduction of a utility function takes into account that the net worth of a person will influence their decision since different people (in terms of their monetary worth) will weigh the gain differently. The utility function introduced by Bernoulli was the logarithm of the monetary outcome $x \in X$ weighted by its probability $p(x)$ which results in an expected utility:

$$U(x) = \mathbb{E}\{u(x)\} = \sum_{x \in X} p(x) \underbrace{\log(x)}_{u(x)}$$

It is later in 1944 that von Neumann and Morgenstern (Von Neumann and Morgenstern (1990)) axiomised Bernoulli's utility function and proved that if a decision maker has a preference over a set of lotteries[1] which satisfy four axioms (completeness, transitivity, continuity, independence) then there exists a utility function whose expectation preserves this preference. An agent whose decisions can be shown to maximise the vNM expected utility are said to be **rational** otherwise they are **irrational**.

This is the theoretical basis of most economic theory. It is a **normative** model of how people should behave given uncertainty. It is also the basis of most if not all decision making, cogitative architectures and control policies in AI and robotics (to the best of the author's knowledge).

[1] the term lottery refers to a probability distribution in the original text.

An aspect to keep in mind regarding the vNM model is that it is normative; it states what should be a rational decision. As a result it is not always consistent with human behaviour. There is great debate regarding the predictions made by vNM models with respect to our behaviour. There have been many studies both demonstrating divergence between the model's predictions and our observed behaviour but also supporting evidence that it does reflect the output of our decision making process. Reasons for divergence have been attributed to how people weigh probabilities and how the decision problem is framed. But probably the most important aspect is that in most decisions we are faced with, the quantification and rationality of our beliefs might not be adequate and limitations of our working memory will come into play in the final decision.

Nevertheless vNM agents are predominantly used in AI and robotics as a means of implementing decision making processes or in control policies. In psychology and cogitative science vNM agents are a used for comparing human behaviour against an optimal strategy (by optimal we mean it is rational in the vNM sense). It is important to remember the origins and assumptions underlying the models that are used to represent control policies or cognitive architectures implemented into robotic systems or software agents.

2.2 Sequential decision making

When Decision Theory is brought up, we are usually referring to a one shot non-temporal decision. However many interesting decision problems are sequential. In such situations, we must consider the effect current decisions will have on future decisions. Expected utility theory (part of Decision Theory) is extendable to a temporal decision problem. There are however two subtle but important differences between the temporal and non-temporal decision problems. The first difference is the utility. In the one time step problem an outcome has one utility assigned to it, $u(x)$. In the temporal decision problem a utility has to be assigned to a sequence of outcomes, $u(x_{0:T})$, where T is the number of sequential decisions taken. The utility of a sequence is the sum of the individual utilities. However if the decision problem is non terminating this will lead to an unbounded utility. To bound the utility a discount factor $\gamma \in [0, 1)$ is introduced and the new temporal utility function becomes:

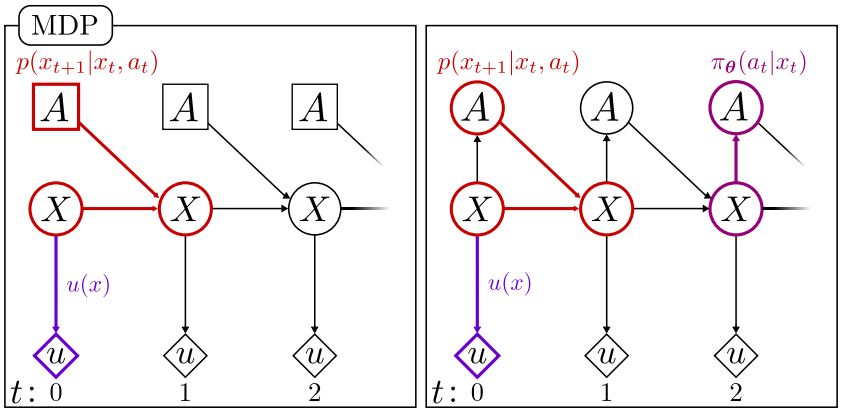
$$u(x_{0:T}) := \sum_{t=0}^{T} \gamma^t u(x_t) \tag{2.1}$$

The discount factor controls the importance that later utilities have on the final utility. If the discount factor is set to zero we obtain the original one shot utility function and if we were to take actions which maximised the expected utility we would not be considering at all the effect current decisions have at

Notation	Definitions
$x_t \in \mathbb{R}^3$	Cartesian state space position of the agent.
$y_t \in \mathbb{R}^M$	Observation/measurement from the agents sensors.
$a_t \in \mathbb{R}^3$	Action, usually the Cartesian velocity of the end-effector of the agent.
X, Y, A	State, observation and action random variables where x, y and a are realisation.
$p(x_t)$	Short hand notation for a probability density function, $p_X(x_t)$.
$x_{0:t}$	$\{x_0, x_1, \cdots, x_{t-1}, x_t\}$, history up to time t.
$p(x_t \vert y_{0:t}, a_{0:t})$	Filtered probability distribution over the state space given the action and observation history.
$b_t \in \mathbb{R}^L$	Belief state, a function of the filtered distribution $b(p(x_t \vert y_{0:t}, a_{0:t}))$ which will be written as b_t for simplicity.
$\pi_{\boldsymbol{\theta}}(a_t \vert \cdot)$	Probabilistic policy, $a_t \sim \pi_{\boldsymbol{\theta}}(a_t \vert \cdot)$
$u(x) \in \mathbb{R}$	Utility function, returns the utility of being in state x. It can also be dependent on the action, $r(x, a)$.
$\gamma \in [0, 1)$	Discount factor, the closer to one the more the later utilities are considered. When set to zero, only immediate rewards are considered which would result in a myopic greedy agent.
$p(x_{t+1} \vert x_t, a_t)$	State transition function, returns the likelihood/probability of reaching state x_{t+1} given that action a_t is applied in state x_t.
$p(y_t \vert x_t)$	Observation/measurement model, returns the likelihood/probability of observing y_t given that the agent is in state x_t.
$\tau(b_{t-1}(x), u_{t-1}, y_t)$	Updates a belief given a motion and observation. It makes use of both the motion and observation functions. The state space estimation function, τ, can be any kind of state space filter such as an Extended Kalman Filter (EKF) or a Particle Filter (PF).

Table 2.1: Definition of common variables used.

future decision points. An agent reasoning in such a way is called **myopic**. The second difference between the temporal and non-temporal decision problem is the way in which probabilities are assigned to outcomes. This was $p(x)$ in the Decision Theory utility function formulation. Now because of the sequential nature of the problem we consider a conditional state transfer probability distribution $p(x_{t+1}|x_t, a_t)$ which models the probability of going from state x_t to x_{t+1} given that action a_t is taken. This particular representation of a sequential decision problem is called a **Markov Decision Process (MDP)** and to be more exact a first order MDP. The necessary models are the state transition and utility functions. The assumption of such a model is that all necessary information to take a decision is encoded in the current state and there is no need to consider the history of state transitions when taking a current decision. In Figure 2.3 we illustrate two graphical representations of a MDP, which are known as **Dynamic Bayesian Networks (DBN)**. A DBN represents the temporal relationship and conditional dependence between random variables, decisions and utilities, which are represented by circles, squares and diamonds. For the MDP to the left the actions are not stochastic, whilst for the MDP on the right the actions taken are governed by a stochastic **policy**, $\pi_{\boldsymbol{\theta}}(a_t|x_t)$. A policy represents the plan of an agent for each state, given a state it will output an action. A policy is considered optimal when it maximises the expected utility function, it is optimal in the vNM sense.



(a) off-policy **(b)** on-policy

Figure 2.3: Dynamical Bayesian Network of a Markov Decision Process; it encodes the temporal relation between the random variables (circles), utilities (diamond) and decisions (squares). The arrows specify conditional distributions. In **(a)** the decision nodes are not considered random variables whilst in **(b)** they are. From these two DBN we can read off two conditional distributions, the state transition distribution (in red) and the action distribution (in purple).

Solving a MDP means finding a policy whose actions in any given state will always maximise the expected utility. Such a policy is usually denoted as π^*, the **optimal policy**. As in decision theory, the expected utility is the utility of a sequence of states $u(x_{0:T})$ weighted by its probability. The graphical representation (Figure 2.3 (a)) allows the probability of a sequence of states and

actions, to be read off directly:

$$p(x_{0:T}, a_{0:T-1}) = p(x_0) \prod_{t=0}^{T-1} p(x_{t+1}|x_t, a_t) \tag{2.2}$$

$$u(x_{0:T}) = u(x_0) + \gamma u(x_1) + \cdots + \gamma^{T-1} u(x_{T-1}) + \gamma^T u(x_T) \tag{2.3}$$

We are interested in finding the sequence of actions, $a_{0:T}$, which will maximise the expected utility function:

$$\underset{a_{0:T-1}}{\operatorname{argmax}} U(x_{0:T}, a_{0:T-1}) = \max_{a_0} \sum_{x_1} \cdots \max_{a_{T-1}} \sum_{x_T} \left(p(x_{0:T}, a_{0:T-1}) u(x_{0:T}) \right) \tag{2.4}$$

Solving the above directly in its current form would to lead to an exponential complexity. Making use of the first order Markov assumption and that current utilities do not depend on future utilities, the summations can be re-arranged and a recursive pattern emerges which can be exploited:

$$\underset{a_{0:T-1}}{\operatorname{argmax}} U(x_{0:T}, a_{0:T-1}) = \max_{a_0} \sum_{x_1} \cdots \max_{a_{T-2}} \sum_{x_{T-1}} p(x_{0:T-1}, a_{0:T-2})$$
$$\left(u(x_{0:T-2}) + \gamma^{T-1} \left(u(x_{T-1}) + \gamma \max_{a_{T-1}} \sum_{x_T} p(x_T|x_{T-1}, a_{T-1}) u(x_T) \right) \right) \tag{2.5}$$

From the rearrangement we notice that Equation 2.5 has the same functional form as Equation 2.4, except that the recursive component can be summarised by Equation 2.6, which is known as the **Bellman** optimal equation (the asterisk indicating that it is optimal),

$$V^*(x_t) := u(x_t) + \gamma \max_{a_t} \sum_{x_{t+1}} p(x_{t+1}|x_t, a_t) V(x_{t+1}) \tag{2.6}$$

where for the terminal state $V_T(x_T) = u(x_T)$. The Bellman equation is a means of solving a sequential decision problem through use of dynamic programming. It shows that the utility of the current state is based on the immediate utility and the discounted maximum utility of the next state. Making use of this recursion reduces the computation complexity which is now quadratic in the number of states, $\mathcal{O}(T\,|A|\,|X|^2)$. To find the optimal value and subsequent policy an approach would be to repeatedly apply the Bellman equation to each state until the value function converges. What makes the problem difficult to solve is maximisation over the actions. This induces two problems, the first is that the optimisation is nonlinear and the second is that if the action space is continuous the maximisation will be expensive to compute. This brings into play the two main approaches to solving a MDP: **off-policy** and **on-policy**. Off-policy methods solve directly for the optimal value function, $V^*(x)$, and perform the maximisation over the actions. **Value-Iteration (VI)** is such a method. On-policy approaches, $V^\pi(x)$, find the optimal value and policy

through repeating **policy evaluation** and **improvement** steps. In the policy evaluation the value or utility of a policy is found through solving the on-policy version of the Bellman equation:

$$V^{\pi}(x_t) := u(x_t) + \gamma \sum_{a_t} \pi_{\boldsymbol{\theta}}(a_t|x_t) \sum_{x_{t+1}} p(x_{t+1}|x_t, a_t) V(x_{t+1}) \tag{2.7}$$

In the policy improvement step, the policy is made more greedy by maximising the value function. Through the repetition of these two steps both the value function and policy converge to the optimal. On-policy methods are preferred in settings where the action space is highly continuous, such as in robotics. Using dynamic programming is however not the method of choice since it requires multiple passes through the entire state space and for this reason it is necessary to have the model of the state transition a priori. Instead **Reinforcement Learning (RL)** methods are used to find an optimal value and policy. RL is a sample based approach in which an agent interacts with the environment gathering examples of state transitions and the utility and uses them to gradually solve the Bellman equation.

We introduced the formulation of a sequential decision process for the MDP model and showed how an optimal policy and value function are obtained through maximising the expected utility. The re-arrangement of the summations, known as variable elimination, allows to exploits a recursive structure present in the Markov chain. The recursive component turns out to be the Bellman optimal equation, which when solved (via dynamic programming or reinforcement learning) results in an optimal value and policy function. A MDP models the uncertainty inherent in the state transition but not the uncertainty of the state. The MDP assumes that the state space is always fully observable, which is a strong assumption. In robotics, the on board sensors return an estimate of the state with a certain amount of uncertainty associated with it. To take this additional uncertainty into consideration the MDP has to accommodate it. This leads to a Partially Observable Markov Decision Process (POMDP).

2.2.1 POMDP

A POMDP is a popular approach for formulating a sequential decision process in which both motion and observation uncertainty are considered. In this partially observable setting the agent does not know with exactitude the state of the environment, but is able to observe it through his **sensors**. We define a sensor mathematically as being a function of the state space, x_t, relating to an observation, y_t, corrupted by some noise, ϵ_t,

$$y_t = h(x_t) + \epsilon_t \tag{2.8}$$

The sensor function $h(x_t)$ can be linear or non-linear and the additive noise term ϵ_t can be Gaussian (usually the case), non-Gaussian, state dependent or not. The uncertainty of the latent state, x_t, is quantified by a probability distribution, $p(x)$. This probability distribution represents all the hypothetical positions in the world in which the agent can be found. In Figure 2.4 **(a)** an agent is located in a square yard containing a wall. Initially the agent is confident of his position; his state uncertainty $p(x_0)$ is low, represented by the blue probability density. However during a circular displacement the agent skids and the state uncertainty is increased by the state transition function, $p(x_{t+1}|x_t, a_t)$; this step is referred to as **motion update**. To reduce the uncertainty, the agent takes a measurement, y_t, with his sensors which provide range, r, and bearing, ϕ, information with respect to the wall, see Figure 2.4 **(b)**. The agent uses the model of his sensor, known a priori, to deduce all possible locations in the world from where the current measurement could have originated. This model is known as the measurement likelihood function:

$$p(y_t|x_t) = \mathcal{N}(y_t - h(x_t); 0, \Sigma) \tag{2.9}$$

The measurement likelihood function makes use of the measurement function $h(x)$ and it models the noise in the sensor. In this case the noise model, ϵ_t, is Gaussian, paramaterized with mean zero and covariance Σ. Typically the parameters of the measurement likelihood function are learned a priori.

In Figure 2.4 **(c)** the likelihood is illustrated. The dark regions indicate areas of high likelihood, which are possible locations from which the sensor measurement could have originated. The value of the measurement likelihood function is then integrated into the state space probability density function; this step is referred to as **measurement update**.

The two update steps, motion and measurement, are part of a recursive state estimation process called a **Bayesian state space filter**, which we formalise below in Equation 2.10-2.11.

The motion model, Equation 2.10, updates the position of the probability distribution according to the applied action, a_t, and adds uncertainty by increasing the spread of the distribution. The measurement information is then incorporated by Equation 2.11. The measurement likelihood always reduces the uncertainty or leaves it constant. The Bayesian state space filter is such an important component to belief space decision making that we define it by the filter function, $\tau(b_t, a_t, y_t)$, which takes as input the current belief, applied action and sensed measurement and returns the resulting belief b_{t+1}. The state space filter is an essential component to a POMDP which will become apparent later.

With the latent state, its relation to the observation variable and the Bayesian filter defined, we can introduce the POMDP model in Figure 2.6 (*left*). It has the same Markov chain structure as the MDP, introduced in the previous section, but the state space X is latent and a new layer of observation variables Y

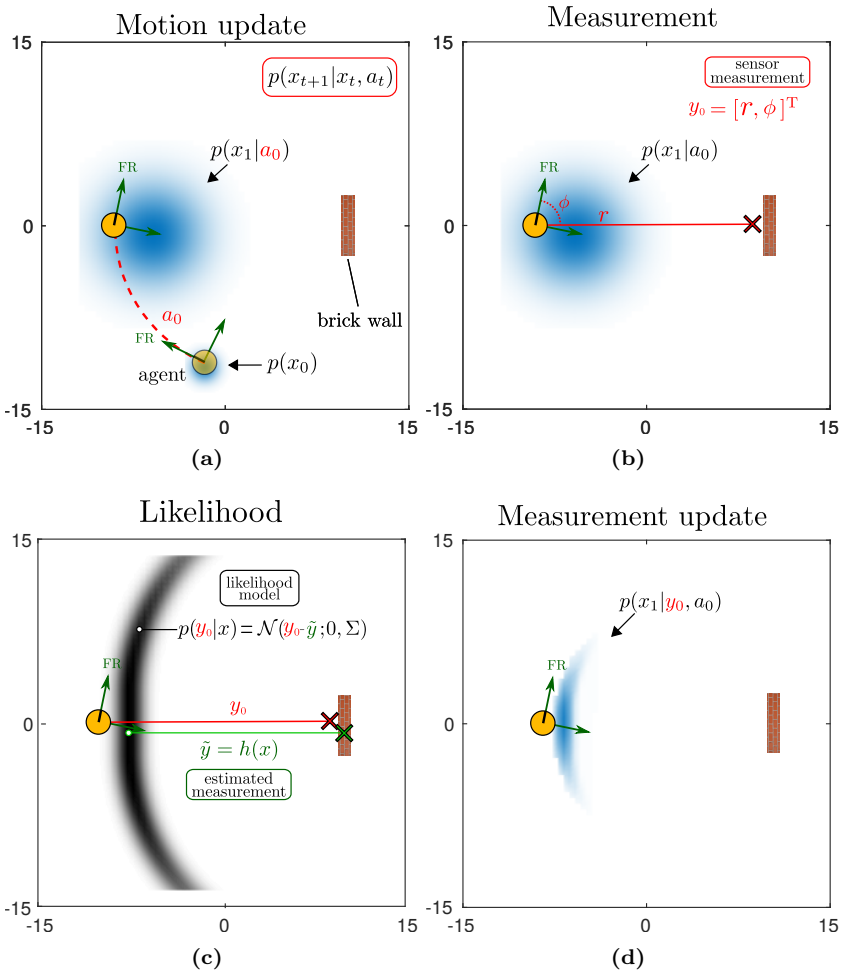


Figure 2.4: **(a)** An agent is located to the south west of a brick wall. It is equipped with a range sensor. The agent takes a forward action but skids, which results in a high increase of the uncertainty.**(b)** The agent takes a measurement, y_0, of this distance to the wall; because his sensor is noisy his estimate is inaccurate. **(c)** The agent uses his measurement model to evaluate the plausibility of all locations in the world which would result in a similar measurement; illustrated by the likelihood function $p(y_0|x_0)$. **(d)** The likelihood is integrated into the probability density function; $p(x_0|y_0) \propto p(y_0|x)p(x_0)$.

Bayesian filter

The Bayesian filter turns a prior probability distribution over the state space, $p(x_t|y_{0:t-1}, a_{0:t-1})$, to a posterior $p(x_t|y_{0:t}, a_{0:t})$ by incorporating both motion and measurement. Applied recursively it keep a probability distribution over the state space which considers all the past history of actions and observations. We define the application of these two steps by the filter function τ, which takes the current belief, the applied action and measurement, and outputs the next belief, b_{t+1}.

Motion update

$$p(x_t|y_{0:t-1}, a_{0:t}) = \int p(x_t|x_{t-1}, a_{t-1})\, p(x_t|y_{0:t-1}, a_{0:t-1})\; da_{t-1} \tag{2.10}$$

Measurement update

$$p(x_t|y_{0:t}, a_{0:t}) = \frac{1}{p(y_t|y_{0:t-1}, a_{0:t})} p(y_t|x_t)\, p(x_t|y_{0:t-1}, a_{0:t}) \tag{2.11}$$

$$p(y_t|y_{0:t-1}, a_{0:t}) = \int p(y_t|x_t)\, p(x_t|y_{0:t-1}, a_{0:t}) dx_t \tag{2.12}$$

Filter function

$$b_{t+1} := \tau(b_t, a_t, y_t) \tag{2.13}$$

Figure 2.5: Bayesian state space filter.

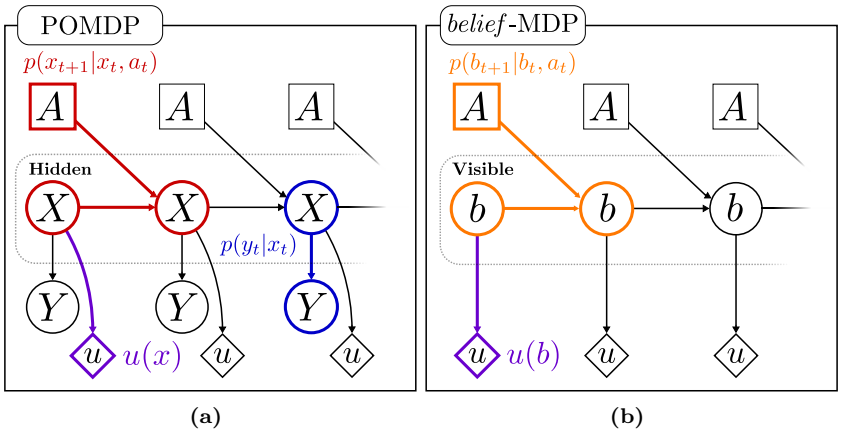


Figure 2.6: **(a)** POMDP graphical model. The state space, X, is hidden, but is still partially observable through a measurement, Y. **(b)** The POMDP is cast into a belief Markov Decision Process, belief-MDP. The state space is a probability distribution, $b(x_t) = p(x_t)$, (known as a belief state) and is no longer considered a latent state. The original state transition function $p(x_{t+1}|x_t, a_t)$ is replaced by a belief state transition, $p(b_{t+1}|b_t, a_t)$. The reward is now a function of the belief.

is added.

As the state space is only partially observable the expected utility has to be computed for each possible history of states, actions and observations. All approaches in the literature instead encapsulate all these possible histories into a belief state $b(x_t)$ (for short notation b_t) which is a probability distribution (also referred to as an information state, I-state) over the state space x_t and use this new state description to cast the POMDP into a **belief-MDP** (states are probability distributions, beliefs). By casting a POMDP into a *belief*-MDP the state space is considered observable and we recover the same structure as in the standard MDP problem.

As we are working within a belief space the reward function has to be adapted to:

$$u(b_t) = \sum_{x_t} u(x_t)\, b(x_t) = \mathbb{E}_{b_t}\{u(x_t)\} \tag{2.14}$$

which is an expectation. The goal as before is to find a sequence of actions which will maximise the expected utility. Since our *belief*-MDP has the same structural form as the MDP, the solution to the problem is the same Bellman equation derived previously. We just substitute the new belief transition function and we get the corresponding belief Bellman Equation, 2.15.

$$V^*(b_t) = u(b_t) + \gamma \max_{a_t} \sum_{b_{t+1}} p(b_{t+1}|b_t, a_t)\, V^*(b_{t+1}) \tag{2.15}$$

However, using this equation in this form is problematic, as we are summing over the space of beliefs (which is high dimensional and infinite for the continuous case) and the transition function is a probability distribution over beliefs. The

key to overcome this problem is to realise that if we know what the current measurement and applied action are, there is only one valid possible belief, b_{t+1}, and the summation over beliefs vanishes. This can be seen by substituting the belief transition function, Equation 2.16, into the Bellman equation Equation 2.15.

$$p(b_{t+1}|b_t, a_t) = \sum_{y_t} p(b_{t+1}|b_t, a_t, y_t)\, p(y_t|y_{0:t-1}, a_{0:t}) \tag{2.16}$$

After the substitution and re-arrangement of the summation we get Equation 2.17. Since the observation is known (because the outer summation is over y_t), the summation over the beliefs vanishes since there is only one possible future belief which is given by the Bayesian filter function $b_{t+1} = \tau(b_t, a_t, y_t)$,

$$u(b_t) + \gamma \max_{a_t} \sum_{y_t} \underbrace{\left(\sum_{b_{t+1}} p(b_{t+1}|b_t, a_t, y_t)\, V^*(b_{t+1}) \right)}_{1 \cdot V^*(\tau(b_t, a_t, y_t))} p(y_t|y_{0:t-1}, a_{0:t}) \tag{2.17}$$

which simplifies to:

$$\begin{aligned} V^*(b_t) &= u(b_t) + \gamma \max_{a_t} \sum_{y_t} p(y_t|y_{0:t-1}, a_{0:t})\, V^*(\tau(b_t, a_t, y_t)) \\ &= u(b_t) + \gamma \max_{a_t} \mathop{\mathbb{E}}_{y_t} \{V^*(b_{t+1})\} \end{aligned} \tag{2.18}$$

The belief Bellman equation is intuitive. The value of the current belief is the immediate utility plus the value of the future belief states weighted by the probability of a measurement which would result in these future belief states. An exact solution exists only when considering a finite state, action and observation space and a finite planning horizon T, Richard D. Smallwood (1973). The belief-MDP can be solved with value iteration but each backup operation (application of the bellman equation) results in an exponential growth in the number of parameters needed to represent the value function, which is computationally intractable.

Most early techniques for solving POMDPs used value iteration. The preference for persisting in doing this, given the computational burden, is that since the utility function uses a linear operator (the expectation) and that the Bellman backup operation (applying the Bellman equation to the current value function) preservers the linearity, the value function after each updates is Piece Wise Linear and Convex (PWLC). A good text on the implementation of exact value iteration for POMDPs can be found in (Thrun et al., 2005, Chap. 15) and Kaelbling et al. (1998).

In summary there are two problems in solving a POMDP:

- **curse of dimensionality**: A discrete state space of size N will result in a belief space of dimension $N - 1$. The discretization choice will greatly impact the computational cost of Value Iteration.

- **curse of history**: The space and computational complexity in the worst case is exponential with respect to the planning horizon, T, Du et al. (2010).

Given such complexity it is hard to see POMDPs being actually usable for real world scenarios. As a result many approximate techniques have emerged with some being very successful. In the next section, we survey the literature and the developments of approximate POMDP algorithms and their applications.

P-SPACE Papadimitriou and Tsitsiklis (1987) complete

2.3 Literature review

We review the latest methods of solving sequential decision problems under uncertainty. This is an extremely dense and spread out area of research, no doubt because of its importance. If uncertainty is not considered adequately, the control policy risks being suboptimal or lead to drastic failure.

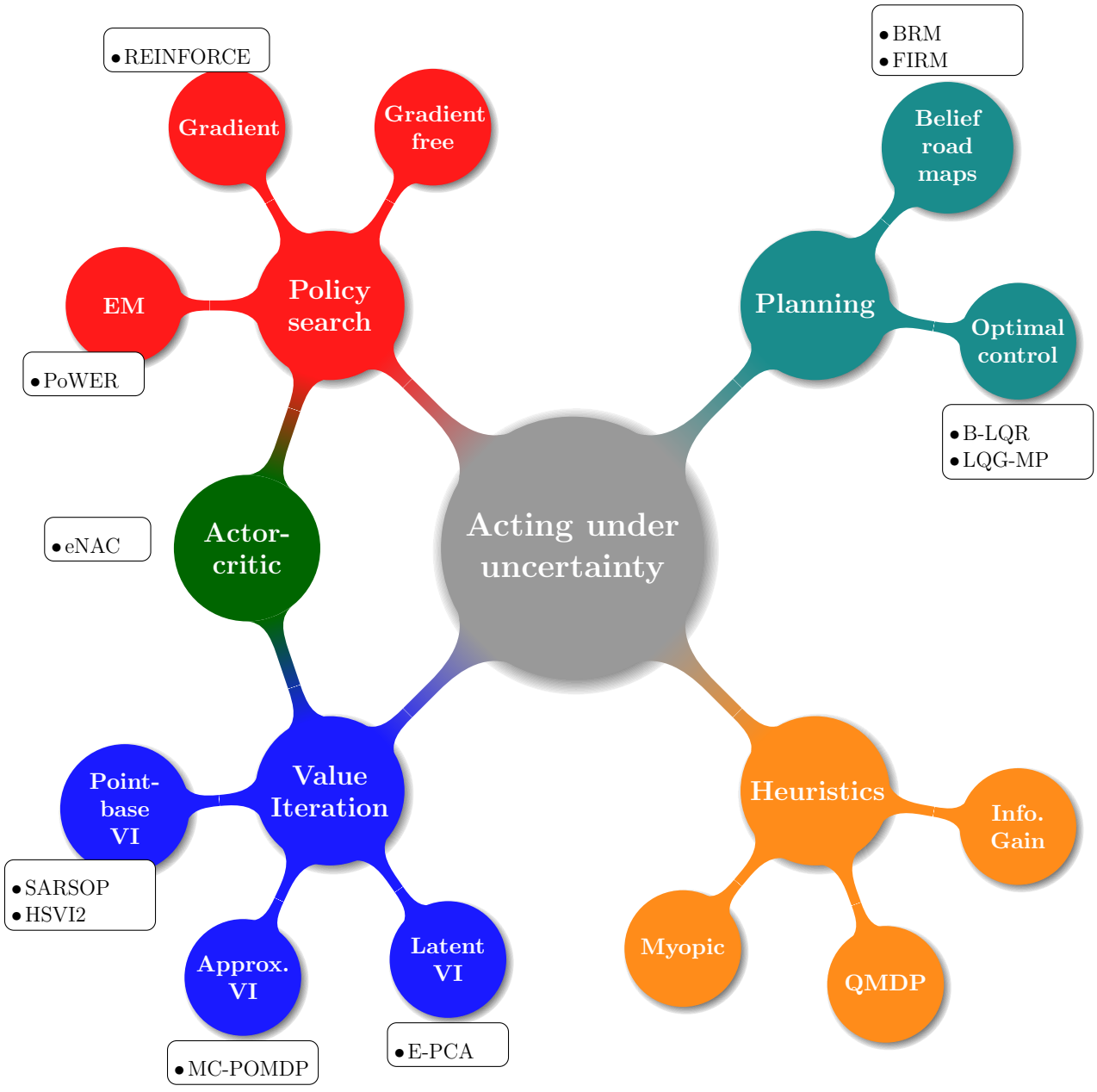


Figure 2.7: Mind-map of AI and robotic methods for acting under uncertainty.

2.3.1 Value Iteration

The POMDP formulation introduced previously is the main theoretical starting point of policies which consider uncertainty optimally. However solving an exact POMDP through dynamic programming (value iteration) is computationally intractable and an exact solution only exists for discrete state, action and observation space (Thrun et al., 2005, Chap. 15). This intractability, in which only problems with a few states could be solved has inhibited the application of the POMDP framework to robotics.

Point-base Value Iteration

The first breakthrough of the application of VI in belief space to a robotic application was Point-Base Value Iteration (PBVI) Pineau et al. (2003). It allowed VI to be applied to a robotic navigation problem consisting of 626 states in a hospital patient search task. The key insight to scale VI was to only consider a subset of belief states which were reachable and relevant to the problem. This is achieved by smart sampling techniques and only performing VI backups on beliefs states which are relevant. From this point most research has focused on determining efficient strategies to sample belief points and on which to apply VI. Heuristic Search Value Iteration (HSVI1) (Smith and Simmons (2004)) and HSVI2 (Smith and Simmons (2012)) use forward search heuristics to find relevant beliefs by keeping a lower and upper bound on the current estimated value function. The belief tree is expanded by choosing an action and observation with relation to the potential future effect on the value of the bounds, which are being minimised. HSVI has a comparable performance with respect to classical PBVI except in the game of tag (a benchmark problem), in which it fairs significantly better. A method developed after HSVI, named Forward Search Value Iteration (FSVI) (Veloso (2007)) takes an alternative approach to keeping an upper and lower bound on the value function, as in HSVI, since doing so results in a drastic increase in the computation time necessary to find a solution. Instead FSVI assumes that the state space is fully observable and first solves the MDP for this case. The MDP is then used to generate a set of belief points for the PBVI solver. This is achieved by taking the Most Likely State (MLS) and to follow the MDP policy accordingly. It is orders of magnitude faster than HSVI and results in comparable polices. FSVI fairs badly however when information gathering actions are necessary. Since it is essentially using a myopic policy to generate its samples, these will be insufficient to find the global optimal policy when the solution requires information gathering actions. The very last sampling generation technique to date, which is considered to be the most efficient, is SARSOP (Kurniawati et al. (2008)). It uses aspects from both HSVI and FSVI. It keeps upper and lower bounds on the value function and

also uses the MDP solution to generate samples. The key idea of SARSOP is to sample belief points which will contain the optimal set of samples necessary to achieve an optimal policy. Both SARSOP and HSVI2 are considered state of the art in PBVI value approximation techniques. See Du et al. (2010) for a review and comparison of both techniques on problems with thousands of states including simulation examples in grasping, tracking and UAV navigation.

These methods are well suited to addressing problems which are easily expressed in a discrete state space. All considered problems are simulation based and no physical interaction problems are considered. Besides the belief set generation problem, interest has also been poised on porting the PBVI to a continuous state space. An example of a continuous action space PBVI method is Perseus Spaan and Vlassis (2005), in which the authors replace the maximisation over the action by sampling the actions from a parametric continuous representation. In Porta et al. (2006) the state space, transition and observation model are represented by Gaussian Mixtures and the authors consider a particle set or Gaussian mixture representation of the belief. The authors show that a continuous representation of the state space preserves the PWLC property of the value function. They extend their method to continuous action and observations through sampling instead of discretising. Results are shown in a 1D continuous corridor setting. In a more recent approach Brechtel et al. (2013) a discrete state presentation of a continuous state space is learned and is combined with sampling techniques to solve the continuous integrals present in the Bellman equation. The explicit learning of the state representation leads to an increased performance when compared to the other continuous state PBVI methods.

PBVI techniques have come fare since their first application to robotic navigation back in 2003 and have lead to a rapid increase of interest. Initially only a few hundred states could be considered and now problems with over tens of thousand of states are being solved in seconds (very problem specific of course). Most of the research has focused on how to gather a good set of sample beliefs efficiently. Later efforts focused on adapting PBVI to continuous state spaces more suited to robotic applications. The main approach consists of using sampling techniques to overcome the maximisation over the actions (when considering continuous actions) or to choose a suitable parametric representation of the transition, observation and utility model so that the Bellman equation can be solved in closed form. Most evaluations of have focused on simulated and simplified robotic navigation problems in 1D and 2D. We have not discussed online POMDP-solvers since they are also based on VI and sampling techniques and thus share a lot of similarities with PBVI. We refer the reader to Ross et al. (2008) for a detailed review. In summary, trying to preserver the PWLC property of the value functions leads to complicated VI methods which are difficult to port to fully continuous state, action and observation space. Efforts which have attempted to do this have not yet be shown to scale. As a result of this dif-

ficulty, of making this transition to a fully continuous space, approximate value iteration methods have been explored as an alternative. In approximate value iteration the PWLC property of the value function is dropped and is represented by a regression function.

Approximate Value Iteration

Point-based Value Iteration techniques try to preserve the PWLC property of the value function. This directly leads to a discretization of the state space which if continuous by nature, is prone to the curse of dimensionality. An alternative approach is to represent the value function by a non-parametric function, parameterize the belief space and perform approximate dynamic programming.

A very first successful example of this approach is Monte Carlo POMDP (MC-POMDP) Thrun (2000) in which a continuous state, action and observation version of the Heaven & Hell benchmark problem was solved successfully with a working implementation on a non-simulated mobile base. The belief was represented by a particle filter and the policy by a Q-value function, whose functional form was a non-parametric regressor (k-nearest neighbour) of the particle filter. The distance metric was the sample KL divergence between two particle sets. The POMDP was solved through Reinforcement Learning (interaction with the environment) and approximated dynamic programming also known as experience replay, batch RL or Fitted Q-Iteration (FQI) Ernst et al. (2005). Although highly computationally demanding the method was successful.

This inspired many similar approaches such as Brooks and Williams (2011) where the belief state filter was an Extended Kalman Filter (EKF), the value function was also non-parametric and the POMDP was solved via FQI. When compared with Perseus in a discretized 2D localisation task both approaches reached equivalent policies but the authors method achieved it far faster than Perseus, a PBVI method.

An alternative approach is to represented the history of the previous states or observations in an augmented state space and the treat the problem as a standard MDP. In this way the partial observability is directly encoded in the state representation. The motivation is that in contrast to POMDPs there has been fare more research focused on MDPs and much work has been done on the application of non-linear function approximators for representing the value function in combination with reinforcement learning optimisation techniques to solving them. A successful example was the usage of a multi-layer perceptron as a Q-value function approximator, Neural Fitted Q-Iteration (NFQ) Riedmiller (2005). This approach was successfully applied to the standard RL benchmarking problems (carte pole, acrobat, mountain care), but no partially observable setting was considered. Later in Hausknecht and Stone (2015) the authors applied a Deep Recurrent Q-Network (DRQN) (extension to the work in Mnih et al. (2015)) to capture the history of states in a game of Pong where the state

space was occluded half the time. By introducing a long term memory component the POMDP in effect is turned into a MDP and the authors apply an optimisation approach similar to FQI.

The advantage of these approaches is that problem with very large state spaces or continuous state spaces can be solved by using standard machine learning function approximation methods. These methods are easier to understand and implement and adapting POMDP methods to them is relatively straight forward. This is one particular way of dealing with the curse of dimensionality but not the only way. An alternative is to find a latent belief space which is of a much lower dimension than the original and perform value iteration in that space.

Latent Value Iteration

Latent belief space or belief space compression is a way of addressing the curse of dimensionality. The assumption is that although the belief space is of considerable size (thousands of dimensions) a latent belief space exists which is considerable smaller in terms of dimensions (a dozen). A first approach of compressing the belief is to transform it into a set of sufficient statistics (first and second moment for example) and treat the problem as a fully observable MDP in which the states are sufficient statistics of the beliefs. In Roy and Thrun (1999) the authors do just this, they compressed the filtered belief to its mean and entropy and performed VI on this augmented state space in a navigation task in which the goal was to reach a location with a minimum amount of uncertainty. This approach, called Augmented MDP (AMDP), brings a great simplification to solving the POMDP but at the cost of a lossy belief compression.

In further developments Roy (2005) compared both PCA and exponential-PCA (E-PCA) Roy and Gordon (2003), as a means of belief compression technique to find a low dimensional belief space. The authors showed that an original belief of thousands of dimensions could be compressed to a 10 dimensional belief space whilst retaining most of the information. This approach was shown to be superior to AMDP. It requires however computationally expensive transitions back and forth between the low and high dimensional belief states, a necessary step for the the application of VI. The latest work in this area is Li et al. (2010) which investigates the use of non-negative matrix factorisation in combination with k-means clustering as a way of compressing the belief. There method showed some improvement over the E-PCA approach but was only evaluated on discrete benchmark problems.

Belief compression as a means of reducing the curse of dimensionality is an interesting approach. The caveat is that it requires discretising the belief to a fixed grid, collecting many samples and learning an appropriate set of belief-basis eigenvectors. As such, the larger the state space, the larger the dimensionality and thus more samples are required to find a suitable set of basis

belief-eigenvectors. Surprisingly, belief space compression methods have not had wide attention although they shown promising results.

Summary: Value Iteration

Value Iteration seeks to find an optimal policy directly through applying the Bellman equation to a belief-MDP (POMDP) and most of the research has focused on finding ways to alleviated the curse of dimensionality so that VI remains tractable in belief space. The first approach, PBVI, considers a relevant subset of the belief space. Because of the complexity involved in keeping the PWLC property of the value function which restricts its use in large state spaces, alternative approaches discard this property in favour of approximating the value function through machine learning regression techniques. These approaches are considerably more simple to implement than PBVI solvers which require heuristic pruning techniques and are difficult to port to continuous state spaces in general. Alternative approaches have considered finding a latent belief state and perform value iteration in this space. There has however been relatively little work in the latent belief space approach.

Overall, the above approaches consider mostly discrete actions even for the large state (history states) MDPs which have been gaining recent attraction. There are only a few exceptions and these resort to sampling strategies or the usage of paramaterized high level actions. The next approach we consider addresses the problem of continuous actions directly and are termed policy search methods.

2.3.2 Policy search

The approaches seen so far use a value function to encode the problem. When the optimal value function is solved, a policy can be derived from it by taking an action which maximises the value function at each time step, a process known as making the policy greedy with respect to the value function. This requires learning a high dimensional value function of the belief space and the resulting policies are not necessarily smooth, as small changes in the value function can lead to drastic changes in the policy. Even small approximation errors in the value function can lead to very bad greedy policies Baxter and Bartlett (2000). There is no doubt that deriving a policy from a generic value function for highly continuous policy, such as in the case of controlling an articulated robotic arm, is not easy.

This has lead to an alternate approach in which a policy is learned directly without a value function. An initial policy is defined in terms of a paramaterized function, $\pi_{\boldsymbol{\theta}}$, and the utility is a function of the policy parameters, $u(\boldsymbol{\theta})$. The optimal policy is found by searching for the parameters $\boldsymbol{\theta}$ which will maximise

the utility function. This is can be accomplished through various optimisation methods: gradient descent, gradient free, expectation-maximisation, etc...

Gradient: policy search

A very early type of policy search was REINFORCE (likelihood ratio) algorithms first introduced by Williams (1992). From a set of task executions, also called roll-outs, the gradient of the utility function is estimated and used to improve the policy through gradient ascent. The key aspect of this approach is that the derivative of the cost function is independent of the state transition model and as a result the gradient can be estimated by Monte Carlo methods. Application of this methodology to a partially observable setting lead to Gradient POMDP, GPOMDP Baxter and Bartlett (2000) in which the authors developed a conjugate stochastic gradient ascent algorithm to optimise a policy as a function of the current observations. To be optimal, the whole history should be considered or some sort of memory (compressed history) should be introduced. In an extension to this method Aberdeen and Baxter (2002), the authors used a HMM to represent the POMDP which they learned the parameters in conjunction with those of the policy. These are early examples of policy search approaches which are able to fair well on the early POMDP benchmark problems (Heaven & Hell). The main difficulty is to reduce the bias and variance of the gradient estimate which preoccupies most gradient based approaches. Optimising the utility function via stochastic gradient ascent typically needs thousands of gradient estimates such that in expectation terms the parameters are maximising the cost function. An approach which mitigates this problem, coined Pegasus Ng and Jordan (2000), removes the stochasticity from the optimisation by setting the seed of the random number generator constant. A policy evaluation becomes deterministic and by repeating this process many times (different random seeds) the stochasticity is present between the different evaluations and not within them. The end result is the same as stochastic gradient ascent (if repeated sufficient times) but is far easier to optimise individual non-stochastic problems. This policy search method was used to learn a set of controllers for a radio controlled helicopter Kim et al. (2004), which is considered to be one of the very first successful applications of RL to a MDP/POMDP problem. Recent approaches to gradient based methods include grasping objects under Gaussian position uncertainty Stulp et al. (2011), Stulp et al. (2012).

Expectation-Maximisation: policy search

One drawback of gradient based optimisation is that the learning rate plays a significant role on the speed of convergence. An alternative approach consists of using Expectation-Maximisation (EM) methods Kober and Peters (2009) which

do not require a learning rate. Successful applications include: ball-in-a-cup, a humanoid learning the skill of archery Kormushev et al. (2010b), learning how to flip a pancake Kormushev et al. (2010a) and keeping balance on a two-wheeled robot Wang et al. (2016). These are just some examples of the application of RL to continuous action and state space problems. When uncertainty is present, the maximum likelihood state estimate is typically taken and is treated as the true state. A good surveys on policy gradient search methods can be found in: Deisenroth et al. (2011), Kober et al. (2013).

Actor-critic: policy search

Gradient and EM methods only optimise the parameters of the policy, also known as actor only methods. An alternative is to have a separate parameterization of the value and policy functions. This approach is known as an **Actor-Critic**, in which the gradient of the utility function is used both to update the value and policy functions. It has been shown that this approach reduces the variance of the gradient estimate and allows to smoothly change the policy which is desirable when controlling a robot for instance, see Grondman et al. (2012) for a survey highlighting differences and advantages of policy search vs actor-critic methods. A successful application of actor-critic is (episodic) Natural Actor Critic (eNAC) Vijayakumar et al. (2003), Peters and Schaal (2008), a method which uses the *natural gradient* of the value function to update the parameters of a policy. The advantage of using the natural gradient is that it guarantees small changes in the distance between the successive roll-out trajectory distributions. Previous policy gradient methods did not have such guarantees, since small parameter changes of the policy could lead to large changes in the roll-out distributions, which is undesirable. In terms of performance NAC converges faster than GPOMDP and has been applied to learn Dynamic Motor Primitives (DMPs) to control a humanoid robot.

Summary: policy search

For problems in which the state and action space are continuous, policy search is preferred to pure value iteration based methods, which is the case for articulated robotics. In this case, the policies are only guaranteed to be **locally optimal** as oppose to the VI methods which can find **global optimal** policies. However if the parameters of the policy are initialised such that it is in the vicinity of the global optimum of the utility function, then the local optimal will be global. A lot resides on the initialisation and dimensionality of the parameter space of the policy. In terms of using them to solve POMDPs, most examples, at least for robotic applications, act according to the Most Likely State (MLS) or are a function of a history of observations. In such a

way the partial observability is **implicitly** encoded into the policy as opposed to explicitly as was the case for PBVI methods.

Policy gradient methods are iterative and generally require a lot of data to be able to achieve a good policy. Also often the policies learned are not transferable between different tasks and have to be completely relearned. This of course depends on the representation of the state space which if task invariant causes no problem, but unfortunately this is not the case. The next approach to treating uncertainty is more aligned with addressing this last issue of re-usability. These are the **planning** methods.

2.3.3 Planning

Belief space planners leverage the power of traditional planning and optimal control techniques such as: A*, D*, RRT, Dijkstra and LQR to the belief state space. In most of the following techniques (with a few exceptions), a fundamental assumption made is that the motion and measurement models are Gaussian and as a result, a point in the belief space can be represented by the first and second moment: the mean and covariance. An important distinction with VI and Policy search methods is that planners do not solve for a policy. These are online methods in the sense that they have to re-compute a set of actions at every time step as oppose to a policy which can directly query which action should be applied given the current state. The generic objective function used in belief space planning penalises for the amount of uncertainty at the goal and a cost is incurred for every step taken. The planned path will be a compromise between the exploitation actions, which seek to go directly to the goal, and information gathering actions, which seek to reduce the uncertainty.

Belief space road maps

An example of belief space planning is the application of Probabilist Road Maps (PMR) to a belief state space, Prentice and Roy (2009), referred to as Belief Road Maps (BRM). By taking advantage of the linear structure of the Kalman Filter update the authors show that the covariance matrix can be factorised such that a sequence of motion and measurement updates between two belief points in the BRM can be computed by a single linear operation parametrised by the current belief. The key advantage of this approach is that it allows for rapid replanning and is able to scale to large state spaces. The authors evaluated their planner in the MIT campus (simulated). Applications of this methodology include the control of an indoor quadrotor helicopter He et al. (2008) and indoor navigation (a. Agha-mohammadi et al. (2011), a. Agha-mohammadi et al. (2014)) (based on Feedback-based Information Road Maps FIRM , a similar approach in spirit to BRM).

Another main approach is based on optimal control theory, from which Linear Quadratic Controllers (LQG) have been adapted to a belief state space. In this setting the dynamics are considered linear (or linearizable) and the motion and measurement processes are Gaussian. The main difficulty of applying LQG to a belief space is that future observations are unknown, which implies that an expensive marginalisation of the observations has to be done. In Platt et al. (2010) the authors assume instead that at each time step the measurement obtained would be the **maximum-likelihood observation**. This assumption removes the stochasticity from the belief update (since the observations are considered known) and receding horizon optimisation techniques can be applied. These optimisation methods require a nominal trajectory which is generally generated assuming a fully observable state space with standard planning algorithms like RRT Van Den Berg et al. (2011), and subsequently refined by dynamical programming methods until a local optimal solution in attained. In Erez and Smart (2010), the authors parametrized the belief by a mixture of two Gaussians to to tackle unilateral constraints and applied their planner to a 16 dimensional attention allocation problem. The optimisation method used was Differential Dynamic Programming (DDP) and maximum likelihood observations were assumed. For implementations based on this approach, when the planned belief trajectory deviates from the observed belief, replanning takes place. In recent improvements, van den Berg et al. (2012), the assumption of maximum-likelihood observation was removed successfully and has been applied in a simulated surgery problem, Sun and Alterovitz (2014), in which a needle has to be navigated through a body without entering into contact with vital organs.

Most optimal control methods assume that the belief space can be parametrized by a single Gaussian function, which can be restrictive. There have been a few approaches which consider **non-Gaussian** belief state spaces. In Platt et al. (2012) the authors introduce a non-Gaussian belief. The approach initially finds the Most Likely State (MLS) and then samples a set of hypothesis states from the belief. The cost function, with respect to the ML and sampled hypothesis, results in a sequence of actions which will seek to generate measurements which will prove or disprove the hypothetical states with respect to the ML state whilst also trying to reach the goal. Recent work Zito et al. (2013) incorporates this optimisation method into a grasping problem under non-Gaussian pose uncertainty. The method in question is able to perform well with only a few drawn samples from the belief. However the object was not picked up and as a result the stability of the grasp was not evaluated.

Most advances in planning methods in belief space have been in optimal control and were able to show applicability to high-dimensional belief state spaces in a variety of applications. To be fast these methods have to make assumptions with respect to the shape of the belief (Gaussian) and the type of future observations which are available. These can be restrictive but in many applications (such as those which use vision) the uncertainty of objects in the world are often parametrized by Gaussians. The main difference between optimal control approaches and policy search methods is that the computational burden is shifted to online resolution of actions as oppose to constructing a policy offline through repeated interactions with the environment which can be very time consuming. The advantage of planning methods is that they are more flexible than parametric policies in the sense that they are more generic. They solve the objective function online and can be used in different environments, as oppose to a policy which would have to be re-learned.

2.3.4 Heuristics

The methods discussed so far can be considered computationally expensive and/or constraining in the type of belief which can be used (typically a unimodal Gaussian). If the problem domain is more complex or an expensive optimisation problem is not necessarily required, simple heuristic methods can achieve a satisfying solution and in some cases the equivalent of a full blown POMDP solver. Heuristic methods for dealing with uncertainty are widespread in robotics due to the high dimensionality and continuity of the state space. We consider here two heuristic approaches, myopic and information gain. Myopic ignores most of the variance in the uncertainty and considers only the Most Likely State (MLS) whist information gain considers actions in terms of their uncertainty reduction.

Myopic & Q-MDP

Myopic policies consider only the most likely states, which in the case of a Gaussian belief is the mean, and act accordingly. These types of approaches ignore the variance in the uncertainty and risk to fail catastrophically or result in sub-optimal behaviour. MLS is typically used in complicated domains such as grasping, especially when the actual shape of the object is considered to be unknown. A successful approach to this problem is to have a prior non-parametric regressor function representing the shape. As contacts are made with the object more points are added to the regressor improving the shape constructed by exploring the unknown object and gradually acquiring points. The uncertainty of the shape in a region is typically a function of the number

samples. At this point either an exploratory movement is done to move a finger towards a region of high uncertainty (the MLS region) or a grasping attempt is carried out. In Hollinger et al. (2012) an AUV maps the hull of a ship by constructing a mesh and encoding the uncertainty of the mesh with a Gaussian Process (GP). A set of viewing locations, where there is uncertainty (MLS), are computed and a trajectory is obtained by solving a *travelling salesman* problem whilst seeking to maximise coverage of areas with high mesh uncertainty. In Chen and von Wichert (2015) a grasping controller uses the uncertainty, encoded by GP, to guide an exploration process. The fingers would move towards regions of high uncertainty whilst keeping contact with the object. For a good review on related methods for grasping objects under shape uncertainty consult Li et al. (2016), where the authors also use a GP based method to encode the shape uncertainty. The exploration methods for all these methods are in the same in spirit; move towards regions which have high uncertainty (exploration) and when the uncertainty is sufficiently low perform a grasp (exploitation).

An improvement is to consider the variance in the uncertainty and not just the MLS. Such an approach is a called Q-MDP Littman et al. (1995), Nowé et al. (2012) in which the underlying MDP is first solved assuming the state space to be fully observable. Then an action is taken which maximised the expected MDP value function weighted by the belief. This approach only considers uncertainty for one time step but it has been shown to be efficient in some domains (Thrun et al., 2005, Chap. 16). The negative aspect of this approach is that no information gathering actions emerge and the method will fail in problems where this is necessary (Heaven & Hell benchmark problem for instance). Most PBVI based research compare their algorithms against a Q-MDP agent and PBVI always fairs better. For a comparison of different heuristics such as Q-MDP and MLS consult Cassandra et al. (1996b) and for a more recent comparison Lin et al. (2014). A recent application of this method include gaze allocation problems Nunez-Varela et al. (2012) where the uncertainty originates from the limited field of view. In Hauser (2011), Q-MDP is used to evaluate nominal trajectories generated from RRT where starting positions were sampled from the initial belief. A recent follow up on this idea, Vien and Toussaint (2015), considers a task in which a robot has to localise itself with respect to a table. A set of macro actions are evaluated in a Q-MDP framework to achieve this task in which each macro action is solved by an optimal control method.

Both MLS and Q-MDP do not fully consider the uncertainty. This of course leads to great computational gain but at the expense of the quality of the policies, which can be very sub-optimal in some cases. It is known that for increasing the chance of success, a policy which deals with uncertainty needs both **goal orientated** and **information gathering** actions. The next heuristic approach, which we call **information gain**, is based on this concept.

Information gain

Information gain is the decrease or increase of uncertainty resulting from the application of an action. It is obtained by forward simulating the belief and computing the difference between the current entropy and resulting entropy of the simulated action. The vast majority of applications consider a set of marco/parametrised actions. In this set there are typically goal orientation actions which will act as if the state space was fully observable (MLS move) and information gathering actions, whose goal is to reduce the amount of uncertainty such that the goal orientated actions have a higher chance to succeed. The cost function which is optimised is typically a compromise between the distance/time taken to reach the goal and the amount of information gained while executing the task. An early example considered path planning problem for a robot in the National museum of American history Roy et al. (1999). An information gain map was first computed off-line in which a map cell gave an estimate of how much information would be acquired at this location. This was incorporated into an objective function which optimised the information gain along a route with respect to the time taken to reach the goal. The path was given by solving the objective function using dynamic programming. In this case no explicit actions where defined, but the uncertainty was taken into account by weighting informative regions more than open space. The result was trajectories which stayed close to walls. Information gain methods are often used in SLAM applications because of the extremely high dimensionality of the belief space which is of the map and robot position. In Stachniss et al. (2005) a mobile robot is exploring and building a map of an office floor and a set of macro actions are available. A portion of the actions are exploratory and lead the robot to unexplored areas which results in an increase of uncertainty in the overall map whilst the other actions brings the robot back to already explored areas resulting in an improved estimate of the map. For each action the information gain is computed and incorporated in an cost function. A one time step look head is done for each action, which potentially implies an expensive forward simulation, and the action giving the maximum information gain is chosen. This approach has been shown to be effective for large state space problems, notably in Active SLAM navigation Vallve and Andrade-Cetto (2014).

Information gain maximisation is not only restricted to navigation, there are many examples in grasping where this approach is used. Examples include tactile driven exploration such as in Hsiao et al. (2010) where a parametrised set of goal orientated and information gathering actions are used in the context of estimating the pose parameters (6D) of a power drill. The information gain of each action is incorporated into a cost function and the best action is chosen accordingly. The authors report a breadth first search depth of one action to achieve a good performance for the task. Later grasping approaches have built on this with different modifications to the information gain metric Javdani et al. (2012) and there have been successful applications such as finding a door handle

Hebert et al. (2013) and opening a door.

SUMMARY: HEURISTIC

Heuristic methods make strong assumptions which alleviates both the curse of dimensionality and the curse of history associated with POMDP problems. Either the MLS is considered (curse of dimensionality) or the planning horizon is restricted to one time step look ahead (curse of history) as it is the case for Information Gain methods. Heuristic methods in robotics have been regaining traction. In the early days of robotics methods such as Q-MDP, MLS, information gain maps and "best fields of view" were the predominant methods for considering uncertainty in policies and planning algorithms. This was simply due to the computational limitations of the time and POMDP solvers could only handle a few states before the arrival of PBVI methods. Since more sensory information is available and used in robotic systems it is again computationally expensive to compute optimal policies. In many cases spending large computational resources does not result in policies which are obviously superior to simple and intuitive heuristics. Lately many DARPA[2] teams when faces with state uncertainty resort to information gain heuristics, for instance.

2.4 Approach

In the literature we characterised four approaches of how artificial agents have been programmed to reason under uncertainty. The performance of all the methods mentioned in the literature review crucially depend on the quality of **exemplary demonstrations**. For instance, PBVI require search heuristics to find an optimal set of belief points, the quality of the optimal policy of policy search methods depend on the exploration-exploitation trade-off and optimal control methods strongly depend on the initial nominal trajectory. In a way this is intuitive, if you initialise your search method or algorithm with an initial solution which is of high quality (close to optimality) then which ever optimisation method used PBVI, Policy Search, Planning,... a solution should be obtained with computational ease. The question is then: *how to generate such exemplary demonstrations ?*

Programming by Demonstration (PbD) is a methodology whose aim is to achieve the transfer of knowledge and behaviour from a teacher to an apprentice. The teacher is usually a human expert (this is not a constraint) who demonstrates to an apprentice how to accomplish a given task. In the case of articulated robots, kinesthetic teaching is often preferred where the teacher would hold the robot which, is back drivable, and demonstrate to him trajectories. Other ways are possible such as using vision or a wearable interfaces

[2] https://www.youtube.com/watch?v=9Oav3JajR7Q

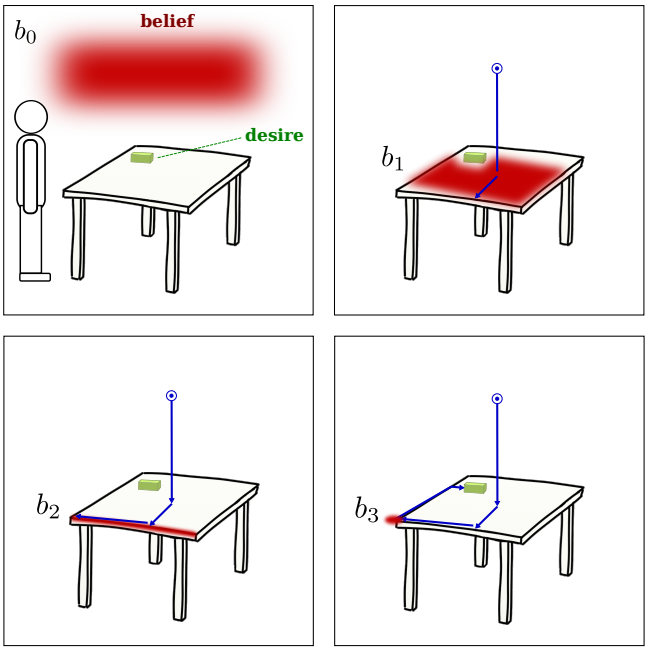


Figure 2.8: Learning from Demonstration in POMDP setting.

which are common to both teacher and expert. We will not go into a great detailed review of PbD, for an in depth review the reader is referred to Billard et al. (2008), Billard and Grollman (2013). However, a particular usage of PbD, which is of interest to use, is learning Dynamical Systems (DS). Learning dynamical systems from programming by demonstration is widely used approach in policy search methods. Most of the lately research in policy search methods reviewed in Section 2.3.2 use some form or another of PbD to initialise the policy. A human first shows a few repetitions of the motion to be accomplished and typically a set of the state-velocity pairs are recorded $\{(x, \dot{x})\}$. Given these examples a stochastic policy $\pi_{\boldsymbol{\theta}}(\dot{x}, x)$ usually represented by a regression function is learned form the recorded demonstrations.

We are considering scenarios which are continuous, sensory information is tactile and haptic and there the level of uncertainty is extremely high. Policy search methods have been shown to be flexible and have been applied to high dimensional robotic problems, such as the full control of a humanoid robot. However, the state space has always been considered fully observable or the MLS was used. Since that there is evidence showing that humans and animals are far better at handling uncertainty than current planners and policies.

References

A. a. Agha-mohammadi, S. Chakravorty, and N. M. Amato. Firm: Feedback controller-based information-state roadmap - a framework for motion planning under uncertainty. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 4284–4291, Sept 2011. doi: 10.1109/IROS.2011.6095010. 2.3.3

A. a. Agha-mohammadi, S. Agarwal, A. Mahadevan, S. Chakravorty, D. Tomkins, J. Denny, and N. M. Amato. Robust online belief space planning in changing environments: Application to physical mobile robots. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 149–156, May 2014. doi: 10.1109/ICRA.2014.6906602. 2.3.3

Douglas Aberdeen and Jonathan Baxter. Scaling internal-state policy-gradient methods for pomdps. In Claude Sammut and Achim Hoffman, editors, *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, pages 3–10, San Francisco, CA, USA, 2002. Morgan Kauffman. ISBN 1-55860-873-7. URL `http://users.rsise.anu.edu.au/~daa/files/papers/gradIstate-icml.pdf`. 2.3.2

Chris Bake, Joshua. Tenenbaum, and Rebecca Saxe. Bayesian theory of mind: Modeling joint belief-desire attribution. *Journal of Cognitive Science*, 2011. 1.2.1

Jonathan Baxter and Peter L. Bartlett. Reinforcement learning in pomdp's via direct gradient ascent. In *In Proc. 17th International Conf. on Machine Learning*, pages 41–48. Morgan Kaufmann, 2000. 2.3.2, 2.3.2

D. Bernoulli. Exposition of a New Theory on the Measurement of Risk (1748). *Econometrica*, 22(1):23–36, 1954. 1.1, 2.1.1

A. Billard and D. Grollman. Robot learning by demonstration. 8(12):3824, 2013. 2.4

A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In B. Siciliano and O. Khatib, editors, *Handbook of Robotics*, pages 1371–1394. Springer, Secaucus, NJ, USA, 2008. 1.1, 2.4

Sebastian Brechtel, Tobias Gindele, and RÃijdiger Dillmann. Solving continuous pomdps: Value iteration with incremental learning of an efficient space representation. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 370–378. JMLR Workshop and Conference Proceedings, May 2013. URL `http://jmlr.org/proceedings/papers/v28/brechtel13.pdf`. 2.3.1

Alex Brooks and Stefan Williams. A monte carlo update for parametric pomdps. In Makoto Kaneko and Yoshihiko Nakamura, editors, *Robotics Research*, volume 66 of *Springer Tracts in Advanced Robotics*, pages 213–223. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-14742-5. doi: 10.1007/978-3-642-14743-2_19. URL http://dx.doi.org/10.1007/978-3-642-14743-2_19. 2.3.1

A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: discrete bayesian models for mobile-robot navigation. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 963–972 vol.2, Nov 1996a. 1.1

A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien. Acting under uncertainty: discrete bayesian models for mobile-robot navigation. In *Intelligent Robots and Systems '96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, volume 2, pages 963–972 vol.2, Nov 1996b. 2.3.4

D. Chen and G. von Wichert. An uncertainty-aware precision grasping process for objects with unknown dimensions. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4312–4317, May 2015. doi: 10.1109/ICRA.2015.7139794. 2.3.4

Guillaume de Chambrier and Aude Billard. Learning search policies from humans in a partially observable context. *Robotics and Biomimetics*, 1 (1):1–16, 2014. ISSN 2197-3768. doi: 10.1186/s40638-014-0008-1. URL http://dx.doi.org/10.1186/s40638-014-0008-1. 1.3

Marc Peter Deisenroth, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. *Foundations and TrendsÂő in Robotics*, 2(1âĂŞ2):1–142, 2011. ISSN 1935-8253. doi: 10.1561/2300000021. URL http://dx.doi.org/10.1561/2300000021. 2.3.2

Y.Z. Du, D. Hsu, H. Kurniawati, W.S. Lee, S.C.W. Ong, and S.W. Png. A pomdp approach to robot motion planning under uncertainty. In *Int. Conf. on Automated Planning and Scheduling, Workshop on Solving Real-World POMDP Problems*, 2010. URL papers/icaps10_pomdpApsInRobotics.pdf. 2.2.1, 2.3.1

Tom Erez and William D. Smart. A scalable method for solving high-dimensional continuous pomdps using local approximation. In *Conf. on Uncertainty in Artificial Intelligence*, 2010. 2.3.3

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556, December 2005. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=1046920.1088690. 2.3.1

I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska. A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):1291–1307, Nov 2012. ISSN 1094-6977. doi: 10.1109/TSMCC.2012.2218595. 2.3.2

Kris Hauser. *Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics*, chapter Randomized Belief-Space Replanning in Partially-Observable Continuous Spaces, pages 193–209. Springer

Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-17452-0. doi: 10.1007/978-3-642-17452-0_12. URL http://dx.doi.org/10.1007/978-3-642-17452-0_12. 2.3.4

Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. 2015. URL https://www.aaai.org/ocs/index.php/FSS/FSS15/paper/view/11673. 2.3.1

Ruijie He, S. Prentice, and N. Roy. Planning in information space for a quadrotor helicopter in a gps-denied environment. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1814–1820, May 2008. doi: 10.1109/ROBOT.2008.4543471. 2.3.3

P. Hebert, T. Howard, N. Hudson, J. Ma, and J.W. Burdick. The next best touch for model-based localization. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 99–106, May 2013. doi: 10.1109/ICRA.2013.6630562. 2.3.4

G. A. Hollinger, B. Englot, F. Hover, U. Mitra, and G. S. Sukhatme. Uncertainty-driven view planning for underwater inspection. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4884–4891, May 2012. doi: 10.1109/ICRA.2012.6224726. 2.3.4

K. Hsiao, L. Kaelbling, and T. Lozano-Perez. Task-driven tactile exploration. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010. doi: 10.15607/RSS.2010.VI.029. 2.3.4

Shervin Javdani, Matthew Klingensmith, Drew Bagnell, Nancy S. Pollard, and Siddhartha S. Srinivasa. Efficient touch based localization through submodularity. *CoRR*, abs/1208.6067, 2012. URL http://arxiv.org/abs/1208.6067. 2.3.4

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1-2):99–134, May 1998. ISSN 0004-3702. doi: 10.1016/S0004-3702(98)00023-X. URL http://dx.doi.org/10.1016/S0004-3702(98)00023-X. 2.2.1

H. J. Kim, Michael I. Jordan, Shankar Sastry, and Andrew Y. Ng. Autonomous helicopter flight via reinforcement learning. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 799–806. MIT Press, 2004. URL http://papers.nips.cc/paper/2455-autonomous-helicopter-flight-via-reinforcement-learning.pdf. 2.3.2

J. Kober and J. Peters. Learning motor primitives for robotics. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 2112–2118, May 2009. doi: 10.1109/ROBOT.2009.5152577. 2.3.2

J. Kober, J. Andrew (Drew) Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, July 2013. 2.3.2

P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with EM-based reinforcement learning. In *Proc. IEEE/RSJ Intl Conf. on Intelligent Robots and Systems (IROS)*, pages 3232–3237, Taipei, Taiwan, October 2010a. 2.3.2

P. Kormushev, S. Calinon, R. Saegusa, and G. Metta. Learning the skill of archery by a humanoid robot icub. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 417–423, Dec 2010b. doi: 10.1109/ICHR.2010.5686841. 2.3.2

Hanna Kurniawati, David Hsu, and Wee Sun Lee. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. In *In Proc. Robotics: Science and Systems*, 2008. 2.3.1

Miao Li, Kaiyu Hang, Danica Kragic, and Aude Billard. Dexterous grasping under shape uncertainty. *Robotics and Autonomous Systems*, 75, Part B:352 – 364, 2016. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2015.09.008. URL http://www.sciencedirect.com/science/article/pii/S0921889015001967. 2.3.4

Xin Li, William K. Cheung, and Jiming Liu. Improving POMDP Tractability via Belief Compression and Clustering. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(1):125–136, February 2010. ISSN 1083-4419. doi: 10.1109/tsmcb.2009.2021573. URL http://dx.doi.org/10.1109/tsmcb.2009.2021573. 2.3.1

Yong Lin, Xingjia Lu, and Fillia Makedon. Approximate planning in pomdps via MDP heuristic. In *2014 International Joint Conference on Neural Networks, IJCNN 2014, Beijing, China, July 6-11, 2014*, pages 1304–1309. IEEE, 2014. doi: 10.1109/IJCNN.2014.6889576. URL http://dx.doi.org/10.1109/IJCNN.2014.6889576. 2.3.4

Michael L. Littman, Anthony R. Cassandra, and Leslie Pack Kaelbling. Learning policies for partially observable environments: Scaling up. In *International Conference on Machine Learning (ICML)*. Morgan Kaufmann, 1995. URL http://people.csail.mit.edu/lpk/papers/ml95.ps. 2.3.4

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 02 2015. URL http://dx.doi.org/10.1038/nature14236. 2.3.1

Andrew Y. Ng and Michael Jordan. Pegasus: A policy search method for large mdps and pomdps. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, UAI'00, pages 406–415, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1-55860-709-9. URL http://dl.acm.org/citation.cfm?id=2073946.2073994. 2.3.2

A. Nowé, P. Vrancx, and Y-M. De Hauwere. *Reinforcement Learning: State-of-the-Art*, chapter Game Theory and Multi-agent Reinforcement Learning, pages 441–470. Springer, 2012. URL http://www.springer.com/engineering/computational+intelligence+and+complexity/book/978-3-642-27644-6. 2.3.4

J. Nunez-Varela, B. Ravindran, and J. L. Wyatt. Where do i look now? gaze allocation during visually guided manipulation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4444–4449, May 2012. doi: 10.1109/ICRA.2012.6225226. 2.3.4

Christos Papadimitriou and John N. Tsitsiklis. The complexity of markov decision processes. *Math. Oper. Res.*, 12(3):441–450, August 1987. ISSN 0364-765X. doi: 10.1287/moor.12.3.441. URL http://dx.doi.org/10.1287/moor.12.3.441. 2.2.1

Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7âĂŞ9): 1180 – 1190, 2008. ISSN 0925-2312. doi: http://dx.doi.org/10.1016/j.neucom.2007.11.026. URL http://www.sciencedirect.com/science/article/pii/S0925231208000532. Progress in Modeling, Theory, and Application of Computational Intelligenc15th European Symposium on Artificial Neural Networks 200715th European Symposium on Artificial Neural Networks 2007. 2.3.2

Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for pomdps. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025 – 1032, August 2003. 2.3.1

R. Platt, L. Kaelbling, T. Lozano-Perez, and R. Tedrake. Non-gaussian belief space planning: Correctness and complexity. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4711–4717, May 2012. doi: 10.1109/ICRA.2012.6225223. 2.3.3

Robert Platt, Russell Tedrake, Leslie Kaelbling, and Tomás Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics Science and Systems Conference (RSS)*, 2010. URL http://groups.csail.mit.edu/robotics-center/public_papers/Platt10.pdf. 2.3.3

Josep M. Porta, Nikos Vlassis, Matthijs T. J. Spaan, and Pascal Poupart. Point-based value iteration for continuous pomdps. *JOURNAL OF MACHINE LEARNING RESEARCH*, 7:2329–2367, 2006. 2.3.1

S. Prentice and N. Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *International Journal of Robotics Research*, 8 (11-12):1448–1465, December 2009. 2.3.3

Kerstin Preuschoff, Peter NC Mohr, and Ming Hsu. Decision making under uncertainty. *Frontiers in Neuroscience*, 7(218), 2013. ISSN 1662-453X. doi: 10.3389/fnins.2013.00218. URL http://www.frontiersin.org/decision_neuroscience/10.3389/fnins.2013.00218/full. 1.1

Akshara Rai, Guillaume De Chambrier, and Aude Billard. Learning from failed demonstrations in unreliable systems. In *Humanoid Robots (Humanoids), 2013 13th IEEE-RAS International Conference on*, pages 410–416. IEEE, 2013. 1.2.3

Edward J. Sondik Richard D. Smallwood. The optimal control of partially observable markov processes over a finite horizon. *Oper. Res.*, 21(5):1071–1088, October 1973. ISSN 0030-364X. doi: 10.1287/opre.21.5.1071. URL http://dx.doi.org/10.1287/opre.21.5.1071. 2.2.1

Martin Riedmiller. Neural fitted q iteration âĂŞ first experiences with a data efficient neural reinforcement learning method. In *In 16th European Conference on Machine Learning*, pages 317–328. Springer, 2005. 2.3.1

Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-draa. Online planning algorithms for pomdps. *J. Artif. Int. Res.*, 32(1):663–704, July 2008. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=1622673.1622690. 2.3.1

N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation-mobile robot navigation with uncertainty in dynamic environments. In *IEEE International Conference on Robotics and Automation*, pages 35–40, 1999. 2.3.4

Nicholas Roy. Finding Approximate POMDP solutions Through Belief Compression. *Journal of Artificial Intelligence Research*, 23, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.35.6180. 2.3.1

Nicholas Roy and Geoffrey J Gordon. Exponential family pca for belief compression in pomdps. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1667–1674. MIT Press, 2003. URL http://papers.nips.cc/paper/2319-exponential-family-pca-for-belief-compression-in-pomdps.pdf. 2.3.1

Nicholas Roy and Sebastian Thrun. Coastal navigation with mobile robots. In *In Advances in Neural Processing Systems 12*, pages 1043–1049, 1999. 2.3.1

Trey Smith and Reid Simmons. Heuristic search value iteration for pomdps. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pages 520–527, Arlington, Virginia, United States, 2004. AUAI Press. 2.3.1

Trey Smith and Reid G. Simmons. Point-based POMDP algorithms: Improved analysis and implementation. *CoRR*, abs/1207.1412, 2012. URL http://arxiv.org/abs/1207.1412. 2.3.1

Matthijs T. J. Spaan and Nikos Vlassis. Planning with continuous actions in partially observable environments. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3469–3474, Barcelona, Spain, 2005. 2.3.1

C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Proc. of Robotics: Science and Systems (RSS)*, Cambridge, MA, USA, 2005. 2.3.4

B.J. Stankiewicz, G.E. Legge, J.S. Mansfield, and E.J. Schlicht. Lost in virtual space: Studies in human and ideal spatial navigation. *Journal of Experimental Psychology: Human Perception and Performance.(under review)*, 32(3):688–704, 2006. 1.1

F. Stulp, E. Theodorou, M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal. Learning motion primitive goals for robust manipulation. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 325–331, Sept 2011. doi: 10.1109/IROS.2011.6094877. 2.3.2

F. Stulp, E. A. Theodorou, and S. Schaal. Reinforcement learning with sequences of motion primitives for robust manipulation. *IEEE Transactions on Robotics*, 28(6):1360–1370, Dec 2012. ISSN 1552-3098. doi: 10.1109/TRO.2012.2210294. 2.3.2

Wen Sun and R. Alterovitz. Motion planning under uncertainty for medical needle steering using optimization in belief space. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 1775–1781, Sept 2014. doi: 10.1109/IROS.2014.6942795. 2.3.3

Sebastian Thrun. Monte carlo POMDPs. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *Advances in Neural Information Processing Systems (NIPS 1999)*, pages 1064–1070. MIT Press, 2000. ISBN 0-262-19450-3. URL http://robots.stanford.edu/papers/thrun.mcpomdp.pdf. 2.3.1

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623. 2.2.1, 2.3.1, 2.3.4

Joan Vallve and Juan Andrade-Cetto. Dense entropy decrease estimation for mobile robot exploration. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 6083–6089, 2014. doi: 10.1109/ICRA.2014.6907755. 2.3.4

Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg. Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *Int. J. Rob. Res.*, 30(7):895–913, June 2011. ISSN 0278-3649. doi: 10.1177/0278364911406562. URL http://dx.doi.org/10.1177/0278364911406562. 2.3.3

Jur van den Berg, Sachin Patil, and Ron Alterovitz. Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research*, 31(11):1263–1278, 2012. doi: 10.1177/0278364912456319. URL http://ijr.sagepub.com/content/31/11/1263.abstract. 2.3.3

Manuela M. Veloso, editor. *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, 2007. 2.3.1

N. A. Vien and M. Toussaint. Pomdp manipulation via trajectory optimization. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 242–249, Sept 2015. 2.3.4

Sethu Vijayakumar, Tomohiro Shibata, and Stefan Schaal. Reinforcement learning for humanoid robotics. In *Autonomous Robot*, page 2002, 2003. 2.3.2

John Von Neumann and O. Morgenstern. *The theory of games and economic behavior*. Princeton, 3 edition, 1990. 1.1, 2.1.1

Jiexin Wang, Eiji Uchibe, and Kenji Doya. Em-based policy hyper parameter exploration: application to standing and balancing of a two-wheeled smartphone robot. *Artificial Life and Robotics*, 21(1):125–131, 2016. doi: 10.1007/s10015-015-0260-7. URL http://dx.doi.org/10.1007/s10015-015-0260-7. 2.3.2

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992. doi: 10.1007/BF00992696. URL http://dx.doi.org/10.1007/BF00992696. 2.3.2

C. Zito, M. S. Kopicki, R. Stolkin, C. Borst, F. Schmidt, M. A. Roa, and J. L. Wyatt. Sequential trajectory re-planning with tactile information gain for dexterous grasping under object-pose uncertainty. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 4013–4020, Nov 2013. doi: 10.1109/IROS.2013.6696930. 2.3.3