Learning Search Strategies from Human Demonstrations

Dissertation (2014)

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. Alireza Karimi, president of the jury
Prof. Aude Billard, thesis advisor
Prof. Hannes Bleuler, examiner
Prof. Jochen Steil, examiner
Prof. Ron Alterovitz, examiner

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 entity. Reasoning given uncertainty is a continuously occurring event throughout our livelihood. When considering long term decisions an abundance of examples come to mind; In economic investments uncertainty is, to the best of efforts, quantified and minimised. 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 which we utilise (or not) to guide us towards the button; Trying to connect a plug to a an occluded power socket under a desk, whilst being crouched, requires the integration of perceptions into a belief such to quantify the uncertainty which we can act upon to achieve the connection. Abilities close to these are not yet present in Artificial intelligence (AI) & robotics.

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, our beliefs and desires, are not directly measurable. The first attempt at modelling the humans decision making process was in mathematics & economics (Bernoulli (1954),Von Neumann and Morgenstern (1990)), where emphasis was on predicting discrete choices formulated as a gamble. It is only recently in Motion and Neuroscience that more incites have been gained.

Artificial intelligence & robotics considered early on uncertainty in decision making, where the predominant application domain was spatial navigation (Cassandra et al. (1996)). The problem is composed into of two parts: the construction and representation of a world model (the map) and a planner which can reason with respect to this model such to accomplish an objective. The world construction problem has attracted a large amount of research with many successfully applications in a wide spectrum of robotic domains (AUV,UAV,etc..). The planning problem is less well developed and is based on either representing the decision problem as a partially observable markov decision process (POMDP) which are 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 and the uncertainty originates from the imprecision of the sensors. As for the planning problem solutions are feasible under the restrictive assumption of a discretization of the world, observations and actions of the robot. As a result there are very few examples where uncertainty is considered in an optimal decision make process when considering a continuous state, action and observation space.

In summary there are still open problems in decision making when considering partial observability, whilst the mapping problem has been studied under a constraining set of assumptions. In this thesis we address both problems under extreme levels of uncertainty. For the decision making side we leverage humans foresight and reasoning in a Learning from Demonstration (LfD) (Billard et al. (2008)) framework, 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, just to name a few. It has been shown, for the moment being, both humans and animals are far better at navigation than robots especially when uncertainty is present (**?**). For the mapping problem we develop a Bayesian filter which is non-parametric and has no explicit representation of a joint distribution.

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.

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. As a consequence the tasks we consider are by nature, haptic and tactile searches.

1.2.1 Learning to reason with uncertainty as humans

A Markov Decision Process (MDP) allows to formulate 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 realises the importance of initial sub-optimal actions which might at first be necessary to achieve the task in the long run. A Partially Observable Markov Decision Process (POMDP), is a generalisation of an MDP to a hidden state space and only observation are available relating to the state space. An exact solution to a

POMDP is only feasible in simple toy problems (Thrun et al. (2005b)) and existing approximate solutions are tailored for discretized representation of states, actions and observations.

In this thesis we propose a Learning from Demonstration approach to solving the POMDP problem 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 behaviour is similar to a Bayesian filter. There is evidence both in cognitive and neuroscience that this is the case (Bake et al. (2011)). From the expert human demonstrations of the task we learn a cognitive model of the humans 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 (position, orientation,...) of an agent 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 usage of SLAM algorithm make the assumption 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 sensor response from the objects (aka landmarks) in question. 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 taken with respect to the shape of the uncertainty (it can be Gaussian, multi-modal, uniform, etc..) and motion noise. From the loose assumptions we stipulate regarding the marginals, 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 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 connection a power plug to a socket, with only haptic and tactile information. We previously addressed this setup by learning a generative model of the beliefs and actions with data provide by human demonstrations following the LfD approach. However, it is usually the requirement in such setups that the teach is an expert, with few notable exceptions (Rai et al. (2013)). Since we were solely learning a statistical controller, bad and good 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 three will have their individual chapter. We first provide and background chapter situating our work in the scientific community and give a conclude with a discussion of the contributions and impact of our work.

In this chapter we review the background literature which are the pillars of this thesis, namely: *Decision Theory*, *Theory of Mind* and *Reasoning under uncertainty*. These three topics are the root nodes of their own respective fields and we do not seek to do all of them justice individually, but highlight their relevance and contribution to our work.

Chapter 2

Background

Planning and reasoning under uncertainty is central to robotic and artificial intelligence research and has been an active area of research for decades. It is an umbrella term which touches a wide spectrum of fields: *economics*, *psychology*, *cognitive science*, *neuroscience*, *robotics* and *artificial intelligence*. The work in this thesis relies on results and assumptions made in cognitive and neuroscience with respect to our beliefs and how we act given them. We complement these results by introducing them in a new light to the field of robotics and demonstrate how the human reasoning and belief system can be used in situations where the state space is partially observable. The second main theme our work builds on is state space estimation. The third component acting given uncertainty in robotics. We make use of results from all three fields. We provide a background overview of acting under uncertainty and situate our work within the state of the art.

This chapter unfolds as follows:

2.1 Decisions under Uncertainty

In this section we introduce and frame the problem we seek to solve in generic terms. We are concerned with finding a sequence of actions which will lead to the successful outcome of a problem being considered; this is the most generic definition.

There are two key attributes which can make this problem difficult: stochastic actions and latent states. Stochastic actions, when applied in the same state will not always result int the same outcome. This type of uncertainty can arise from many sources; the outcome of chaotic actions are impossible to predict with certainty, think of throwing a die or flipping a coin; In outdoor robotics the terrain might lead to slippage, causing the robot to skid or underwater currents might drastically offset the position of an UAV; In articulated robots the friction between joints can accumulate to a large error in the end-effector position (especially true for cable driven robots). The second source of uncertainty is when the underlying state is partially known, in the sense that we do not have all the necessary information to reliably determine the state beyond reasonable doubt. In robotics this uncertainty can arise from inadequate or noisy sensors.

If the environmental conditions in which the robot is located is humid, misty or dark. It can make it difficult for the robot to ascertain its position and to plan how to achieve a given objective.

The uncertainty of the state and actions have to be quantified. The predominant approach is to represent them by probabilities. For instance the application of a forward action (for a wheeled robot) will result in a new position further ahead and a position to the right (due to slippage) with some probability. An observation through the robots sensors will result in probability distribution over the robots probable location. Given this quantification of action and observation uncertainty in terms of a probability distribution over the state, the agent must now take actions towards accomplishing its goal. To take a decision the agent must assign a utility to the outcome of his actions. The utility is to indicate a preference over the outcomes and when combined with probabilities leads to decision theory.

2.1.1 Decision theory

The central question of decision theory is; *how do we take decisions when faced with uncertain outcomes ?* To answer such a question we need to ground 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 in which the 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 a **rational belief**. As for desires they are linked to our disposition to take action to achieve them; for example if I want to switch of my alarm clock I have to look for it in the last area I believed it to be. 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 are the **rational choices** to take given our beliefs to achieve our desires.

Early interest in such questions were typically centred around economics such as deciding what should be an appropriate investment or wager for a particular gamble. It was noted that the expected monitory outcome of a gamble as a mean of basing a decision, would often lead to a course of action which contradicts common sense; a famous example is the St. Petersburg paradox. In this paradox a bookmaker proposes you 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 you leave with the content of the pot. As an avid gambler and expected value max-

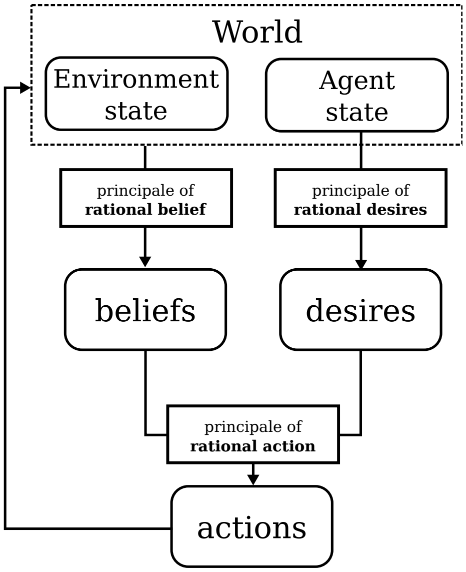

World
Environment state
Agent state
principale of rational belief
principale of rational desires
beliefs
desires
principale of rational action
actions

Figure 2.1: ad

imiser how much would you be willing to pay to enter this gamble ? A value maximiser would computed the expected monetary outcome. The amount of money increases by $2^n£$, where n is the number of non-final tosses and and the probability of reaching n is $1/2^n$. In this case the expected monitory outcome is an infinite number,

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

So your expected gain or return for paying to enter such game is an infinite amount of money, so in principal if you were seeking to maximise your expected return value you would be willing to pay an amount close to infinity. This does not seem a good decision rule; no person in the world would be willing to pay more than 1£ to enter such game.

Nicola Bernoulli proposed a solution to the problem (later published by his brother Daniel (Bernoulli (1954))) by introducing the notion of a **utility function**, and he claimed that people should base their decision on the expected utility instead of solely the monetary outcomes of a gamble.

> "...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 their probability $p(x)$ which results in an expected utility,

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

Different utility functions characterise different levels of risk. When the it is concave as it for Bernoulli's utility function the person will be **risk-averse**, when linear **risk-neural** and convex **risk-seeking**. This was the first introduction of a utility function.

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 who's expectation preserves this preference. An agent whose decisions can be shown to maximise the vNM expected utility are said to be **rational** and otherwise **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 authors knowledge).

An aspect to keep in mind regarding the vNM model is that it is normative; it states what should be a rational decision or behaviour. 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 ours. There have been many studies both demonstrating divergence between the models 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 the way we 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 a decision making process or a control policy. 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).

2.1.2 Beliefs & desires

2.2 Sequential decision process

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

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, $a_t \sim \pi_{\boldsymbol{\theta}}(a_t \vert \cdot)$
$r(x) \in \mathbb{R}$	Reward 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 later utilities/rewards 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).

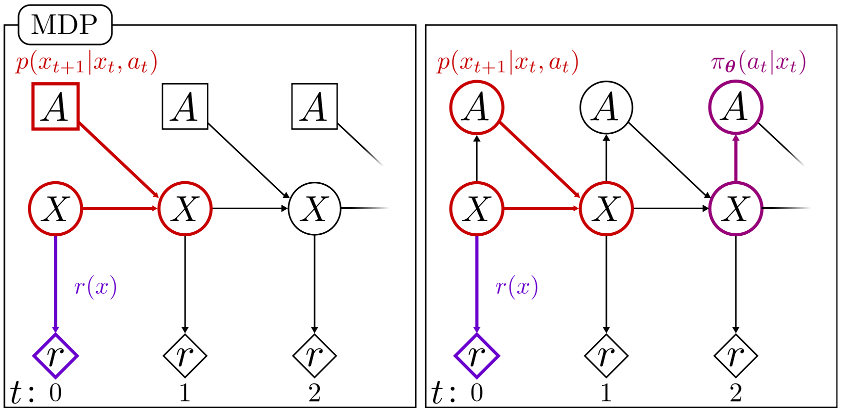
Table 2.1: Definition of common variables used.

When referring outright to decision theory with no extensions, we usually are talking about a one-shot non-temporal decision. However many interesting decision problems are sequential. In such a situation we must consider the effect current decisions will have on future decisions. Expected utility theory (part of decision theory) is extendible to a temporal decision problem. There are however a two subtle but important differences between the temporal and non-temporal decision problems. The first is the utility, in the one time step problem an outcome has one utility assigned to it, $u(x)$. Now 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 outcomes themselves. 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 utility function becomes:

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

The discount factor allows to control the importance later utilities have on the final utility. If the discount factor is set to zero we recover 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 later decisions have on our overall utility. An agent reasoning in such a way is called myopic. The second 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 2.2 we illustrate two graphical representations of a MDP, which are known as **Dynamic Bayesian Networks (DBN)**. A DBN represents the 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 decision process of an agent, given a state it will output an action. A stochastic policy means that given the same input they will produce different outputs. A policy is considered optimal when it maximises the expected utility function, it is optimal in the vNM sense.

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^*,



(a) off-policy

(b) on-policy

Figure 2.2: 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).

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 allows to read off directly the probability of a sequence of state transitions and actions $(x_{0:T}, a_{0:T-1})$, Equation 2.2.

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

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

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

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

Solving the above directly in its current form would have an exponential complexity. By making use of the first order markov assumption and that current rewards do not dependent future rewards, the sums in Equation can be rearranged and and recursive patter will emerge which we can take advantage off. Lets start from the last time step and move out of the sum all elements which do not depend on T and $T-1$, which results in Equation 2.5

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

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.

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

For the terminal state $V_T(x_T) = r(x_T)$. The bellman equation is a means of solving a sequential decision problem through use of dynamic programming. It says the the utility of the current state is based on the immediate reward and the discounted maximum utility of the next state. Making use of this recursion reduced the computation complexity is 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 hard 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 use to the two main approaches to solving a the MDP process: **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 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:

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

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 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 rewards (the utility) and uses them to gradually solve the bellman equation.

We introduced the formulation of a sequential decision process in the form of a MDP model and showed how an optimal policy and value function are obtained through maximising the expected utility. The re-arrangement of the sums via variable elimination allows to take advantage of 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 bored 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 is 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 mathematically define a sensor as being a function of the state space, Equation 2.8.

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

The sensor function $h(\cdot)$ 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. In this setting the state x is a latent hidden variable and we can only get information about it via the observations, y. The uncertainty of the latent state is quantified in a probability distribution over the state space, $p(x)$. This probability distribution represents all the hypothetical positions in the world in which the agent can be. In Figure 2.3 **(a)** an agent is located in a environment a in a yard containing a wall. Initially the agent is confident regarding 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, $p(x_{t+1}|x_t, a_t)$. To reduce the uncertainty, the agent takes a measurement, y, with his sensors which provide range, r, and bearing, ϕ, information of the wall, see Figure 2.3 **(b)**. The agent uses the model of his sensor, known a priori, to deduce all possible locations in the world, x, where the current measurement could have originated from. This model is known as the measurement likelihood function, Equation 2.9

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

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

In Figure 2.3 **(c)** the likelihood is illustrated; the dark regions indicate areas

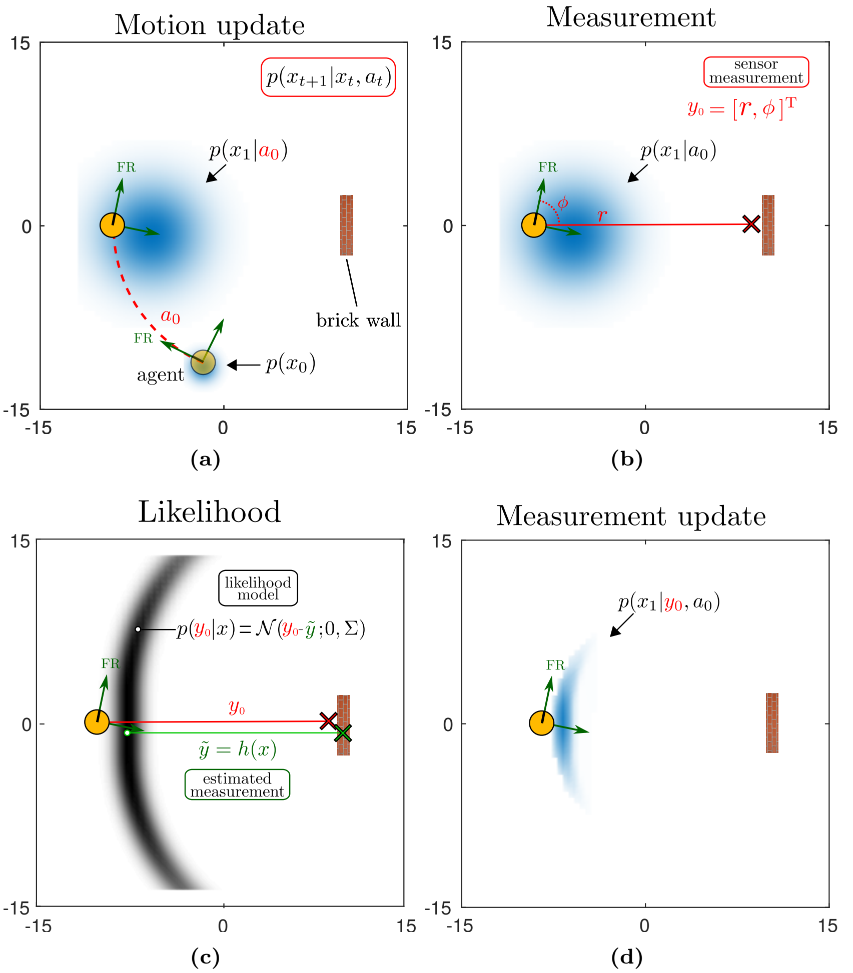


Figure 2.3: **(a)** An agent is located to the south west of a brick wall, it is equipped with a range sensor. The agent takes a forward action, but skids which results in a high increase of the uncertainty.**(b)** The agent takes a measurement, y_0, of this distance to the wall; because his sensor is noisy his estimate is off. **(c)** The agent uses with 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 returns takes the current belief, applied action and measurement to return 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

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

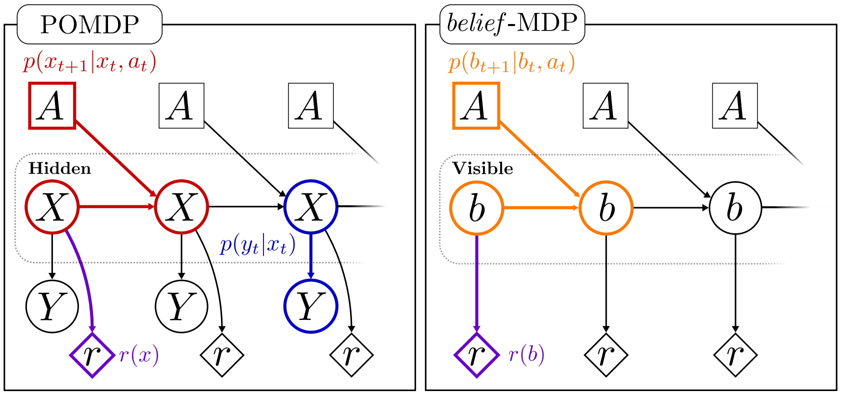
Figure 2.4: Bayesian state space filter.

of high likelihood, which are possible locations from which the sensor measurement could have originated from. The value of the measurement likelihood function is then integrated into the state space probability density function.

These two steps: **motion** and **measurement** updates are part of a 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 the incorporated by Equation 2.11. The measurement likelihood always decreases the uncertainty or leaves it constant. It never results in an increase of uncertainty. 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 as it will become apparent.

With the latent state and its relation to the observation variable and the Bayesian filter defined, we can introduce the POMDP model in Figure 2.5 (*left*). It has the same markov chain structure present in MDP, introduced in the



(a)

(b)

Figure 2.5: **(a)** POMDP graphical model. The state space, X, is hidden, but is still partially observable through a measurement, Y. **(b)** belief-MDP, the POMDP is cast into a belief Markov Decision Process. 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.

previous section, but the state space X is latent and a new layer of observation variables Y is present.

Because the state space is partially observable the expected reward 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 the POMDP to a *belief*-MDP the state space is considered observable and we recover the same structure as in the standard MDP problem.

Because we are working with in a belief-space the reward function has to be adapted to:

$$r(b_t) = \int_{x_t} r(x_t)\, b(x_t)\; dx_t \tag{2.14}$$

which is the expected reward $r(b_t) = \mathbb{E}_{b_t}\{r(x_t)\}$. 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 equation derived previously. We just substitute the new belief transition function and we get the corresponding belief bellman Equation, 2.15.

$$V^*(b_t) = r(b_t) + \gamma \max_{a_t} \int_{b_{t+1}} p(b_{t+1}|b_t, a_t)\, V^*(b_{t+1})\; db_{t+1} \tag{2.15}$$

Using this equation in this form is problematic, we are integrating over the space of beliefs 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. Thus the integration over beliefs vanishes. This can be seen by substituting the belief transition function, Equation 2.16, into the bellman equation Equation 2.15.

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

After the substitution and re-arrangement of the sums we get an integral over all future value functions weighted by there probability, see Equation 2.17 which is the section of the bellman equation after the max. Since the observation is known (because the outer integral is over y_t), the integral over the beliefs vanishes since there is only one possible future belief which is given by the Bayesian filter function $\tau(b_t, a_t, y_t)$.

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

In the final belief state bellman equation, the integral of the belief state is replaced by an integration over observations, Equation 2.18.

$$\begin{aligned} V^*(b_t) &= r(b_t) + \gamma \max_{a_t} \int_{y_t} p(y_t|y_{0:t-1}, a_{0:t})\, V^*(f(b_t, a_t, y_t))\; dy_t \\ &= r(b_t) + \gamma \max_{a_t} \mathbb{E}_{y_t}\{V^*(f(b_t, a_t, y_t))\} \end{aligned} \tag{2.18}$$

The belief bellman equation is intuitive, the value of the current belief is the immediate reward plus the value of the future belief states weighted by the probability of a measurement which would result in these future belief states. It turns out that computing a value function using the above bellman function is not computationally tractable. Solving a POMDP problem as for the MDP case consists of finding the optimal value function from which the optimal policy can be derived. Essentially the same dynamic programming and reinforcement learning techniques can be applied to solve this problem. An exact solution is however only feasible when considering a finite state, action and observation space and a finite planning horizon T. Most early techniques for solving POMDPs used value iteration. It has been shown (Richard D. Smallwood (1973)) that because the reward 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 continuous (PWLC). The intractability comes from the successive applications of the bellman backup operation which result in an exponential time and space complexity with respect to the planning horizon. A good text on the implementation of exact value iteration for POMDPs can be found (Thrun et al., 2005a, Chap. 15) and here Kaelbling et al. (1998).

Why is POMDP so good: combine information gathering and goal directed actions in one. Give taxonomy of planning under uncertainty here.

2.3 State of the art

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 and the fact that it cannot be ignored. If uncertainty is not considered adequately, the control policy risks being suboptimal or lead to drastic failure.

2.3.1 Point-based Value Iteration

The POMDP formulation introduced previously is the main theoretical starting point of policies which consider uncertainty. However solving an exact POMDP, problem through dynamic programming (value iteration) is computationally intractable and an exact solution only exists for discrete state, action and observation space (Thrun et al., 2005a, Chap. 15). This intractability, in which only problems with a few states could be solved, inhibited the application of the POMDP framework to robotics. The first breakthrough, Point-Base Value Iteration (PBVI) Pineau et al. (2003), allowed to apply VI to a robotic navigation problems (626 states). The key insight which allowed VI to scale was to only consider a subset of the belief states which were reachable and relevant to the problem. This is achieved by smart sampling techniques and only perform VI backups on beliefs states which are relevant. Before this point and before only value iteration on this subset instead of considering all beliefs. If you have a 10×10 gird, this results in a total of a 100 states and a belief state of a 100 dimensions! The PBVI approach initiated a renewed effort in porting VI to POMDPs with larger and larger state spaces. From this point most research as focused on determining efficient strategies to sample belief points and which to backup via VI. Heuristic Search Value Iteration (HSVI1) (Smith and Simmons (2004)) and HSVI2 (Smith and Simmons (2012)) uses forward search heuristic to find relevant beliefs by keeping a lower and upper bound on the current estimated value function. The belief tree gets expanded by choosing the action and observation whilst considering there potential future effect on the value of the bounds, which thy try to minimise. It has equivalent results than classical PBVI with an exception to game of tag problem in which it fairs significantly better. Later, Forward Search Value Iteration (FSVI) (Veloso (2007)) takes an alternative approach than keepings an upper and lower bound on the value function, because doing so results in a drastically increases in the computation time necessary to find a solution. Instead they assume that the state space is fully observable, solve the the MDP problem and use this policy in the POMDP setting to generate a set of beliefs. 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 it samples, these will be insufficient to find the global optimal policy when the solution requires information gathering actions. One of the very last sampling generation techniques which is considered to be the most efficient is SARSOP (Kurniawati et al. (2008)), it uses aspects present in both upper and lower bounds on the value function and the equivalent uncertainty free MDP solution to the problem. It tries 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 address problems which are easily expressed in a discrete state space. All considered problems are simulation based and no physical interaction problems are considers. Besides the belief set generation problem, interest has also been poised on porting the PBVI to a continuous state space. A 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 instead sampling them 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 there method to continuous action and observations through a sampling instead of discretising. Results are shown in 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 lead to an increased performance when compared to the other continuous state PBVI methods and the others considered both continuous state and observation space.

PBVI techniques have come fare since their first application to robotics 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 ten thousand states are being solved in seconds. Most of the research has focused on how to gather efficiently a good set of sample beliefs. Later there have been efforts to expand PBVI to continuous state spaces such to be 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 reward model such that the bellman equation can be solved in closed form. Most evaluation of have focused on simulated and simplified robotic navigation problems in 1D and 2D. We have not discussed online POMDP-solvers since there 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.

2.3.2 Parametric Value Iteration

Point-based Value Iteration techniques tried to preserve the PWLC property of the value function. This directly leads to a discretization of the state space which if continuous by nature leads use prone to the curse of dimensionality. An alternative approach is to represent the belief space by a parametric representation, for instance as a Gaussian function. The value function is then defined as a function of the parameters of the Gaussian. This approach in effect greatly diminishes the dimensionality problem but does away with the PWLC property of the value function. Instead to generalise the value of particular parameter a regressor function is needed and to learn it approximate dynamic programming techniques are used.

A very first successful example of this approach is Thrun (2000) were a POMDP problem in a continuous state, action and observation space was solved; with a working implementation on a physical 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 lead to 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 much faster than Perseus, a PBVI method.

Another similar approach to parametrising the belief is to compress it to sufficient statistics and treat the decision problem as a fully observable augmented MDP (AMDP) in this new state representation. In Roy and Thrun (1999) the authors compress 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 brings a great simplification to solving the POMDP by 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 techniques to find a low dimensional belief space. This approach showed to be superior than the AMDP, it however requires transitioning back and forth between the low and high dimensional belief states. A step necessary for the the application of VI. The latest work in this area is Li et al. (2010) which investigates the use of nonnegative matrix factorisation in combination of k-means clustering as a means of compressing the belief which showed some improvement over the E-PCA approach but was only evaluated on discrete benchmark problems. Belief compression as mean of reducing the complexity of the belief state is interesting. The down side is that it requires discretising the belief to a fixed non-dynamic grid, collected many samples and learn 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.

Recent developments lean more to the idea of very large state space representation and treat the problem as a MDP. In contrast to POMDPs there has been fare more research focused on MDPs and a lot of 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 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. It is later in Hausknecht and Stone (2015) that 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.

Parameterization of the belief is a way of reducing the curse of dimensionality. By doing so the VI solution is no longer closed form and function approximators are need to represent the value function. The value function is then optimised via approximate dynamic programming or reinforcement learning. 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. There is a tendency to represent the POMDP has an augmented state MDP which then allows to apply well developed RL techniques. By using advanced non-linear function approximators the large state spaces can be handled in an efficient way and optimal policies can be found. However most of the applications seen so fare do no consider continuous actions at all. Even for standard MDP the application of continuous action is problematic, which is due to need to maximise over the actions. We next review a third approach of dealing with POMDPs which is more adapted to continuous actions.

2.3.3 Policy search

The approaches seen so fare use a vale function to encode the problem which when solved a policy can be derived from it. This requires learning a high dimensional value function over the belief space and the resulting policies are not necessarily smooth. This is because small changes in the value function can lead to drastic changes in the policy [cite]. There is no doubt that deriving a policy from a generic value function for highly continuous policy such in the case of controlling an articulated robotic arm is no easy task. This has lead to development of an alternate approach in which a policy is learned directly without a value function. Instead an initial policy is defined in terms of a parametrised function, $\pi_{\boldsymbol{\theta}}$, and the utility is a function of the 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, expectation-maximisation, etc...

One of the very first applied class of policy search algorithms were the REINFORCE (likelihood ratio) algorithms first introduced by Williams Williams (1992). From a set of example episodes, also called roll-outs, the gradient is estimated. The key aspect to this approach is that the derivative of the cost function is independent of the state transition model and as a result the gradient is easier to estimate from samples. 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 with respect to the average reward. The policy is a function of observations. To be optimal the hole history should be considered or some sort of memory (compressed history) should be introduced. In an extension Aberdeen and Baxter (2002) the authors use a HMM has a state estimator which they learn the parameters in conjunction with those of the policy which depends on the state estimate of the HMM. 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 which preoccupies most gradient based approaches is the bias and variance of the gradient. As a result of optimising the stochastic problem via stochastic gradient ascent, typically thousands of gradient estimates are necessary 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 fixing the random number generator. 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 would be the same as stochastic gradient ascent (if repeated sufficient times) but is fare more 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. It is not until Natural-Actor Critic (NAC) (Vijayakumar et al. (2003), Peters and Schaal (2008)), a policy gradient method which uses the *natural gradient* to update the parameters of a policy. The advantage 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. A drawback of gradient based optimisation is that the learning rate plays a significant important on the time to convergences. 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, 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 typically the maximum likelihood state estimate is taken.

Good survey on policy gradient search methods can be found here, Deisenroth et al. (2011), Kober et al. (2013).

PI2 stuff just mention it and wrap it up: Stulp et al. (2012) Stulp et al. (2011b)

2.3.4 Heuristics

Hebert et al. (2013) Roy et al. (1999) Vallve and Andrade-Cetto (2014) Zhang et al. (2015)

2.3.5 Belief space planning

Belief space planning leverage's the power of traditional fully observable state space planning and optimal control techniques such as: A*, D*, Dijkstra and LQR to the belief belief state space. The fundamental assumption made in most of the following techniques (with a few exceptions) is that the motion and measurement modes are Gaussian and as a result a point in the belief space can be represented by a mean and covariance function. An example 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 the the authors show that the covariance matrix can be factorised such that a set 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. They 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 [another one].

Another approach ports the methods of optimal control theory, namely Linear Quadratic Control (LQG) to belief space planning. In this setting the dynamics are considered linear and the motion and measurement processes are Gaussian. The main difficulty of applying LQG planning to a belief space is that future observations are unknown, which implies that an expensive marginalisation over the observations would have to be carried out. In Platt et al. (2010) the authors assume instead that at each time step the measurement obtained is the *maximum-likelihood observation*. This assumption removes the stochastic from the belief update (since the observation is considered known) and a multiple shooting optimisation methods can be employed. To apply this method a nominal trajectory is first created, assuming that the state space is fully observable, and subsequently refined by dynamical programming methods until a local optimal solution in attained. 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 the cost function is quadratic and the authors imploy iterative LQG.

In Erez and Smart (2010), the authors consider a 16 dimensional continuous state, action, etc. 6 continuous actions. Use Differential Dynamic Programming (DDP) and maximum likelihood of observations. Use a mixture of two Gaussians to represent the state space to tackle unilateral constraints. DDP gives a sequence of linear feedback gain matrices.

In Martinez-Cantin et al. (2009b), finite horizon planning problem, belief dependent utility function. non-Gaussian, trades off exploration vs exploitation. The pomdp policy is represented by a parameterised path. Use PID controller to follow the planned path. Dynamic model is non-differentiable, cannot use gradient based optimisation methods. Use bayesian technique to approximate the cost function with a surrogate function that is cheaper to evaluate. sample parameters wit expected cost, learn a function relating parameter choice to cost. The parameterised policy trades off exploration vs exploitation. Aim to minimise the number of cost function evaluations. Minimise uncertainty about its pose (location and heading). Policy is parameterised by a finite set of way points. Same cost function as RL, but cost is with respect to the map (doing SLAM). Cannot use LQG to solve this problem

Platt et al. (2012) Non-gaussian belief and non-linear optimisation problems. The computational complexity of the algorithm is dominated by the number of samples used. Receding horizon control approach. Given belief, first find most likely state. Generate a set of plausible locations (very likely). Choose u such that set of observations generated by paths from state point x are as different as possible. Confirm or disprove as many of the hyopthesis states or not.

a. Agha-mohammadi et al. (2014)

Lee et al. (2013) Kaelbling and Lozano-Pérez (2013)

2.3.6 Optimal control

$b = (\mu, \Sigma)$

Grasping Chen and von Wichert (2015) Stulp et al. (2011a) Zito et al. (2013) Li et al. (2016)

Hollinger et al. (2012) Martinez-Cantin et al. (2009a) He et al. (2011)

2.4 Summary

References

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. 2.3.5

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. 2.3.3

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. 2.3.3

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, 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

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. 2.3.2

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 1996. 1.1

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. 2.3.6

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. 2.3.3

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. 2.3.1

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. 2.3.5

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. 2.3.2

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. 2.3.2

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. 2.3.5

Ruijie He, Emma Brunskill, and Nicholas Roy. Efficient planning under uncertainty with macro-actions. *J. Artif. Int. Res.*, 40(1):523–570, January 2011. ISSN 1076-9757. URL http://dl.acm.org/citation.cfm?id=2016945.2016959. 2.3.6

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. 2.3.6

Vadim Indelman, Luca Carlone, and Frank Dellaert. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, 34(7):849–882, 2015. doi: 10.1177/0278364914561102. URL http://ijr.sagepub.com/content/34/7/849.abstract. 2.3.5

Leslie Pack Kaelbling and Tomás Lozano-Pérez. Integrated task and motion planning in belief space. *Int. J. Rob. Res.*, 32(9-10):1194–1227, August 2013. ISSN 0278-3649. doi: 10.1177/0278364913484072. URL http://dx.doi.org/10.1177/0278364913484072. 2.3.5

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. 2.3.3

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. 2.3.3

J. Kober, J. Andrew (Drew) Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, July 2013. 2.3.3

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. 2.3.3

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. 2.3.3

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

A. Lee, Y. Duan, S. Patil, J. Schulman, Z. McCarthy, J. van den Berg, K. Goldberg, and P. Abbeel. Sigma hulls for gaussian belief space planning for imprecise articulated robots amid obstacles. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 5660–5667, Nov 2013. doi: 10.1109/IROS.2013.6697176. 2.3.5

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. 2.3.6

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. 2.3.2

Ruben Martinez-Cantin, Nando de Freitas, Eric Brochu, JosÃľ Castellanos, and Arnaud Doucet. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2):93–103, 2009a. ISSN 0929-5593. doi: 10.1007/s10514-009-9130-2. URL http://dx.doi.org/10.1007/s10514-009-9130-2. 2.3.6

Ruben Martinez-Cantin, Nando Freitas, Eric Brochu, José Castellanos, and Arnaud Doucet. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots*, 27(2):93–103, 2009b. ISSN 1573-7527. doi: 10.1007/s10514-009-9130-2. URL http://dx.doi.org/10.1007/s10514-009-9130-2. 2.3.5

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. 2.3.2

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. 2.3.3

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. 2.3.3

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. 2.3.5

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. 2.3.5

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. 2.3.5

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. 2.3.2

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. 2.3.2

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. 2.3.2

Nicholas Roy and Sebastian Thrun. Coastal navigation with mobile robots. In *In Advances in Neural Processing Systems 12*, pages 1043–1049, 1999. 2.3.2

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

F. Stulp, E. Theodorou, J. Buchli, and S. Schaal. Learning to grasp under uncertainty. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5703–5708, May 2011a. doi: 10.1109/ICRA.2011.5979644. 2.3.6

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 2011b. doi: 10.1109/IROS.2011.6094877. 2.3.3

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. 2.3.3

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. 2.3.2

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

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. Probabilistic robotics (intelligent robotics and autonomous agents). 2005b. 1.2.1

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, 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. 2.3.5

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

Sethu Vijayakumar, Tomohiro Shibata, and Stefan Schaal. Reinforcement learning for humanoid robotics. In *Autonomous Robot*, page 2002, 2003. 2.3.3

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. 2.3.3

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. 2.3.3

Q. Zhang, I. Rekleitis, and G. Dudek. Uncertainty reduction via heuristic search planning on hybrid metric/topological map. In *Computer and Robot Vision (CRV), 2015 12th Conference on*, pages 222–229, June 2015. doi: 10.1109/CRV.2015.36. 2.3.4

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. 2.3.6