Application of POMDPs to Cognitive Radar

Charles Topliff, William Melvin, Douglas Williams
Center for Signal & Information Processing
School of Electrical & Computer Engineering
Georgia Insitute of Technology, Atlanta, Georgia
ctopliff0@gatech.edu, bill.melvin@gtri.gatech.edu, doug.williams@coe.gatech.edu

***Abstract*— In recent years, hardware advances have resulted in software configurable radar systems that lend themselves well to decision-making systems. Partially observable Markov decision processes (POMDPs) are evaluated herein as a framework for decision-making in radar scenarios, and value iteration is examined as a method for computing an optimal decision policy with a POMDP. A scenario is investigated wherein a radar is competing with a greedy agent for spectrum. Results demonstrate improvement over a heuristic decision-making agent that seeks to maximize immediate reward.**

I. INTRODUCTION

Improvements in radio frequency hardware and embedded computing enable a new class of wideband, multichannel, configurable radar systems with the potential to adapt to vastly changing characteristics in the operating environment. These systems are a marked change from years past, where radar systems were custom-designed to operate within stringent constraints, such as a narrow frequency allocation, and operating mode (e.g., airport terminal wind shear monitoring). Contemporary radar designs include all-digital arrays with arbitrary waveform generation at the element level, 10:1 operating frequency range, and highly flexible back-end processing. The ability to automatically and dynamically manipulate the configuration settings of software-defined radars opens up the possibility for enhanced performance in environments where spectrum access is competitive, dynamic, and conflicted by multi-user operating needs.

A cognitive radar interacts with the environment, senses the corresponding response, and then attempts to optimize resource allocations to achieve desired objectives, such as access to contiguous spectrum bands over a sufficient time duration to generate a high quality, high range resolution target profile. In [1], cognitive radar is described from the perspective of the perception-action cycle, as shown in Figure 1. An alternative,

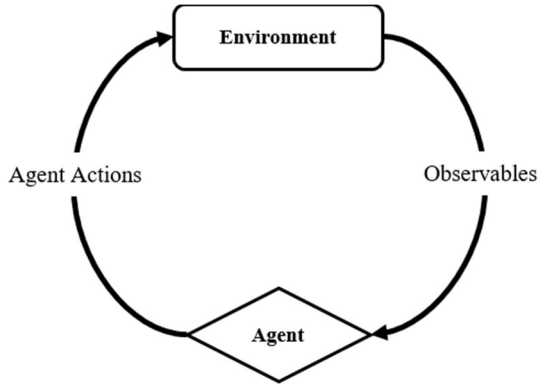

Figure 1: The Perception-Action Cycle

Identify applicable funding agency here. If none, delete this text box.

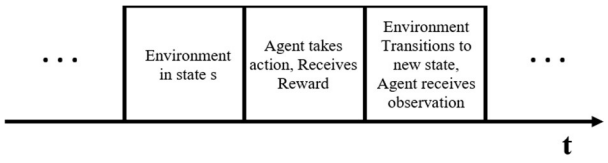

Figure 2: POMDP perception-action cycle.

but related, view given in [2] is based on the Rasmussen model commonly used in robotics and human factors engineering, which asserts the perception-action cycle on three levels: the skill-based layer, the rule-based layer, and the knowledge-based layer. The Rasmussen model suggests an implementation strategy based on the use of a partially observable Markov decision processes (POMDP). A number of papers on cognitive radar focus on radar enhancement techniques leveraging prior knowledge through parametric, model-based strategies [3-5], or machine learning algorithms for perception functions [6-7], thus differing substantially from the POMDP approach involving implementation of the perception-action cycle.

A POMDP includes agent states, observations, a transition model, and costs/rewards for various actions. A policy is a course of action for various observables and exists for a specified horizon. In this paper, we investigate the application of cognitive radar using the POMDP approach for the case of a radar competing for contiguous spectrum slots with another user who wishes to maximize its own access to spectrum.

The rest of this paper is organized as follows. Section II gives an overview of POMDPs and value iteration algorithms for computing optimal policies. In Section III, we describe the POMDP model used for our experiments. We present simulation results and discussion in Section IV.

II. POMDPS & VALUE ITERATION

A. The POMDP Model

A POMDP is an extension of a Markov Decision Process (MDP) where uncertainty in environment state is embedded into the model. In an MDP, although there may exist uncertainty in the effects of an agent's actions, the agent always is completely aware of the environment state [8]. Given the uncertainty inherent to the radar environment, POMDPs are the more desirable model for radar scenarios. A POMDP is defined by the tuple $\langle S, \Omega, A, T, O, R, \gamma \rangle$, where S is a discrete set of states, Ω is a discrete set of observations, A is a discrete set of actions, T is a state transition matrix for the states in S, O is an observation probability matrix for the observations in Ω, R is a reward matrix, and γ is a reward discount factor between zero and one [9]. One example of a POMDP is the two-state tiger problem, where an agent is faced with two doors, one of which has a tiger behind it [10]. The actions are opening door A,

opening door B, or listening for the tiger. The observations are that the tiger is behind door A or door B. Clearly, the agent would prefer to open the door that does not have the tiger behind it.

In a POMDP, at a given time step, an agent lies in a state $s \in S$, and takes an action $a \in A$. The agent then immediately receives a reward $R(s,a)$, transitions to a new state $s' \in S$, and receives an observation $o \in \Omega$. Figure 2 demonstrates the relationship of the POMDP dynamics to the perception-action cycle. The agent maintains a belief state b for every time step, whereupon receiving a new observation after taking action a and transitioning to s', the new belief state b' is given by

$$b'(s') = \frac{O(o,s',a)\sum_{s\in S} T(s, s', a)b(s)}{\sum_{s'\in S} O(o, s', a)\sum_{s\in S} T(s, s', a)b(s)} \quad (1)$$

where *b'(s)* denotes the probability of being in state *s* according to the agent belief state, *O(o, s', a)* is the probability of receiving observation *o* after taking action *a* and transitioning to state *s'*, and *T(s, s', a)* is the probability of transitioning to state *s'* after taking action *a* in state *s*. The goal of the decision making agent is to maximize the total discounted reward (we have some discount factor γ as part of a POMDP which effects how influential future rewards are at in the value function) over a horizon of *K* time steps:

$$R_{tot} = \sum_{i=1}^{K} \gamma R(s_i, a_i) \quad (2)$$

We seek to compute an optimal policy π^* that the continuous belief space can be mapped into to maximize the expected sum of discounted rewards. In the tiger POMDP, the optimal policy is to take the listen action until the belief state enters some region of the space which corresponds to the tiger lying behind one of the doors, and then open the opposite door.

B. *Value Iteration for computing optimal policies*

There exist a number of methods for computing the optimal policy for a POMDP [11]. One commonly used method for computing optimal policies is value iteration. In this work, the incremental pruning algorithm is used [12-13]. Here, an overview of value iteration and the incremental pruning algorithm are given.

A value function is a mapping from a belief state b' to expected discounted reward. A value function V' includes an additional step of reward from the previous value function. V' is given as

$$V'(b) = \max_{a\in A} \left(r_a^T b + \gamma \sum_{o\in O} O(o, s', a)V(b)\right) \quad (3)$$

The authors of [13] decompose (3) into the following equations:

$$V'(b) = \max_{a\in A} V^a(b) \quad (4)$$

$$V^a(b) = \sum_{o\in O} V_o^a(b) \quad (5)$$

$$V_o^a(b) = \frac{\sum_s r_a(s)b(s)}{|O|} + \gamma O(o, s', a)V(b_o^a) \quad (6)$$

Where $|\cdot|$ is the cardinality operator. In [14], Smallwood &

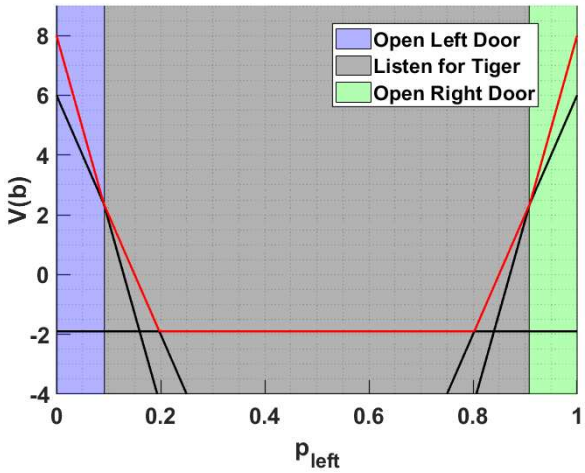

Figure 3: Tiger POMDP Value Function. $\mathbf{p_{left}}$ represents the probability that the tiger is behind the left door according to the agent belief state. The shaded regions represent the corresponding optimal action to the regions in belief space The black lines represent the portion of each line that does not lie at the maximum of the value function. The red lines represent the maximum of the value function.

Sondik prove that the value function $V(b)$ is piecewise linear and convex and can, thus, be written as

$$V(b) = \max_{\lambda\in\Lambda} b^T\lambda \quad (7)$$

for λ in a finite set of $|S|$-vectors Λ. [13] proceeds by rewriting (4)-(6) as $V'(b) = max_{\lambda\in\Lambda'}\lambda^T b$,$V^a(b) = max_{\lambda\in\Lambda^a}\lambda^T b$, and $V_o^a(b) = max_{\lambda\in\Lambda_o^a}\lambda^T b$ for some finite sets of $|S|$-vectors Λ', Λ^a, and Λ_o^a. These sets of vectors are given by

$$\Lambda' = purge\left(\bigcup_{a\in A}\Lambda^a\right) \quad (8)$$

$$\Lambda^a = purge\left(\bigoplus_{o\in O}\Lambda_o^a\right) \quad (9)$$

$$\Lambda_o^a = purge(\{\tau(\lambda, a, o)|\lambda \in \Lambda\}) \quad (10)$$

Where $purge(\cdot)$ is a function defined below that reduces a set of vectors to its minimum size representation, $\tau(\lambda, a, o)$ is the vector given by $\tau(\lambda, a, o)(s) = \frac{r_a(s)}{|O|} + \gamma\sum_{s'}\lambda(s')O(o, s', a)T(s', s, a)$, and the cross sum of two sets of vectors $A \oplus B$ is given by $A \oplus B = \{\alpha + \beta | \alpha \in A, \beta \in B\}\}$.

For a set of vectors A and an additional vector λ, the witness region is the set of information states for which vector λ has the largest dot product compared to other vectors in A, given by:

$$R(\lambda, A) = \{b|b \geq 0, b^T\mathbf{1} = 1, b^T\lambda > b^T\lambda' \; \forall \; \lambda' \in A\} \quad (11)$$

We can define the $purge$ function using (11):

$$purge(A) = \{\lambda|\lambda \in A, R(\lambda, A) \neq 0\} \quad (12)$$

This function takes as input a set of vectors A and returns the vectors in A with non-empty witness regions. The implementation in [13] (FILTER) uses a linear programming approach for finding points in belief space, where a single vector is dominant over all others in the set (i.e., this vector has the

maximal dot product with the points in this region as compared to the rest of vectors in the set). We refer the reader to [13] and [15] for the more technical details of this algorithm.

The incremental pruning algorithm relies on an efficient implementation of (7). We note that $purge(A \oplus B \oplus C) = purge(purge(A \oplus B) \oplus C)$. We can thus write (7) as

$$\Lambda^a = purge\left(\ldots purge\left(purge\left(\Lambda^a_{o_1} \oplus \Lambda^a_{o_2}\right) \oplus \Lambda^a_{o_3}\right) \ldots \oplus \Lambda^a_{o_k}\right) \tag{13}$$

The incremental pruning algorithm proceeds by initializing with an empty set W, populates W with vectors in the cross sum of $\Lambda^a_{o_1}$ and $\Lambda^a_{o_2}$ that have a non-empty witness region, and then iteratively applies the aforementioned FILTER algorithm to the cross-sum of W and $\Lambda^a_{o_k}$. We can then purge Λ^a and carry actions along to obtain the set of vectors that represent Λ'. An example of the value function for the tiger POMDP mentioned in Section 2 can be seen in Figure 3. Note that all of the lines in the lie at the maximum of the value function at some point in the belief space. The red lines indicate the maximum of the value function at any point in the belief space.

Complexity analysis of the incremental pruning algorithm is discussed in depth in [13]. We remark that the advantage of implementing the algorithm as in (13) comes from the fact that if $A = purge(A)$, $B = purge(B)$, and $W = purge(A \oplus B)$, then $|W| \geq \max(|A|, |B|)$. Thus, the size of W in the incremental pruning algorithm is monotonically non-decreasing. Cassandra, et al. [13] also discuss more general implementations of incremental pruning. For our work, we apply the FILTER algorithm proposed in [15] for purging sets of vectors.

III. Greedy Radar Scenario as a POMDP

In this section, we seek to model the scenario where a radar is competing with a benign agent for two spectrum slots as a POMDP. The benign agent is operating in some configuration of the available spectrum slots at each given time step. The goal of the radar is to transmit without colliding with the other user of the spectrum. The radar also has some ability to influence the behavior of the benign agent. If the radar transmits and collides with the benign agent, the benign agent may move to an unoccupied spectrum slot or stop transmitting completely. The radar can also spoof in an attempt to force the benign agent out of its current configuration, with a greater success chance than a normal transmission. In some circumstances spoofing can be more desirable, because the radar is more certain to influence the behavior of the benign agent.

To capture this scenario in a POMDP, we must define the tuple as in Section II. This scenario has a total of four states, where the benign agent is not present in the spectrum (both slots free), the benign agent occupies only slot one, the benign agent occupies only slot two, or the benign agent occupies both spectrum slots. For the set of actions, the radar can transmit in any of the non-empty slot configurations or spoof in any of these configurations. We also include the action of 'sensing,' where the radar senses the environment but occupies no spectrum slots. These options give a total of seven actions. The set of observations is simply the set of states. The radar has some chance of observing the new state of the environment given the new state s' and action a.

For the reward matrix, if the radar takes a transmit action and the environment state is such to avoid collision with the benign agent, the radar is rewarded. If the environment state is such that the radar transmits in a slot occupied by the benign agent, the radar is assessed a penalty. The radar is penalized for taking the sensing action and penalized (albeit less harshly) for taking a spoof action.

For the observation probability matrix, we define a transmit certainty parameter c_{tx}, a spoof certainty parameter c_{sp}, and a sensing certainty parameter c_{sns}, which are the probabilities of correctly observing the state given the radar transmits, spoofs, or senses, respectively (typical values used are c_{tx}~$[.4, .6]$, c_{sp}~$[.7, .9]$, c_{sns}~$[.95, .99]$. The sets $A_{tx} \subset A$, $A_{sp} \subset A$, and $A_{sns} \subset A$ are denoted as the sets of actions corresponding to transmit, spoof, and sensing actions, respectively. The quantity q(a) is defined as

$$q(a) = c_{tx}\mathbb{1}_{A_{tx}}(a) + c_{sp}\mathbb{1}_{A_{sp}}(a) + c_{sns}\mathbb{1}_{A_{sns}}(a) \tag{14}$$

Where $\mathbb{1}_A(a)$ is the indicator function, defined as

$$\mathbb{1}_A(a) = \begin{cases} 1, & if\ a \in A \\ 0, & otherwise \end{cases} \tag{15}$$

$O(o, s', a)$ is then given by

$$O(o, s', a) = \begin{cases} q(a) & if\ o = s' \\ \dfrac{1 - q(a)}{|S| - 1} & otherwise \end{cases} \tag{16}$$

For the transition probability matrix, similar parameters are defined as before for the amount of influence that an action has over the benign agent. The terms m_{tx}, m_{sp} are the transmit and spoof influence parameters, which denote the total probability that the benign agent will move to a state not including any of the slots occupied by the radar (typical values are m_{tx}~$[.4, .6]$, m_{sp}~$[.8, .9]$). Define S^a_{rad} as the set of states that have a frequency slot in common with the radar's current configuration when the radar takes action a. The quantity $p(a)$ is defined as

$$p(a) = m_{tx}\mathbb{1}_{A_{tx}}(a) + m_{sp}\mathbb{1}_{A_{sp}}(a) \tag{17}$$

For the sensing action (a = 1), take the state transition matrix to be the identity matrix (i.e., the benign agent maintains its state configuration if the radar chooses to stay silent for a time step). Otherwise, $T(s, s', a)$ is given by:

$$T(s, s', a) = \begin{cases} \dfrac{p(a)}{|S| - |S^a_{rad}|} & if\ s' \notin S^a_{rad} \\ \dfrac{1 - p(a)}{|S^a_{rad}|} & otherwise \end{cases} \tag{18}$$

We now remark on a disadvantage we have encountered with the use of POMDPs. POMDPs present a very rigid model, and significant deliberation is necessary to completely capture a dynamic scenario as a POMDP. Such a rigid model may be especially problematic if the radar environment is nonstationary. In this work, the radar is assumed to have access to a POMDP model through some form of learning or observation. A simple

radar scenario is considered with only two spectrum slots. Moving to more than two slots would require significantly more state enumeration. The incremental pruning algorithm would also grow significantly in computational complexity, as the main component in the algorithm involves the solving of linear programs that become large as the number of states and actions grows.

IV. Simulation Results

In this section, simulation results are detailed for the POMDP approach to decision making. The Incremental Pruning algorithm was implemented in MATLAB, and all linear programming was implemented using the optimization toolbox's *linprog* function. We profiled the code for our experiments, and over 90% of the runtime was spent solving linear programs.

Naturally, a more efficient linear program solver would have a faster runtime. Also, a number of other algorithms for approximate value iteration exist that show interesting results and characteristics. [11, 16].

For our experiments, we compare the performance of the decision-making agent to a heuristic agent that simply selects the most rewarding action based on the most recent observation. For example, if the agent receives the observation that the benign agent is occupying no portion of the spectrum, the agent would transmit in both slots for its next action. We compute the optimal policy for the radar for a horizon of five time steps. We simulate the scenario for a total of 25 time steps. The increased number of states of this environment prevents visualization of the value function surface as in Figure 3. Instead, a time-frequency diagram is presented for both the intelligent agent and the heuristic agent. Figure 4 demonstrates a comparison of the time-frequency diagrams of the intelligent agent and the heuristic agent. In this example, observation matrix parameters $c_{tx} = 0.5$, $c_{sp} = 0.9$, $c_{sns} = 0.99$ and transition influence parameters $m_{tx} = 0.5$, $m_{sp} = 0.9$ were chosen. Note that, while both agents are subject to the same initial conditions and take the same first action, the total state progression diverges over time. Fewer spectrum collisions are desirable, and the intelligent agent can be seen to yield substantially better results. In this example, the intelligent agent generated a collision

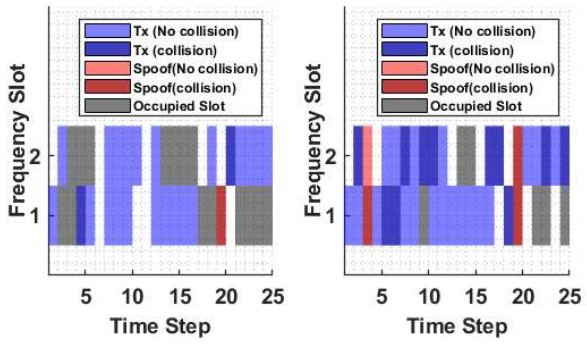

Figure 4: (a) Time-frequency diagram for the intelligent agent over 25 time steps. (b) Time-frequency diagram for the heuristic agent over 25 time steps.

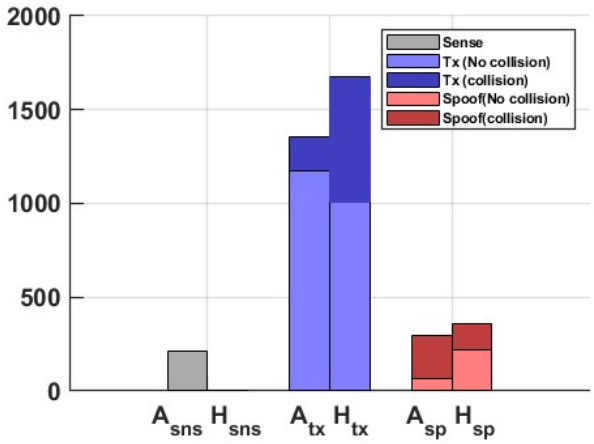

Figure 5: Bar chart demonstrating overall behavior of the two decision-making agents over 2000 time steps. $\mathbf{A_{sns}}$, $\mathbf{A_{tx}}$, $\mathbf{A_{sp}}$ correspond to the intelligent agent sensing, transmitting, and spoofing, respectively while $\mathbf{H_{sns}}$, $\mathbf{H_{tx}}$, $\mathbf{H_{sp}}$ correspond to the heuristic agent sensing, and spoofing, respectively. The vertical axis indicates the total number of times an action was taken.

penalty of 2, while the heuristic agent generated a collision penalty of 11. The intelligent agent also has a significantly higher reward score according to the reward matrix as a result of the great reduction in the number of collisions. A longer simulation was run over 2000 time steps to get a sense of the overall behavior of the two agents. Figure 5 characterizes the overall behavior of the two agents. Note that the heuristic agent does not take the sensing action. Sensing is never the most rewarding action in any of the states and can t hus be interpreted as an information-gaining action. The intelligent agent tends to sense when it is unsure of the environment. Also note that the intelligent agent not only has significantly fewer collisions with the benign agent than the heuristic agent but also correctly spoofs the benign agent out of its spectrum configuration more often. The goal of the decision-making agent is to maximize reward. In these results, this goal is achieved by minimizing the number of collisions, which also happens to have the greatest penalty.

V. Conclusion

POMDPs have been evaluated as a framework for radar decision making. One incarnation of the Incremental Pruning Algorithm has been applied for value iteration and computation of an optimal decision policy that maximizes discounted reward. Results were simulated for a scenario where a radar competes with a benign spectrum user for spectrum slots. These results demonstrate a clear improvement over maximizing immediate reward and suggest the usefulness of POMDPs for decision making and planning.

References

[1] Haykin, S., "Cognitive dynamic systems: radar, control, radio," *Proceedings of the IEEE*, Vol. 100, No. 7, July 2012, pp. 2095-2103.

[2] Ender, J.H.G., "Cognitive radar – enabling techniques for next generation radar systems," in *Proc. 16th Int'l Radar Symposium*, Vol. I, Dresden, Germany, June 2015, pp. 3-12.

[3] Guerci, J. R., "Cognitive radar: a knowledge-aided fully adaptive approach," in *Proc. 2010 IEEE Radar Conf.,* May 2010, pp. 1365-1370.

[4] Romero, R. A. and Goodman, N. A., "Cognitive radar network: cooperative adaptive beamsteering for integrated search-and-track application," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. *49, No.* 2, 2013, pp. 915-931.

[5] Zhang, J., Qiu, X., Shi, C., and Wu, Y., "Cognitive radar ambiguity function optimization for unimodular sequence," *EURASIP Journal on Advances in Signal Processing*, *2016*(1), 31.

[6] Charlish, A., and Hoffmann, F., "Anticipation in cognitive radar using stochastic control," in *Proc. IEEE Radar Conf.,* May 2015, pp. 1692-1697.

[7] Stinco, P., Greco, M., Gini, F., and Himed, B., "Cognitive radars in spectrally dense environments," *IEEE Aerospace and Electronic Systems Magazine*, *31*(10), 2016, pp. 20-27.

[8] Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1995, December). Partially observable markov decision processes for artificial intelligence. In *International Workshop on Reasoning with Uncertainty in Robotics* (pp. 146-163). Springer, Berlin, Heidelberg.

[9] Cassandra, A. R., Kaelbling, L. P., & Littman, M. L., "Acting optimally in partially observable stochastic domains," in *Assosiaction for the Advancement of Artificial Intelligence*, Vol. 94, October 1994, pp. 1023-1028.

[10] Cassandra, A. R. (1994). *Optimal policies for partially observable Markov decision processes*. Report CS-94-14, Brown Univ.

[11] Murphy, K. P. (2000). A survey of POMDP solution techniques. *Environment*, *2*, X3.

[12] Zhang, N. L., and Liu, W., *Planning in stochastic domains: problem characteristics and approximation*, Technical Report HKUST-CS96-31, Hong Kong University of Science and Technology, 1996.

[13] Cassandra, A., Littman, M. L., and Zhang, N. L., "Incremental pruning: A simple, fast, exact method for partially observable Markov decision processes," in *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence,* Morgan Kaufmann Publishers Inc., August 1997, pp. 54-61.

[14] Smallwood, R. D., & Sondik, E. J. (1973). "The optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, *21*(5), 1071-1088.

[15] White, C. C. (1991). "A survey of solution techniques for the partially observed Markov decision process," *Annals of Operations Research*, *32*(1), 215-230.

[16] Aberdeen, D. (2003). *A (revised) survey of approximate methods for solving partially observable Markov decision processes*. Technical report, National ICT Australia.