Approximating State Estimation in Multiagent Settings Using Particle Filters

Prashant Doshi and Piotr J. Gmytrasiewicz
Department of Computer Science
University of Illinois at Chicago, IL 60607
{pdoshi,piotr}@cs.uic.edu

ABSTRACT

State estimation consists of updating an agent's belief given executed actions and observed evidence to date. In single agent environments, the state estimation can be formalized using the Bayes filter. Exact estimation can be performed in simple cases, but approximate techniques, like particle filtering, have been used in more realistic cases. This paper extends the particle filter to multiagent settings resulting in the interactive particle filter. The main difficulty we tackle is that to fully represent an agent's beliefs in such environments, one has to specify probability distributions over the physical state and over the beliefs of other agents. This leads to interactive hierarchical belief systems first developed in game theory. Since the update of such beliefs proceeds recursively, the interactive particle filter samples and propagates on all levels of the belief hierarchy. We present algorithms, discuss some of their properties, and illustrate the performance of our implementation using simple examples.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Performance

Keywords

Multiagent state estimation, Bayesian Learning, Approximation, Particle Filters

1. INTRODUCTION

To act rationally, agents need to track the evolution of the state of the environment over time, based on the actions they perform and on the available observations. In realistic single agent and multiagent settings, the agent faces uncertainty due to both, imperfect sensors and non-deterministic action outcomes. This results in an incomplete knowledge about the true state of the world, usually represented as a probability distribution and called an agent's belief. In single agent settings, the state estimation is usually accomplished with a technique called the *Bayes filter* [15]. A Bayes filter allows the agent to maintain a belief about the state of the world at any given time, and update this belief each time an action is performed and new sensory information arrives. The convenience of this approach lies in the fact that the update is independent of the past percepts and action sequences. Technically, this means that the agent's belief is a *sufficient statistic*: it fully summarizes all of the information contained in past actions and observations.

The operation of a Bayes filter can be decomposed into a two-step process:

- *Prediction:* When an agent performs a new action, a^{t-1}, its prior belief state is updated:

$$Pr(s^t|a^{t-1}, b^{t-1}) = \int_{s^{t-1}} b^{t-1}(s^{t-1})T(s^t|s^{t-1}, a^{t-1})ds^{t-1}$$

- *Correction:* Thereafter, when an observation, o^t, is received, the intermediate belief state, $Pr(\cdot|a^{t-1}, b^{t-1})$, is corrected:

$$\begin{aligned} Pr(s^t|o^t, a^{t-1}, b^{t-1}) = \; & \alpha O(o^t|s^t, a^{t-1}) \\ & \times Pr(s^t|a^{t-1}, b^{t-1}) \end{aligned}$$

  where α is the normalizing constant, T and O are the transition and observation functions, respectively.

An important application of the Bayes filter is in single agent planning. Specifically, the Bayes filter performs the belief update step in the POMDP planning framework [12]. However, performing the exact update is possible only in the simplest environments [1] – in realistic cases one is forced to rely on approximations. One popular approximation is based on representing beliefs over continuous state spaces with a finite number of discrete samples, or particles. Particle filtering propagates the samples forward in time using the prediction and correction steps above, and generates an approximate representation of the agent's updated belief [17].

In multiagent settings matters are more complicated since others' actions also change the state of the system. A straightforward approach to extending the Bayes filter to the multiagent case is to consider other agents' actions as exogenous events (or noise) which occur with static probability distributions. However, this treats other agents as random devices and does not capture the other agents' belief updates over time which, of course, causes changes in their behavior.

A more refined approach is for the agent to maintain beliefs over the other agent's state. [2] The fact that the other agent's state can

[1] For continuous spaces strong assumptions of Gaussian beliefs and linear dynamics lead to Kalman filters and computable updates.

[2] For clarity we assume that the agent interacts with only one agent.

AAMAS'05 July 25-29, 2005, Utrecht, Netherlands

be described in terms of its beliefs [3] (as well as in terms of other factors, say, preferences, abilities, etc.) has two important consequences. First, the original agent's beliefs are necessarily defined over a continuous space so exact update is difficult. This suggests that sampling methods could be convenient approximations. Second, a nested hierarchy of beliefs obtains which includes other agent's beliefs over the original agent's beliefs, and so on. Such *interactive beliefs* have been the subject of much work in game theory and in theoretical computer science [1, 2, 3, 5, 10, 11, 14]. According to this approach, the multiagent state estimation problem, therefore, consists of keeping track of other's beliefs and of the physical state changes due to actions of all agents.

We explore the second approach and explicitly include comprehensive models of the interacting rational agents in the state space. The models encompass all information influencing the agent's behavior which includes its preferences, capabilities, and beliefs, and are analogous to *types* in Bayesian games [10]. In this representation, the state estimation filter recursively updates an agent's nested belief, given the agent's action and received observation. As in the case of POMDPs, important applications of the update are in autonomous agent planning in multiagent settings [7, 8].

Of course, as we mentioned, exact filtering is possible in only the simplest of cases and we propose using the particle filtering technique for approximating the multiagent state estimation filter. Mirroring the hierarchical character of interactive beliefs, our approach involves nested sampling at each of the hierarchical levels of beliefs. The result is a multiagent version of the basic bootstrap filter [4, 9] often used in single agent settings, which we call the *interactive particle filter*. Particle filters have been employed in collaborative multi-robot localization [6]; however, the emphasis was on predicting the position of the robot, and not the decisions and actions of the other robots. Additionally, to facilitate fast localization, beliefs of other robots encountered during motion were considered to be fully observable.

Rest of this paper is structured in the following manner. We introduce our multiagent state estimation filter in Section 2. In Section 3, we illustrate the operation of our filter using an example. We then present the interactive particle filter that approximates the multiagent state estimation, in Section 4. In Section 5, we empirically analyze the performance of our particle filter in two multiagent domains. We conclude this paper with a discussion and future work in Section 6.

2. MULTIAGENT STATE ESTIMATION

The formalism we present in this section follows [8]. Let us consider an agent i, that is interacting with one other agent j. Our arguments generalize to a setting with more than two agents in a straightforward manner. Consider a space of physical states S. Agent j's belief over S is then a probability distribution over S. We will call this a 0^{th} level belief, $b_{j,0}$. Additionally, j can be modeled by specifying its set of actions A_j, set of observations Ω_j, transition and observation functions, T_j and O_j, a reward function R_j, and an optimailty criterion OC_j. j's 0^{th} level model is then $\theta_{j,0} = \langle b_{j,0}, A_j, \Omega_j, T_j, O_j, R_j, OC_j \rangle$. The 0^{th} level models are therefore POMDPs with the other agent's actions folded in as *noise* into the T, O, and R functions. Agent i's first level beliefs are defined over the physical states of the world and the 0^{th} level models of agent j. We call this enriched state space a set of **interactive** states. In general, let $IS_{i,l}$ denote a set of interactive states defined as, $IS_{i,l} = S \times \Theta_{j,l-1}$, for $l \geq 1$, and $IS_{i,0} = S$, where S is the set of states of the physical environment, and $\Theta_{j,l-1}$, is the set of $(l-1)^{th}$ level *intentional models* of agent j. Let us rewrite $\theta_{j,l-1}$ as, $\theta_{j,l-1} = \langle b_{j,l-1}, \widehat{\theta}_j \rangle$, where $\widehat{\theta}_j \in \widehat{\Theta}_j$, is the agent j's *frame*. We give a recursive bottom-up construction of the state spaces below:

$$\begin{array}{ll} IS_{i,0} = S, & \Theta_{j,0} = \{\langle b_{j,0}, \widehat{\theta}_j \rangle : b_{j,0} \in \Delta(IS_{j,0})\}, \\ IS_{i,1} = S \times \Theta_{j,0}, & \Theta_{j,1} = \{\langle b_{j,1}, \widehat{\theta}_j \rangle : b_{j,1} \in \Delta(IS_{j,1})\}, \\ \vdots & \vdots \\ IS_{i,l} = S \times \Theta_{j,l-1}, & \Theta_{j,l} = \{\langle b_{j,l}, \widehat{\theta}_j \rangle : b_{j,l} \in \Delta(IS_{j,l})\}. \end{array}$$

A belief of agent i therefore comprises of finitely nested beliefs over others' models and their beliefs about others [1, 2, 3, 10, 11, 14]. [4] Increasing the value of l results in deeper levels of nesting.

One way to express agent i's beliefs nested to some level l is to assume that all probability density functions are polynomials. [5] We construct the belief representation in a bottom-up manner, as follows. i's level 0 belief is a probability distribution over S, i.e., a vector of length $|S|$. i's first level belief, which includes a distribution over j's level 0 beliefs is represented using a polynomial over j's level 0 beliefs, for each state and j's frame. Formally, $b_{i,1}$ is represented by $\{f^1_{i,1}, f^2_{i,1}, \ldots, f^{|S||\widehat{\Theta}_j|}_{i,1}\}$. Each polynomial $f^k_{i,1}$ can be written in a parametric form: $f^k_{i,1} = \langle d, c_1, c_2, \ldots, c_{(d+1)^{|S|-1}} \rangle$, where $d \in \mathbb{N}$ is the degree, and $c \in \mathbb{R}$ is a coefficient of $f^k_{i,1}$. To be a legal probability distribution the areas under the polynomials must sum to 1. i's level 2 belief is represented using a tuple of polynomials over the parameters of each of j's $|S||\widehat{\Theta}_i|$ level 1 polynomials ($f^k_{j,1}$), for each state and j's frame. Within each tuple, the first polynomial is defined over the degree d, of $f^k_{j,1}$. For each d, the remaining $(d+1)^{|S|-1}$ polynomials are defined over each coefficient of $f^k_{j,1}$. In other words, if j's level 1 belief is represented by $\{f^1_{j,1}, f^2_{j,1}, \ldots, f^{|S||\widehat{\Theta}_i|}_{j,1}\}$, then i's level 2 belief, $b_{i,2}$, is represented by $\{\{\langle f^1_{i,2}, f^2_{i,2}, \ldots \rangle^1, \ldots, \langle f^1_{i,2}, f^2_{i,2}, \ldots \rangle^{|S||\widehat{\Theta}_i|}\}_1, \ldots, \{\langle f^1_{i,2}, f^2_{i,2}, \ldots \rangle^1, \ldots, \langle f^1_{i,2}, f^2_{i,2}, \ldots \rangle^{|S||\widehat{\Theta}_i|}\}_{|S||\widehat{\Theta}_j|}\}$. Here the polynomials in each of the innermost tuples represent densities over the parameters of the corresponding level 1 polynomial of j (that with the same superscript as the tuple). We specify higher levels of beliefs in an analogous way.

As we mentioned before, there are two differences that complicate state estimation in multiagent settings, when compared to single agent ones. First, since the state of the physical environment depends on the actions performed by both agents, the prediction of how the physical state changes has to be made based on the predicted actions of the other agent. The probabilities of other's actions are obtained based on its models. Thus, we do not assume that actions are fully observable by other agents. Rather, agents can attempt to infer what actions other agents have performed by sensing their results on the environment. Second, changes in the models of other agents have to be included in the update. Specifically, in the case of intentional models, update of the other agent's beliefs due to its new observation has to be included. In other words, the agent has to update its beliefs based on what it anticipates that the other agent observes and how it updates.

For ease of comparison with the traditional Bayes filter, we decompose our multiagent state estimation filter into two steps:

[3]Including an agent's belief in the description of its state is, technically, not required. However, since actions depend on beliefs it is convenient to keep track of agent's beliefs to predict its actions. Predicting other agent's actions is, in turn, required to predict the physical state.

[4]Implicit in interactive beliefs is the assumption of coherency [3].

[5]We use polynomials as a well known representation capable of approximating any function over Euclidean space to arbitrary accuracy.

- *Prediction:* When an agent, say i, performs an action a_i^{t-1}, and the other agent j performs a_j^{t-1}, the predicted belief state is,

$$\begin{aligned}&Pr(is^t|a_i^{t-1}, a_j^{t-1}, b_{i,l}^{t-1}) = \int_{is^{t-1}:\hat{\theta}_j^{t-1}=\hat{\theta}_j^t} b_{i,l}^{t-1}(is^{t-1})\\ &\times Pr(a_j^{t-1}|\theta_{j,l-1}^{t-1})T(s^{t-1}, a_i^{t-1}, a_j^{t-1}, s^t)\\ &\times \textstyle\sum_{o_j^t} O_j(s^t, a_i^{t-1}, a_j^{t-1}, o_j^t)\\ &\times\delta(SE_{\hat{\theta}_j^t}(b_{j,l-1}^{t-1}, a_j^{t-1}, o_j^t) - b_{j,l-1}^t)d(is^{t-1})\end{aligned} \tag{1}$$

  where δ is the Dirac-delta function, $SE(\cdot)$ is an abbreviation for the belief update, $Pr(a_j^{t-1}|\theta_{j,l-1}^{t-1})$ is the probability that a_j^{t-1} is Bayes rational for the agent described by $\theta_{j,l-1}^{t-1}$, and T and O are the transition and observation functions respectively, extended to include actions of all agents.

- *Correction:* When agent i perceives an observation, o_i^t, the corrected belief state is a weighted sum of the predicted belief states for each possible action of j,

$$\begin{aligned}&Pr(is^t|o_i^t, a_i^{t-1}, b_{i,l}^{t-1}) = \alpha\textstyle\sum_{a_j^{t-1}} O_i(s^t, a_i^{t-1}, a_j^{t-1}, o_i^t)\\ &\times Pr(is^t|a_i^{t-1}, a_j^{t-1}, b_{i,l}^{t-1})\end{aligned} \tag{2}$$

  where α is the normalizing constant.

In the multiagent state estimation presented above, when j is modeled using an intentional model, i's belief update will invoke j's belief update (via the term $SE_{\hat{\theta}_j^t}(b_{j,l-1}^{t-1}, a_j^{t-1}, o_j^t)$), which in turn will invoke i's belief update and so on. This recursion in belief nesting will bottom out at the 0^{th} level. At this level, belief update of the agent reduces to a Bayes filter.

In a manner similar to the single agent Bayes filter, the following proposition holds for the multiagent state estimation filter. The proposition results from noting that Eq. 2 expresses the belief in terms of parameters of the previous time step only. A complete proof of the state estimation filter and this proposition appears in [8].

PROPOSITION 1 (SUFFICIENCY). *Agent i's current belief, i.e. probability distribution over the set $S \times \Theta_{j,l-1}$ is a sufficient statistic for the past history of i's observations.*

3. EXAMPLE: MULTIAGENT TIGER GAME

We analyze the operation of our multiagent state estimation filter using a simple example.

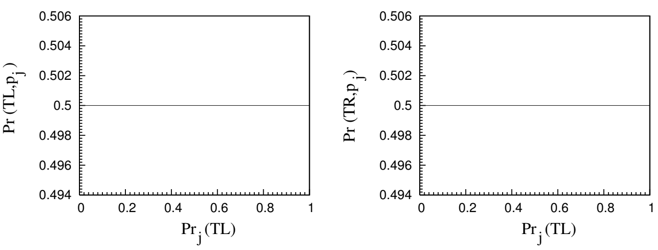

Figure 1: Example level 1 belief of agent i. Here, i is uninformed about j's beliefs.

3.1 Description

The multiagent tiger game resembles a game-show like situation in which there are two doors, behind one is a tiger ready to eat anyone who opens the door, and behind another is a pot of gold. The game is played by two agents, i and j, who are unaware of where the gold is, and each can either open any one of the doors, or listen. The tiger emits a growl periodically, which reveals its position but only with some certainty. Additionally, each agent can also hear a creak with some certainty, if the other agent opens a door. We will assume that neither agent can perceive other's observations nor actions. The transition, reward, and observation functions describing the game appear in [8]. In such a situation, what is the sequence of actions conditioned on observations that each agent should take to maximize its gains over a period of fixed number of steps?

When an agent makes its choice in the multiagent tiger game, it has to consider what it believes about the location of the tiger, as well as whether the other agent will listen or open a door, which in turn depends on the other agent's beliefs. For the purpose of illustration, let us now make a simplifying assumption: Agent i is aware of all of agent j's properties except its belief. Therefore, i is uncertain only about j's beliefs.

An example level 1 belief state of i is one in which i is uninformed about j's beliefs, and believes with equal probability that the tiger is behind one of the two doors. In Fig. 1, we show this belief state. Using polynomials, we express this belief, $b_{i,1}$, using the polynomials $\{f_{i,1}^{TL}, f_{i,1}^{TR}\}$, where the polynomials $f_{i,1}^{TL} = f_{i,1}^{TR} = \langle 0, 0.5\rangle$. Note that i's belief over the location of the tiger is computed by marginalizing away j's beliefs. As an example level 2 belief of i, consider a belief in which i considers increasingly complex level 1 beliefs of j (i.e. $f_{j,1}$ with increasing degrees) less likely (Occam's Razor), and is uninformed of the location of the tiger. We express the level 2 belief, $b_{i,2}$, using $\{\{\langle f_{i,2}^1, f_{i,2}^2, \ldots\rangle^{TL}, \langle f_{i,2}^1, f_{i,2}^2, \ldots\rangle^{TR}\}_{TL}, \{\langle f_{i,2}^1, f_{i,2}^2, \ldots\rangle^{TL}, \langle f_{i,2}^1, f_{i,2}^2, \ldots\rangle^{TR}\}_{TR}\}$. Here, $f_{i,2}^1$ in each tuple is the parametric form of the normalized Taylor series expansion of 2^{-d} defined over the degree d of j's level 1 polynomials: $\alpha\sum_{n=0}^{\infty}\frac{1}{k!}(-1)^n ln(2)^n(d-d_{max})^n$, where α is the normalizing constant and d_{max} is an upper bound on d. [6] The remaining upto $d_{max}+1$ polynomials in each tuple are p.d.f.s over the coefficients of j's level 1 polynomials and are of degree 0.

3.2 Operation of the State Estimation Filter

Let us trace the exact filtering process for a simple level 1 belief using the multiagent tiger game. In Fig. 2, we display an example trace of the state estimation that starts with a prior belief and then assumes that agent i listens and hears a growl from the left and no creaks.

As we mentioned, state estimation for agent i involves solving $\theta_{j,0}$ which is a traditional POMDP in which we assume that i's actions are "folded" in with a probability of 0.33, 0.33, and 0.33 for OL, L, and OR, respectively (akin to ascribing a *no-information* model to agent i). The corresponding two steps to go policy of j prescribes OL when $b_{j,0} < 0.1$, L when $0.1 < b_{j,0} < 0.9$, and OR when $0.9 < b_{j,0} < 1$. When $b_{j,0} = 0.1$, OL and L are executed with a probability of 0.5 each, and at 0.9, L and OR are executed with equal probabilities.

Fig. 2(a) depicts the prior according to which i is uninformed about j's beliefs. Fig. 2(b) is the result of the *prediction* step after i's listening action (L). The first column of (b) shows i's belief after it has listened and given that j also listened (j listens when its belief is between 0.1 and 0.9). The two observations j can make, GL and GR, each with probability dependent on the tiger's location, give rise to flat portions representing what i knows about j's belief in each case. The second column of (b) shows i's belief after i has

[6] We use $2^{-K(x)}$ where $K(\cdot)$ is the Kolmogorov complexity as a mathematical formalization of Occam's razor [13].

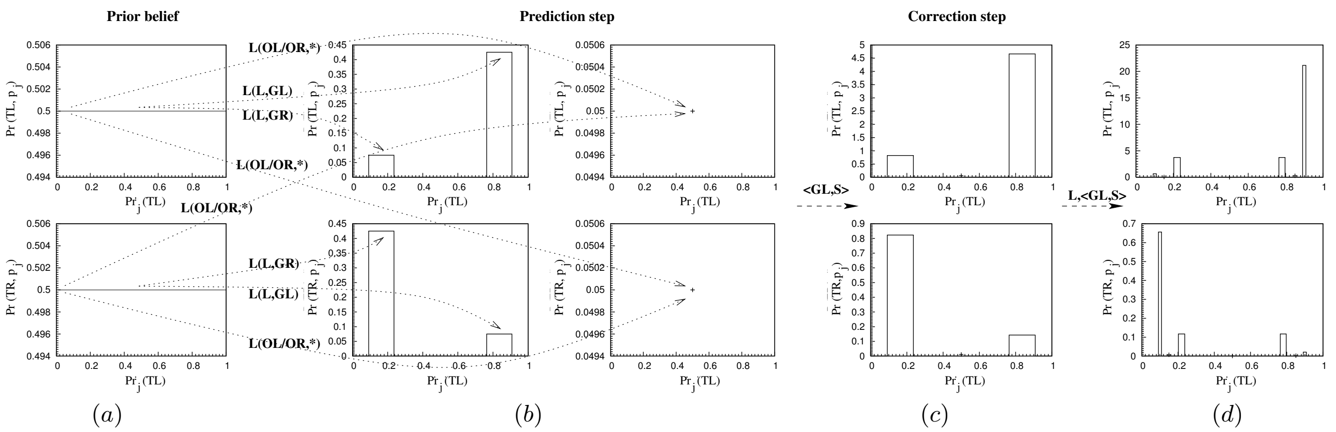

Figure 2: State estimation in the multiagent tiger game. (a) is the level 1 prior belief. (b) is the result of *prediction* given i's listening action, and a pair denoting j's action and observation. (c) is the result of *correction* after i observes $\langle GL, S\rangle$. (d) depicts the result of another belief update after i listens and receives $\langle GL, S\rangle$.

listened and j has opened the left (j's belief is less than or equal to 0.1) or right door (j's belief is greater than or equal to 0.9). The plots are identical for each action and only one of them is shown. In this case, i knows that j has no information about the tiger's location in this case. Fig. 2(c) is the result of *correction* after i observes tiger's growl on the left and no creaks ⟨GL,S⟩. The plots in (c) are obtained by performing a weighted summation of the plots in (b). The probability i assigns to TL (0.85) is now greater than TR (0.15), and information about j's beliefs allows i to refine its prediction of j's action in the next time step. In particular, while agent i in column (a) believed that j would listen, open-left, and open-right with probabilities 0.8,0.1,and 0.1 respectively, in column (c), i believes these probabilities to be 0.95, 0.013, and 0.037 respectively. Fig. 2(d) illustrates the posterior after another step of the belief update during which i again listens and hears a ⟨GL,S⟩.

4. INTERACTIVE PARTICLE FILTER

As we mentioned, there is a continuum of intentional models of an agent. Since an agent is unaware of the true models of interacting agents *ex ante*, it must maintain a belief over all possible candidate models. This precludes practical implementations of our multiagent state estimation filter for all but the simplest settings, and approximate implementations are critically required.

The general algorithm for the basic particle filter appears in [4]. We shall concentrate on a specific implementation of this algorithm, that has previously been studied under various names such as Monte Carlo localization, survival of the fittest, and bootstrap filter. We extend this implementation to the multiagent state estimation filter presented in Section 2, and call it the interactive particle filter. Our extension to the multiagent case, similar to basic particle filtering, requires the key steps of *importance sampling* and *selection*. The resulting algorithm, inherits the convergence properties of the original algorithm [4]. The extension to the multiagent setting turns out to be non-trivial because we are faced with an interactive belief hierarchy.

The interactive particle filter described in Fig. 4 requires an initial set of N particles, $\widetilde{b}^{t-1}_{k,l}$, that is approximately representative of the agent's belief, along with the action, a^{t-1}_k, the observation, o^t_k, and the level of belief nesting, $l > 0$. Each particle in the sample set represents the agent's possible interactive state. Here, k will stand for either agent i or j, and $-k$ for the other agent, j or i, as appropriate. We generate $\widetilde{b}^{t-1}_{k,l}$ by recursively sampling N particles from beliefs represented using polynomials at each level of nesting (see the PRIORSAMPLE algorithm in Fig. 3). For simplicity of presentation, we assume that the agent uses the same polynomial for each state and other's frame. This means that the agent believes each physical state and other's frame to be equally likely. The procedure generalizes immediately to the case where the agent

Function PRIORSAMPLE($\{f^1_{k,l}, \ldots, f^I_{k,l}\}$, $l > 0$) **returns** $\widetilde{b}_k$

for n **from** 1 **to** N **do**
  Sample $s^{(n)} \sim \mathcal{U}(1, 2, \ldots, |S_k|)$, $\widehat{\theta}^{(n)}_{-k} \sim \mathcal{U}(1, 2, \ldots, |\widehat{\Theta}_{-k}|)$
  if (l=1) **then**
    Sample $b^{(n)}_{-k} = \langle Pr(s_1), Pr(s_2), \ldots, Pr(s_{|S|-1})\rangle \sim f^1_{k,1}$
    $is^{(n)}_k \leftarrow \langle s^{(n)}, \langle b^{(n)}_{-k}, \widehat{\theta}^{(n)}_{-k}\rangle\rangle$
  else if (l=2) **then**
    Sample $d \sim f^1_{k,2}$
    for i **from** 1 **to** $(d+1)^{|S|-1}$ **do**
      Sample $c_i \sim f^{i+1}_{k,2}$
    $f_{-k,1} \leftarrow \langle d, c_1, \ldots, c_{(d+1)^{|S|-1}}\rangle$
    $\widetilde{b}^{(n)}_{-k} \leftarrow$ PRIORSAMPLE($\{f_{-k,1}\}$, 1)
    $is^{(n)}_k \leftarrow \langle s^{(n)}, \langle \widetilde{b}^{(n)}_{-k}, \widehat{\theta}^{(n)}_{-k}\rangle\rangle$
  else
    for i **from** 1 **to** $(1 + (d_{max}+1))^{l-2} \cdot (1 + (d_{max}+1)^{|S|-1})$ **step** $1 + (d_{max}+1)$ **do**
      Sample $d \sim f^i_{k,l}$
      for j **from** 1 **to** $(d+1)$ **do**
        Sample $c_j \sim f^{i+j}_{k,l}$
      $f_{-k,l-1} \overset{\cup}{\leftarrow} \{\langle d, c_1, \ldots, c_{d+1}\rangle\}$
    $\widetilde{b}^{(n)}_{-k} \leftarrow$ PRIORSAMPLE($f_{-k,l-1}$, $l-1$)
    $is^{(n)}_k \leftarrow \langle s^{(n)}, \langle \widetilde{b}^{(n)}_{-k}, \widehat{\theta}^{(n)}_{-k}\rangle\rangle$
  $\widetilde{b}_k \overset{\cup}{\leftarrow} \{is^{(n)}_k\}$
return $\widetilde{b}_k$
end function

Figure 3: Algorithm for sampling from nested beliefs using a polynomial, $f^k_{i,l} = \langle d, c_1, c_2, \ldots\rangle$, based representation for the p.d.f.s. at each level. Here, k denotes either agent i or j, and $-k$ denotes j or i respectively.

Function I-PARTICLEFILTER($\widetilde{b}^{t-1}_{k,l}, a^{t-1}_k, o^t_k, l > 0$)
returns $\widetilde{b}^t_{k,l}$
1. $\widetilde{b}^{tmp}_{k,l} \leftarrow \phi, \widetilde{b}^t_{k,l} \leftarrow \phi$
Importance Sampling
2. **for all** $is^{(n),t-1}_k = \langle s^{(n),t-1}, \theta^{(n),t-1}_{-k} \rangle \in \widetilde{b}^{t-1}_{k,l}$ **do**
3. $Pr(A_{-k}|\theta^{(n),t-1}_{-k}) \leftarrow$ SOLVEMODEL($\theta^{(n),t-1}_{-k}, l-1$)
4. Sample $a^{t-1}_{-k} \sim Pr(A_{-k}|\theta^{(n),t-1}_{-k})$
5. Sample $s^{(n),t} \sim T_k(S^t|a^{t-1}_k, a^{t-1}_{-k}, s^{(n),t-1})$
6. **for all** $o^t_{-k} \in \Omega_{-k}$ **do**
7. **if** (l = 1) **then**
8. $b^{(n),t}_{-k} \leftarrow$ LEVEL0BELIEFUPDATE($b^{(n),t-1}_{-k}, a^{t-1}_{-k}, o^t_{-k}$)
9. $\theta^{(n),t}_{-k} \leftarrow \langle b^{(n),t}_{-k}, \widehat{\theta}^{(n)}_{-k} \rangle$
10. $is^{(n),t}_k \leftarrow \langle s^{(n),t}, \theta^{(n),t}_{-k} \rangle$
11. **else**
12. $\widetilde{b}^{(n),t}_{-k} \leftarrow$ I-PARTICLEFILTER($\widetilde{b}^{(n),t-1}_{-k}, a^{t-1}_{-k}, o^t_{-k}, l-1$)
13. $\theta^{(n),t}_{-k} \leftarrow \langle \widetilde{b}^{(n),t}_{-k}, \widehat{\theta}^{(n)}_{-k} \rangle$
14. $is^{(n),t}_k \leftarrow \langle s^{(n),t}, \theta^{(n),t}_{-k} \rangle$
15. Weight $is^{(n),t}_k$: $w^{(n)}_t = O_{-k}(o^t_{-k}|s^{(n),t}, a^{t-1}_k, a^{t-1}_{-k})$
16. Adjust weight: $w^{(n)}_t = w^{(n)}_t \times O_k(o^t_k|s^{(n),t}, a^{t-1}_k, a^{t-1}_{-k})$
17. $\widetilde{b}^{tmp}_{k,l} \stackrel{\cup}{\leftarrow} (is^{(n),t}_k, w^{(n)}_t)$
18. Normalize all $w^{(n)}_t$ so that $\sum_{n=1}^{N} w^{(n)}_t = 1$
Selection
19. Resample with replacement N particles $\{is^{(n),t}_k, n = 1...N\}$ from the set $\widetilde{b}^{tmp}_{k,l}$ according to the importance weights.
20. $\widetilde{b}^t_{k,l} \leftarrow \{is^{(n),t}_k, n = 1...N\}$
21. **return** $\widetilde{b}^t_{k,l}$
end function

Figure 4: Interactive particle filtering for approximating the multiagent state estimation filter. A nesting of particle filters is used to update all levels of the belief.

Function LEVEL0BELIEFUPDATE($b^{t-1}_k, a^{t-1}_k, o^t_k$) **returns** b^t_k
1. $Pr(a^{t-1}_{-k}) \leftarrow 1/a^{t-1}_{-k}$
2. **for all** $s^t \in S$ **do**
3. sum $\leftarrow 0$
4. **for all** $s^{t-1} \in S$ **do**
5. $Pr(s^t|s^{t-1}, a^{t-1}_k) \leftarrow 0$
6. **for all** $a^{t-1}_{-k} \in A_{-k}$ **do**
7. $Pr(s^t|s^{t-1}, a^{t-1}_k) \stackrel{+}{\leftarrow} T_k(s^t|s^{t-1}, a^{t-1}_k, a^{t-1}_{-k}) Pr(a^{t-1}_{-k})$
8. sum $\stackrel{+}{\leftarrow} Pr(s^t|s^{t-1}, a^{t-1}_k) b^{t-1}_k(s^{t-1})$
9. $Pr(o^t_k|s^t, a^{t-1}_k) \leftarrow 0$
10. **for all** $a^{t-1}_{-k} \in A_{-k}$ **do**
11. $Pr(o^t_k|s^t, a^{t-1}_k) \stackrel{+}{\leftarrow} O_k(o^t_k|s^t, a^{t-1}_k, a^{t-1}_{-k}) Pr(a^{t-1}_{-k})$
12. $b^t_k(s^t) \leftarrow Pr(o^t_k|s^t, a^{t-1}_k) \times$ sum
13. Normalize the belief, b^t_k
14. **return** b^t_k
end function

Function SOLVEMODEL($\theta_k, l > 0$) **returns** $Pr(A_k|\theta_k)$
1. $\widetilde{b}^0_{k,l} \leftarrow \{is^{(n)}_k, n = 1...N | is^{(n)}_k \sim b_{k,l} \in \theta_k\}$
Reachability Analysis
2. reach(0) $\leftarrow \mathrm{S}_{b^0_k}$
3. **for** $t \leftarrow 1$ **to** T **do**
4. reach(t) $\leftarrow \phi$
5. **for all** $\widetilde{b}^{t-1}_k \in$ reach($t-1$), $a_k \in A_k$, $o_k \in \Omega_k$ **do**
6. reach(t) $\stackrel{\cup}{\leftarrow}$ I-PARTICLEFILTER($\widetilde{b}^{t-1}_{k,l}, a_k, o_k, l$)
Value Iteration
7. **for** $t \leftarrow T$ **downto** 0 **do**
8. **for all** $\widetilde{b}^t_{k,l} \in$ reach(t) **do**
9. U($\widetilde{b}^t_{k,l}$) $\leftarrow -\infty$, OPT($\widetilde{b}^t_{k,l}$) $\leftarrow \phi$
10. **for all** $a_k \in A_k$ **do**
11. $\mathrm{U}^{a_k}(\widetilde{b}^t_{k,l}) \leftarrow 0$
12. **for all** $is^{(n),t}_k = \langle s^{(n),t}, \theta^{(n)}_{-k} \rangle \in \widetilde{b}^t_{k,l}$ **do**
13. $Pr(A_{-k}|\theta^{(n)}_{-k}) \leftarrow$ SOLVEMODEL($\theta^{(n)}_{-k}, l-1$)
14. **for all** $a_{-k} \in A_{-k}$ **do**
15. $\mathrm{U}^{a_k}(\widetilde{b}^t_{k,l}) \stackrel{+}{\leftarrow} \frac{1}{N} R(s^{(n),t}, a_k, a_{-k}) Pr(a_{-k}|\theta^{(n)}_{-k})$
16. **if** ($t < T$) **then**
17. **for all** $o_k \in \Omega_k$ **do**
18. sum $\leftarrow 0$
19. **for all** $is^{(n),t}_k = \langle s^{(n),t}, \theta^{(n)}_{-k} \rangle \in \widetilde{b}^t_{k,l}$ **do**
20. $Pr(A_{-k}|\theta^{(n)}_{-k}) \leftarrow$ SOLVEMODEL($\theta^{(n)}_{-k}, l-1$)
21. **for all** $a_{-k} \in A_{-k}$, $s^{t+1} \in S_k$ **do**
22. sum $\stackrel{+}{\leftarrow} O_k(o_k|s^{t+1}, a_k, a_{-k}) \times T(s^{t+1}|s^{(n),t}, a_k, a_{-k}) Pr(a_{-k}|\theta^{(n)}_{-k})$
23. $\mathrm{U}^{a_k}(\widetilde{b}^t_{k,l}) \stackrel{+}{\leftarrow} \frac{1}{N} \times$ sum $\times$ U(reach(t)[$|\Omega_k| a_k + o_k$])
24. **if** ($\mathrm{U}^{a_k}(\widetilde{b}^t_{k,l}) \geq \mathrm{U}(\widetilde{b}^t_{k,l})$) **then**
25. **if** ($\mathrm{U}^{a_k}(\widetilde{b}^t_{k,l}) > \mathrm{U}(\widetilde{b}^t_{k,l})$) **then**
26. U($\widetilde{b}^t_{k,l}$) $\leftarrow \mathrm{U}^{a_k}(\widetilde{b}^t_{k,l})$
27. OPT($\widetilde{b}^t_{k,l}$) $\leftarrow \phi$
28. OPT($\widetilde{b}^t_{k,l}$) $\stackrel{\cup}{\leftarrow} a_k$
29. **for all** $a_k \in A_k$ **do**
30. **if** ($a_k \in$ OPT($\widetilde{b}^0_{k,l}$)) **then**
31. $Pr(a_k|\theta_k) \leftarrow \frac{1}{|\mathrm{OPT}(\widetilde{b}^0_{k,l})|}$
32. **else**
33. $Pr(a_k|\theta_k) \leftarrow 0$
34. **return** $Pr(A_k|\theta_k)$
end function

Figure 5: Algorithm for recursively solving an agent's model given its nested belief. When $l = 0$, SOLVEMODEL reduces to solving a POMDP given an initial belief which is carried out exactly.

believes that certain states and frames are more likely than others. The interactive particle filtering proceeds by *propagating* each particle forward in time. However, as opposed to the basic particle filtering, this is not a one-step process. In order to perform the propagation, other agent's action must be known. This is obtained by solving the other agent's model (using the algorithm SOLVEMODEL described in Fig. 5) to get its policy, and using its belief (contained in the particle) to find a distribution over its actions (line 3 in Fig. 4). Additionally, analogously to the exact belief update, for each of the other agent's possible observations, we must obtain its next belief state (line 6). If $l > 1$, updating the other agent's belief requires invoking the particle filter for performing its belief update (lines 12–14). This recursion in depth of the belief nesting terminates when the level of nesting becomes one, and a level 0 belief update is performed (lines 8–10). Though the propagation step generates $|\Omega_{-k}|N$ appropriately weighted particles, we *resample* N particles out of these (line 19), using an unbiased resampling scheme. A visualization of our implementation is shown in Fig. 6.

Let us now analyze the gains in computation we will accrue us-

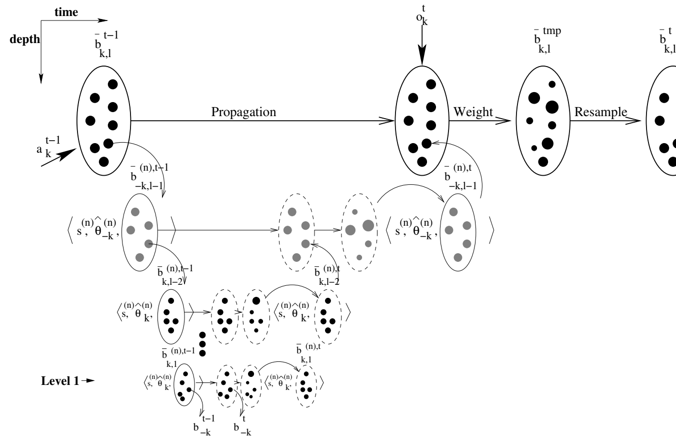

Figure 6: An illustration of the nesting in the interactive particle filter. Colors black and gray distinguish filtering for the two agents. Because the propagation step involves updating the other agent's beliefs, we perform particle filtering on its beliefs. The filtering terminates when it reaches the level 1 nesting, where an exact POMDP (level 0) belief update is performed for the other agent.

ing our algorithm. First, we focus on the reduction in the number of agent models that must be solved. In an M+1-agent setting with the number of particles bounded by N, each particle in $\widetilde{b}_{k,l}^{t-1}$ of level l will have M models of level $l-1$. Solution of each of these level $l-1$ models will require solving the lower level models recursively. The upper bound on the number of models that will be solved is $O((MN)^{l-1})$. Taking into account that there are M level $l-1$ models in a particle, and N such possibly distinct particles, we are required to solve $O((MN)^l)$ models. Our upper bound is polynomial in M, compared to $O((M|\Theta_*|^M)^l)$ models that need to be solved in the exact case, which is exponential in M. Here, amongst the spaces of models of all agents, Θ_* is the largest space. Typically, $N << |\Theta_*|^M$, resulting in a substantial reduction in computation. Our filtering algorithm demonstrates substantial computational gains even when we assume that solutions of models are available in constant time (computed offline). Referring to Fig. 4, in a setting of M+1 agents, recursive calls to the algorithm will be made $O(NM|\Omega_*|)$ times, and the recursion terminates when the LEVEL0BELIEFUPDATE is executed which has time complexity $O(|S|^2|A_*|^M)$. Here, A_* and Ω_* is the largest action and observation space respectively, among all such spaces of the other agents. The total time complexity of our particle filtering algorithm is $O((NM|\Omega_*|)^l|S|^2|A_*|^M)$, compared to that of the exact case which is $O((|S||\Theta_*|^M)^2|A_*|^M M|\Omega_*|)^l|S|^2|A_*|^M)$.

5. PERFORMANCE

As part of our empirical investigation of the performance of the interactive particle filter, we show, using a standard distance metric and visually, that the particle filter approximates the exact multiagent state estimation filter closely. For our analysis, we utilize the two-agent tiger game, that has two physical states, as described in Section 3, and a two-agent version of the machine maintenance problem(MM) [16], that has three physical states. For both these problems, we make the simplifying assumption that models of the other agent differ only in their beliefs. We use a numerical integration implementation [7] for the exact filter as the baseline for comparison. Because the test problems are rather simplistic (tiger: $|S|$=2, $|A_i|$=$|A_j|$=3, $|\Omega_i|$=$|\Omega_j|$=6; MM: $|S|$=3, $|A_i|$=$|A_j|$=4, $|\Omega_i|$=$|\Omega_j|$=2), our results should be considered preliminary.

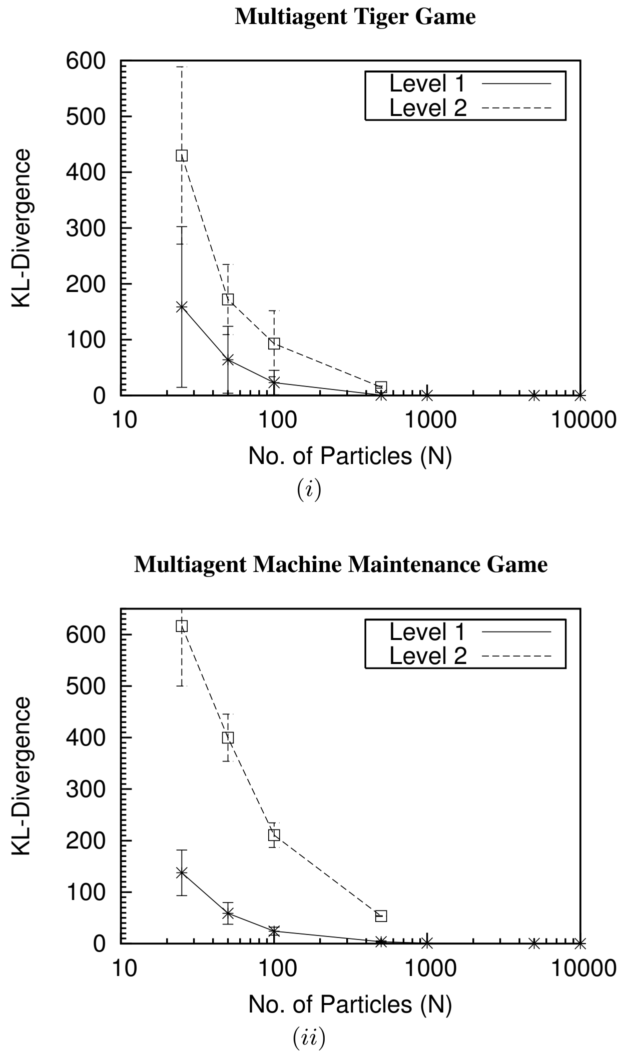

Figure 8: Performance of the interactive particle filter as a function of the number of particles, on (i) multiagent tiger game, (ii) multiagent machine maintenance game.

The lineplots in Fig. 8 show that the quality of the approximation, as measured by KL-Divergence [8], increases as the number of particles increases, for both the problem domains. As we may expect, level 2 belief approximations require considerably more particles as compared to level 1 approximations, to achieve similar performance. Also, note that the performance of the interactive particle filter remains consistent for both the two-state tiger and the three-state MM problem indicating that our implementation is not affected by the dimensionality of the underlying state space. Each data point in the lineplots is the average of 10 runs of our particle filter. In the case of the tiger game, the posterior used for comparison is the one that is obtained after agent i listens and hears a growl left and no creaks (Fig. 2(c)). For the machine maintenance game,

[7]We obtained the points for numerical integration by superimposing a high resolution regular grid on the interactive state space.

[8]When the level of nesting of the beliefs is greater than one, we compute the sum of the average of the KL-Divergences of the lower level beliefs and the upper level KL-Divergence.

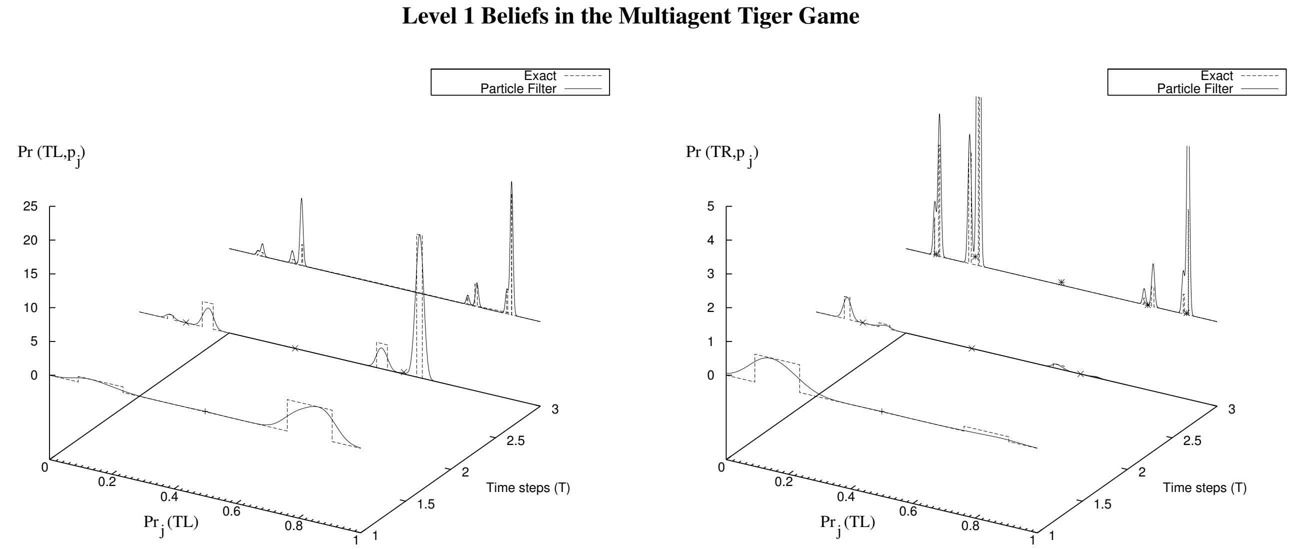

Figure 7: The exact and approximate p.d.f.s after successive filtering steps. The peaks of the approximate p.d.f.s align correctly with those of the exact p.d.f.s, and the areas under the approximate and exact p.d.f.s are approximately equal.

Belief	Game	Particle Filter		Numerical
		N=500	N=1000	Integration
Level 1	Multiagent Tiger	0.148s ± 0.001s	0.332s ± 0.007s	21.80s ± 0.036s
	Multiagent MM	0.452s ± 0.009s	0.931s ± 0.0146s	1m 18.20s ± 0.45s
Level 2	Multiagent Tiger	2m 23.28s ± 1.1s	11m 41.30s ± 1.52s	51m 12.24s ± 5.66s
	Multiagent MM	1m 37.59s ± 0.17s	8m 27.29s ± 1.65s	151m 29.48s ± 1m 55.73s

Table 1: Comparison of the average running times of our numerical integration and interactive particle filter implementations on same the platform (*Pentium 4, 1.7GHz, 512MB RAM, Linux*).

the posterior obtained after i manufactures and perceives no defect in the product, is used for comparison. We selected the belief state in Fig. 1 as the prior level 1 belief of agent i when playing the tiger game, and analogously for the machine maintenance game. Prior level 2 belief of i is the one mentioned in Section 3 with $d_{max} = 3$. A comparison of the run times of the state estimation filter implemented using numerical integration and the interactive particle filter is shown in Fig. 1. Our interactive particle filter implementation significantly outperforms the numerical integration based implementation, while providing comparable performance quality. Additionally, the run times of the numerical integration implementation significantly increase when we move from the two-state tiger problem to the three-state machine maintenance problem, in contrast to the particle filter. This is because numerical integration requires more points for larger state spaces to maintain comparable quality. In order to assess the quality of the approximations after successive belief updates, we graphed the p.d.f.s produced by the interactive particle filter and the exact filter. The p.d.f.s arising after each of three filtering steps on the level 1 belief of agent i in the tiger game, are shown in Fig 7. Each approximate p.d.f. is the average of 10 runs of the particle filter which contained 5000 particles, and is estimated using a standard Gaussian kernel. The action/observation sequence followed was $\langle L, GL, S\rangle$, $\langle L, GL, S\rangle$, $\langle OR, GL, S\rangle$. As can be seen, the interactive particle filter produces a good approximation of the true densities.

6. DISCUSSION

Bayes filter has become a cornerstone in the area of single agent state estimation for partially observable settings. It finds important applications in robot localization and tracking, as well as POMDP planning. However, no analogous state estimation filter existed for multiagent settings. The difficulty lies in modeling the uncertainty in others' decisions that entails reasoning with nested interactive belief systems. In this paper, we have filled this gap, by proposing a novel multiagent state estimation process for a rational setting that ascribes sophisticated nested models to others, predicts their actions using these models, and tracks these models by recursively updating them.

The agent possibly not knowing the true models of other agents *a'priori* forces it to maintain beliefs over a continuum of other agents' models, thereby precluding practical implementations of our state estimation filter for realistic settings. To address this problem, we use interactive particle filtering to implement the multiagent state estimation process approximately. Reflecting the nested nature of the interactive belief system, the interactive particle filter maintains sampled representations of belief at all levels of nesting, and propagates them forward in time. Our experimental results on simple problem domains indicate reasonable performance, both in terms of quality of approximation as well as speed of computation. These positive results pave the way to experiments in more substantive applications, and this is one line of our future work. Another direction of research is to utilize the interactive particle filter in approximate planning in multiagent settings. This is analogous to the use of the traditional particle filter in POMDP planning [17].

7. ACKNOWLEDGEMENTS

This research is supported by the National Science Foundation CAREER award IRI-9702132, and NSF award IRI-0119270.

8. REFERENCES

[1] R. J. Aumann and A. Heifetz. *Handbook of Game Theory with Economic Applications*, volume 3. Elsevier Science, 2002.

[2] P. Battigalli and G. Bonanno. Recent results on belief, knowledge, and the epistemic foundations of game theory. *Research in Economics*, 53:149–225, 1999.

[3] A. Brandenburger and E. Dekel. Hierarchies of beliefs and common knowledge. *Journal of Economic Theory*, 59:189–198, 1993.

[4] A. Doucet, N. D. Freitas, and N. Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[5] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots on Heterogenous Multi-Robot Systems*, 8(3), 2000.

[7] P. Gmytrasiewicz and P. Doshi. Interactive pomdps: Properties and preliminary results. In *AAMAS*, pages 1374–1375, NYC, NY, 2004.

[8] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of AI Research*, 23, 2005.

[9] N. Gordon, D. Salmond, and A. Smith. Novel approach to non-linear/non-gaussian bayesian state estimation. *IEEE Proceedings-F*, 140(2):107–113, 1993.

[10] J. C. Harsanyi. Games with incomplete information played by 'bayesian' players. *Mgmt. Science*, 14(3):159–182, 1967.

[11] A. Heifetz and D. Samet. Topology-free typology of beliefs. *Journal of Economic Theory*, 82:324–341, 1998.

[12] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 2, 1998.

[13] M. Li and P. Vitanyi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, 1997.

[14] J. Mertens and S. Zamir. Formulation of bayesian analysis for games with incomplete information. *International Journal of Game Theory*, 14:1–29, 1985.

[15] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (Second Edition)*. Prentice Hall, 2003.

[16] R. Smallwood and E. Sondik. The optimal control of partially observable markov decision processes over a finite horizon. *Operations Research*, 21:1071–1088, 1973.

[17] S. Thrun. Monte carlo pomdps. In *NIPS 12*, pages 1064–1070, 2000.