Approximating Reachable Belief Points in POMDPs

Kyle Hollins Wray and Shlomo Zilberstein

***Abstract*— We propose an algorithm called σ-approximation that compresses the non-zero values of beliefs for partially observable Markov decision processes (POMDPs) in order to improve performance and reduce memory usage. Specifically, we approximate individual belief vectors with a fixed bound on the number of non-zero values they may contain. We prove the correctness and a strong error bound when the σ-approximation is used with the point-based value iteration (PBVI) family algorithms. An analysis compares the algorithm on six larger domains, varying the number of non-zero values for the σ-approximation. Results clearly demonstrate that when the algorithm used with PBVI (σ-PBVI), we can achieve over an order of magnitude improvement. We ground our claims with a full robotic implementation for simultaneous navigation and localization using POMDPs with σ-PBVI.**

I. Introduction

Automated planning domains have been steadily growing in complexity, especially for partially observable Markov decision processes (POMDPs) [1]. They now encapsulate problems ranging from water reservoir control [2] to autonomous vehicles [3], [4]. The growing number of possible states and observations in these problem domains requires POMDP solvers to handle a large space of agent's beliefs over domain states. The complexity of planning has inspired the development of numerous approximate planning algorithms.

One approximation method that proved particularly effective is point-based value iteration (PBVI) [5], which restricts value function computations to a subset of the belief space, thereby accelerating *value iteration* techniques [6], [7], [8], [9], [10], [11]. We propose an algorithm called σ-approximation that exploits a bounded quantity of zero-values over the set of beliefs to greatly improve belief operations in POMDP algorithms.

The σ-approximation method addresses an orthogonal issue from PBVI; both methods can, in fact, be used together or separately. PBVI concerns itself with the number of reachable beliefs and the selection of an approximate subset. Our algorithm focuses on the number of non-zero values *within* each belief point. Specifically, we construct a new set of beliefs to use for updates given a non-zero value constraint r_z (e.g., $r_z \approx \log n$, where n is the number of states). For each belief, we sort the belief values and select only the top r_z values, then normalize these values to create a new belief. These are then used in update equations, allowing for dot products with beliefs to be computed much faster based on this constraint r_z. We formally show that this simple routine is the optimal projection given the r_z constraint. Then, we prove a strong bound on the error for the σ-approximation used in point-based algorithms. Finally, we demonstrate its vast performance gains with low error in six larger domains.

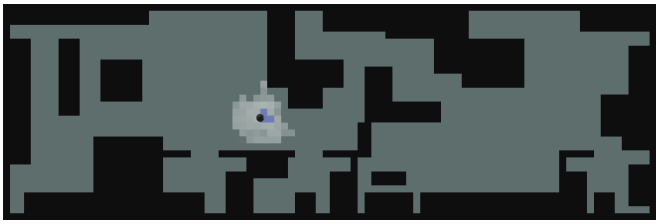

Fig. 1. Example POMDP navigation in a real world laboratory map (2914 states; $\sim$28m-by-8.8m). The black circle is the robot. Gray and black cells are free space and obstacles, respectively. Blue and white cells visually depict a single belief point; their opacity is a log-probability of the robot's location. Blue highlights the top $k=3$ probability masses in the belief. For *planning*, the σ-approximation uses the fixed top k weights for each belief.

To our knowledge, this is a new form of *belief compression* for POMDPs, with theoretical guarantees in conjunction with PBVI. A similar method was briefly suggested for the separate Bayes-Adaptive POMDP model [12]. They did not, however, provide any theoretical or empirical analysis, nor the general algorithm presented here. Value directed belief state compression [13] performs intelligent state space compression to only discard (mostly) irrelevant parts of the belief state, yielding the smallest invariant Krylov subspace. They use a distinct linear lossy compression method that approximates the original POMDP. Exponential family Principle Components Analysis (E-PCA) has also been used to compress beliefs into a low-dimensional belief space [14]. They instead solve the compressed POMDP, then map the policy back to the original POMDP. This operates over all beliefs at once, whereas ours operates on individual beliefs. Both compression methods and their numerous variants differ markedly from our fixed non-zero values, sort-based algorithm.

The σ-approximation exploits *sparse beliefs*. While few algorithms leverage this fact, such as sparse stochastic finite state controllers [15], it has been suggested as a measure of POMDP complexity [16]. Other related work includes Algebraic Decision Diagrams (ADDs) used to solve large factored POMDPs, and approximate belief points in the process [17], albeit in a very different manner from our approach.

Our paper begins with a review of the POMDP model (Section 2), followed by our σ-approximation algorithm (Section 3). Additionally, we present two main propositions

This work was supported in part by NSF (grant IIS-1405550). Any opinions, findings, and conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

College of Information and Computer Sciences, University of Massachusetts, Amherst, MA 01002, USA. Emails: {wray,shlomo}@cs.umass.edu

(correctness and an error bound) as well as two supporting lemmas. Then, we present experiments on standard benchmark domains, and a full robot implementation for navigation and localization, that demonstrate our approximation vastly improves performance with minor error (Section 4). We conclude with a discussion of our approach and potential future work (Section 5).

II. Background

A partially observable Markov decision process (POMDP) is represented by a tuple $\langle S, A, \Omega, T, O, R \rangle$. S is a set of n states, A is a set of m actions, and Ω is a set of z observations. $T : S \times A \times S \to [0,1]$ is a state transition function mapping a state s and action a to a successor state s' with probability $T(s,a,s') \equiv Pr(s'|s,a)$. It is common in practice, however, to define T with a successor function that returns only the non-zero valued successor states and their probabilities. Let the maximum number of possible successor states be denoted as $n_s \leqslant n$. $O : A \times S \times \Omega \to [0,1]$ is an observation function that stochastically emits an observation ω given action a led to state s' with probability $O(a,s',\omega) \equiv Pr(\omega|a,s')$. $R : S \times A \to \mathbb{R}$ is a reward function, denoted $R(s,a)$ for state s and action a.

The agent does not necessarily know the true state of the POMDP at any given time. Instead noisy observations are made and the agent is able to maintain a *belief* over the true state. We denote a set of r beliefs as $B \subseteq \triangle^n$, with $\triangle^n$ denoting the standard n-simplex. The agent updates a current belief $b \in B$ after taking an action a and making an observation ω to a new belief b' for a state $s \in S$ following:

$$b'(s'|b,a,\omega) = \eta O(a,s',\omega) \sum_{s \in S} T(s,a,s') b(s) \qquad (1)$$

with normalization constant $\eta = Pr(\omega|b,a)^{-1}$. Importantly, let r_z denote the maximum number of non-zero values over all belief vectors $b \in B$.

Agents operate for a number of discrete time steps called the *horizon* $h \in \mathbb{N}$. The agent's reward is reduced by a *discount factor* $\gamma \in (0,1)$ per time step. Infinite horizon ($h = \infty$) POMDPs can often be approximated by some finite horizon. A *policy* $\pi : B \to A$ describes how the agent acts based on its beliefs. We also define the *value function* $V : B \to \mathbb{R}$ as the expected reward at each belief, which is piecewise linear and convex in this space [18]. This fact enables us to represent the value function as a collection of α-*vectors* $\Gamma = \{\alpha_1, \ldots, \alpha_x\}$ with each $\alpha_i = [V(s_1), \ldots, V(s_n)]^T$ and $V(s_j)$ denoting the value of state s_j. We record a policy by marking an action with each α-vector, so we have the compact notation: $V(b) = \alpha \cdot b$ and $\pi(b) = a_\alpha \in A$.

A. Point-Based Solution Methods

Point-based value iteration (PBVI) [5] and other belief point-based approaches, such as heuristic search value iteration (HSVI2) [6] and Perseus [7], do not expand all reachable beliefs from an initial seed belief. Instead, they operate on a different set (e.g., a subset) to avoid the exponential growth of reachable beliefs over the horizon. In PBVI, we have an initial *expand* step (denoted as $expand(\cdot)$ in Algorithm 1) which produces a set of beliefs $B \subseteq \triangle^n$. Then, we apply value iteration over these beliefs, producing α-vectors at each time step t denoted as Γ^t. Formally, this procedure is applied h times (denoted as $update(\cdot)$ in Algorithm 1), given Γ^{t-1}, to produce Γ^t, is given by:

$$\Gamma^t_{a\omega} = \{[V^t_{s_1 a\omega\alpha}, \ldots, V^t_{s_n a\omega\alpha}]^T, \forall \alpha \in \Gamma^{t-1}\}, \quad \forall a \in A, \omega \in \Omega$$

$$\Gamma^t_b = \{r_a + \sum_{\omega \in \Omega} \operatorname*{argmax}_{\alpha \in \Gamma^t_{a\omega}} \alpha \cdot b, \forall a \in A\}, \quad \forall b \in B$$

$$\Gamma^t = \{\operatorname*{argmax}_{\alpha \in \Gamma^t_b} \alpha \cdot b, \forall b \in B\}$$

with variables $V^t_{sa\omega\alpha} = \gamma \sum_{s' \in S} O(a,s',\omega) T(s,a,s') \alpha(s')$, $r_a = \sum_{s \in S} b(s) R(s,a)$, and initial α-vectors be $\alpha(s) = \underline{R}/(1-\gamma)$, for all $s \in S$, with $\underline{R} = \min_{s \in S} \min_{a \in A} R(s,a)$ guaranteeing α-vectors increase [18].

III. The σ-Approximation Method

Our inspiration comes from the realization that: (1) belief dot products are nested throughout PBVI and other algorithms, (2) zero-multiplied values may be skipped, (3) a similar definition of n_s for beliefs might be exploitable, and (4) there is a significant performance improvement in practice when $r_z \ll n$ as opposed to $r_z \approx n$. With these insights, we designed a variant that can be applied to *any* belief-based algorithm that reduces the beliefs from an expand step to be of size $\hat{r}_z \leqslant r_z$ for use within an update step. For the sake of clarity, we focus here on PBVI applications only; however, the algorithm can be easily applied to commonly used value iteration (VI) methods such as HSVI2 or Perseus in a natural way. We call this general algorithm the *σ-approximation*. For brevity, we denote the use of our algorithm on any point-based algorithm with the prefix 'σ' (e.g., σ-PBVI, σ-HSVI2, σ-Perseus, etc.). The σ denotes the measure of approximation, a value that can be computed, with a guarantee that $\sigma \in [1/n, 1]$.

The algorithm separates the true set of beliefs used in the expand step B from the (approximate) set used in the update step $\hat{B}$. Importantly, each expand step continues to use the true beliefs B. Since our method removes non-zero beliefs, which are small in belief vectors, if we used $\hat{B}$ for expansions, then algorithms that explore *reachable* beliefs might never explore the full set of reachable beliefs. By preserving B for expand, we are able to explore the full set of reachable beliefs, and then approximate these with a bounded size of non-zero values for beliefs in $\hat{B}$ for updates. *Thus, how should we best approximate beliefs in B given the $\hat{r}_z$ constraint*?

A. Optimal Selection in the σ-Approximation

Let $b \in B$ be any belief point from the expanded set of beliefs B. Let $N = \{1, \ldots, n\}$. Assume we are given a constraint $\hat{r}_z \leqslant r_z \in N$ that denotes the desired maximum number of non-zero belief point values in any belief. Let $\hat{B}$

Algorithm 1 The σ-Approximation Method for basic PBVI.

Require: $\langle S, A, \Omega, T, O, R\rangle$: The POMDP.
Require: $\hat{r}_z$: The desired maximum number of non-zero values.
Require: b^0: The initial belief.
1: $B \leftarrow expand(b^0)$
2: $\hat{B} \leftarrow \varnothing$
3: **for** $b \in B$ **do**
4: $\quad \hat{b} = [0, \ldots, 0]^T$
5: $\quad$ **for** $i \in \{1, \ldots, n\}$ **do**
6: $\quad\quad o \leftarrow sort(b_i)$
7: $\quad\quad \hat{I} \leftarrow \{i \in N | o_r(i) \leqslant \hat{r}_z\}$
8: $\quad\quad \hat{b}_i \leftarrow \begin{cases} \frac{b_i}{\sigma_b}, & \text{if } i \in \hat{I} \\ 0, & \text{otherwise} \end{cases}$
9: $\quad$ **end for**
10: $\quad \hat{B} \leftarrow \hat{B} \cup \{\hat{b}\}$
11: **end for**
12: $\Gamma \leftarrow update(\hat{B})$

denote the approximated beliefs of B given the $\hat{r}_z$ constraint. Formally, this constraint guarantees that for $\hat{b} \in \hat{B}$:

$$|\{i \in N | \hat{b}_i > 0\}| \leqslant \hat{r}_z \tag{2}$$

The σ-approximation operates in the following manner. For all beliefs $b \in B$, $b = [b_1, \ldots, b_n]^T$. We sort the belief's values in $O(n \log n)$ time (denoted $sort(\cdot)$ in Algorithm 1). Optionally, this is much faster if: (1) we cleverly expand so the beliefs are already sorted, and/or (2) if we sparsely store beliefs. Let $o_r : N \rightarrow N$ denote the resulting *descending* ordering (rank index) of the belief vector's indices after sorting. Let $\hat{I} = \{i \in N | o_r(i) \leqslant \hat{r}_z\}$ be the reduced set of indices of only the top $\hat{r}_z$ with respect to their probabilities. We define the new approximate belief $\hat{b}$, to be added to $\hat{B}$, of the original b, for $i \in N$ as:

$$\hat{b}_i = \begin{cases} \frac{b_i}{\sigma_b}, & \text{if } i \in \hat{I} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

with $\sigma_b = \sum_{i \in \hat{I}} b_i$. This also ensures Equation 2 holds. We let $\sigma = \min_{b \in B} \sigma_b$ denote the overall worst-case approximation error using our method. Interestingly, the definition of $\hat{I}$ implies that the worst-case approximation error is bounded to an interval $\sigma \in [1/n, 1]$. This only arises with $\hat{r}_z = 1$ and a uniform belief b. The procedure is shown in Algorithm 1.

B. Theoretical Analysis of the σ-Approximation

First, we prove in Proposition 1 that the σ-approximation algorithm yielding $\hat{b}$ from Equation 3 returns the correct optimal approximate belief given the fixed $\hat{r}_z$.

Proposition 1 (Correctness): For belief $b \in \triangle^n$ and $\hat{r}_z \in N$, for all other beliefs $b' \in \triangle^n$ with the same $\hat{r}_z$ constraint: $|\{k \in N | b'_k > 0\}| \leqslant \hat{r}_z$, we have the property that $\hat{b} \in \triangle^n$ produced by the σ-approximation:

$$\|\hat{b} - b\|_1 \leqslant \|b' - b\|_1 \tag{4}$$

Proof: Assume by contradiction there exists a $b' \in \triangle^n$ with the $\hat{r}_z$ constraint (Equation 2) such that $\|\hat{b} - b\|_1 > \|b' - b\|_1$. Let $K' = \{k \in N | b'_k > 0\}$. By definition of 1-norm we have:

$$\sum_{i \in \hat{I}} |\hat{b}_i - b_i| + \sum_{i \notin \hat{I}} |b_i| > \sum_{k \in K'} |b'_k - b_k| + \sum_{k \notin K'} |b_k|$$

By rearranging and the definition of $\hat{b}$ in Equation 3:

$$\sum_{i \in \hat{I}} \Big| \frac{b_i}{\sigma_b} - b_i \Big| + \sum_{k \in K'} |b'_k - b_k| > \sum_{k \notin K'} |b_k| - \sum_{i \notin \hat{I}} |b_i|$$

By Equation 2, $\hat{I} = \{i \in N | o_r(i) \leqslant \hat{r}_z\}$, which by the descending ordering o_r, we guarantee $\hat{b}$ selected the largest $\hat{r}_z$ values from b. Thus, $\forall X \subseteq N$ such that $|X| \leqslant \hat{r}_z$, $\sum_{i \in \hat{I}} b_i \geqslant \sum_{x \in X} b_x$. By rearranging and applying probability normalization requirement: $\sum_{i \notin \hat{I}} b_i \leqslant \sum_{x \notin X} b_x$. With this fact and properties of absolute values, we obtain:

$$\Big| \frac{1}{\sigma_b} - 1 \Big| \sum_{i \in \hat{I}} b_i - \sum_{k \in K'} |b'_k - b_k| > 0$$

By the definition of σ_b, rearranging, and subadditivity:

$$\Big| \frac{1}{\sigma_b} - 1 \Big| \sigma_b > \sum_{k \in K'} |b'_k - b_k| \geqslant \Big| \sum_{k \in K'} b'_k - b_k \Big|$$

By definition of b' and that probabilities sum to 1:

$$\Big| \frac{1}{\sigma_b} - 1 \Big| \sigma_b > \Big| 1 - \sum_{k \in K'} b_k \Big| = 1 - \sum_{k \in K'} b_k$$

Rearrange, apply the definitions of $\hat{I}$, K', and σ_b, as well as the properties of absolute values with $\sigma_b \in (0, 1]$ to obtain:

$$1 < \Big| \frac{1 - \sigma_b}{\sigma_b} \Big| \sigma_b + \sum_{k \in K'} b_k \leqslant \Big| \frac{1 - \sigma_b}{\sigma_b} \Big| \sigma_b + \sum_{k \in \hat{I}} b_k = \frac{1 - \sigma_b}{\sigma_b} \sigma_b + \sigma_b$$

This implies that $1 < 1 - \sigma_b + \sigma_b = 1$, hence a contradiction is reached. Therefore, $\hat{b}$ is optimal following Equation 4. ■

Next, we would like to know how much error (in terms of value at a belief) this approximation adds to PBVI and the other point-based methods. First, Lemma 1 provides an upper bound on the distance from any approximate belief $\hat{b} \in \hat{B}$ to an arbitrary belief $b' \in \triangle^n$. Importantly, this bound is only in terms of the corresponding $b \in B$ for which $\hat{b}$ was an approximation and σ_b.

Lemma 1: For any belief $b' \in \triangle^n$, and belief $\hat{b} \in \triangle^n$ produced by the σ-approximation of belief $b \in B$, we have:

$$\|b' - \hat{b}\|_1 \leqslant \|b' - b\|_1 + 2(1 - \sigma_b) \tag{5}$$

Proof: Take any belief $b' \in \triangle^n$ and σ-approximate belief $\hat{b} \in \triangle^n$ for belief $b \in B$. We apply the triangle inequality (using b_i), the definition of $\hat{b}$ (Equation 3), rearrange, apply the definition of σ_b, and simplify.

$$\begin{aligned} \|b' - \hat{b}\|_1 &= \sum_{i=1}^{n} |b'_i - \hat{b}_i| \leqslant \sum_{i=1}^{n} |b'_i - b_i| + \sum_{i=1}^{n} |b_i - \hat{b}_i| \\ &= \|b' - b\|_1 + \sum_{i \in \hat{I}} \Big| b_i - \frac{b_i}{\sigma_b} \Big| + \sum_{i \notin \hat{I}} |b_i| \\ &= \|b' - b\|_1 + \Big| 1 - \frac{1}{\sigma_b} \Big| \sum_{i \in \hat{I}} |b_i| + (1 - \sigma_b) \\ &= \|b' - b\|_1 + \frac{1 - \sigma_b}{\sigma_b} \sigma_b + (1 - \sigma_b) \end{aligned}$$

which implies $\|b' - \hat{b}\|_1 \leqslant \|b' - b\|_1 + 2(1-\sigma_b)$. ■

We use this result in Lemma 2 and Proposition 2, which proves a bound on σ-PBVI's value error in terms of the density of the *original* belief points $\delta_B = \max_{b' \in \triangle^n} \min_{b \in B} \|b - b'\|_1$ [5] and the worst-case approximation error σ. The bound also utilizes $\overline{R} = \max_{s,a} R(s,a)$ and $\underline{R} = \min_{s,a} R(s,a)$. Importantly, this proof extends the original by Pineau *et al.* [5] and contains components of it.

Lemma 2 (σ-PBVI One Step Error Bound): The error ϵ introduced in σ-PBVI when performing one iteration of value backup over $\hat{B}$ instead of B or $\triangle^n$, is bounded by:

$$\epsilon \leqslant \frac{\overline{R} - \underline{R}}{1-\gamma}(\delta_B + 2(1-\sigma)) \tag{6}$$

Proof: We start with the belief $b' \in \triangle^n$ that had the largest error after a σ-PBVI update, and the closest $\hat{b} \in \hat{B}$ (which σ-approximates belief $b \in B$) to b' via a 1-norm, with maximal α-vector α' for b' and would be maximal α-vector $\hat{\alpha}$ at $\hat{b}$.

$$\begin{aligned}
\epsilon &\leqslant \alpha' b' - \hat{\alpha} b' \leqslant \|\alpha' - \hat{\alpha}\|_\infty \|b' - \hat{b}\|_1 && \text{By Pineau } \textit{et al.} \\
&\leqslant \|\alpha' - \hat{\alpha}\|_\infty (\|b' - b\|_1 + 2(1-\sigma_b)) && \text{By Lemma 1} \\
&\leqslant \frac{\overline{R} - \underline{R}}{1-\gamma}(\delta_B + 2(1-\sigma_b)) && \text{By Pineau } \textit{et al.} \\
&\leqslant \frac{\overline{R} - \underline{R}}{1-\gamma}(\delta_B + 2(1-\sigma)) && \text{By } \sigma = \min_{b \in B} \sigma_b
\end{aligned}$$

■

Proposition 2 (σ-PBVI Error Bound): For any set of beliefs $B \subseteq \triangle^n$, σ-approximation $\hat{B}$ of B, and horizon t, the error of the σ-PBVI algorithm $\epsilon_t = \|V_t^{\hat{B}} - V_t^*\|_\infty$ is bounded by:

$$\epsilon_t \leqslant \frac{\overline{R} - \underline{R}}{(1-\gamma)^2}(\delta_B + 2(1-\sigma)) \tag{7}$$

with $V_t^{\hat{B}}$ and V_t^* denoting the estimate and optimal value functions, respectively.

Proof: Again by Pineau *et al.* we have the error ϵ_t at time t bounded as:

$$\begin{aligned}
\epsilon_t &\leqslant \|\tilde{H} V_{t-1}^{\hat{B}} - H V_{t-1}^{\hat{B}}\|_\infty + \gamma e_{t-1} && \text{By Pineau } \textit{et al.} \\
&\leqslant \frac{\overline{R} - \underline{R}}{1-\gamma}(\delta_B + 2(1-\sigma)) + \gamma e_{t-1} && \text{By Lemma 2} \\
&\leqslant \frac{\overline{R} - \underline{R}}{(1-\gamma)^2}(\delta_B + 2(1-\sigma)) && \text{By geometric series}
\end{aligned}$$

with $\tilde{H}$ and H above above denoting the PBVI and exact update operators, respectively. Note that σ-PBVI has the same value update operator just on a different belief set. ■

An interesting facet of this bound is the relation between δ_B and $2(1-\sigma)$. Since beliefs are probabilities, $\delta_B \in [0,2]$. Similarly, $\sigma \in [1/n, 1]$ implies the other term is on the same range $2(1-\sigma) \in [0, 2(n-1)/n] \to [0,2]$ as $n \to \infty$. We call this term the *σ-error*. Both also measure an approximation and are orthogonal considerations. In other words, one could have dense beliefs with high σ-error (σ-VI), sparse beliefs with low σ-error (PBVI), sparse beliefs and high σ-error (σ-PBVI), or dense beliefs and low σ-error (VI).

The best-case scenario that will yield the largest performance gains using our σ-approximation consists of domains in which beliefs are almost all collapsed to a few states, but have a lot of very small spread out beliefs over other states. The σ-approximation will then replace these beliefs and efficiently perform updates on most of the denser parts of the belief vector's space.

The theoretical complexity of our PBVI's update equation is $O(n^2 mzr^2)$ in the worst case with $n_s = \hat{r}_z = r_z = n$. In comparison, the σ-approximation has a reduced complexity of $O(mzrn(n + r\hat{r}_z))$ in the worst case with $n_s = n$. Note that the absolute worst-case cost of sorting, $O(rn \log n)$, is greatly overshadowed by the update cost. Additionally, this reduces memory requirements. PBVI requires $O(rn)$ space to store all belief points, whereas σ-PBVI requires $O(r\hat{r}_z)$. While this may not seem like much for smaller problems, larger problems can have beliefs that are spread out over many states. Thus, we can approximate large belief vectors with the σ-approximation, while maintaining the original size of smaller ones. This largely preserves the accuracy of PBVI with a minor modification that vastly improves overall runtime performance, especially if $\hat{r}_z \approx \sqrt{n}$ or $\hat{r}_z \approx \log n$. This observation is empirically supported by our experiments, described in the next section.

Furthermore, parallel implementations of PBVI (multi-core CPU, GPU, or cluster) eliminate the major bottleneck: number of belief points r [19], [20]. With $n_s \ll n$, one of the remaining major bottleneck variable becomes r_z, which a parallelized σ-PBVI addresses. Finally, communication overhead is one of the biggest factors for parallel algorithms, particularly on clusters. σ-PBVI enables belief points to be transferred over a network on a cluster much faster because of its tunable bounded memory size $O(r\hat{r}_z)$.

IV. Experimentation

We begin with a comparison of σ-PBVI over six standard POMDP benchmark domains, varying the levels of the approximation. Then, we experiment with σ-approximation on a real robot performing simultaneous navigation and localization.

A. Performance of σ-Approximation on Benchmarks

We implement σ-PBVI to investigate its performance improvements and solution quality. Table I shows the results over six larger well-known domains using ranges of $\hat{r}_z$ values. In particular, we compute the base r_z without our σ-approximation, then vary $\hat{r}_z$ to be r_z, $r_z/3$, $r_z/10$, and $r_z/30$. Importantly, this version of PBVI is already much more efficient than a naive implementation that stores all n probabilities for each belief point, even with $\hat{r}_z = r_z$.

Aloha-30, Hallway2, and Tiger Grid all obtain over an order of magnitude improvement. Even the largest domain, Rock Sample (7x8), results in over three times improvement with almost zero error in value $V(b_0)$. Results can be further

Domain							PBVI			σ-PBVI								
							$\hat{r}_z{=}r_z$			$\hat{r}_z{=}\lceil r_z/3 \rceil$			$\hat{r}_z{=}\lceil r_z/10 \rceil$			$\hat{r}_z{=}\lceil r_z/30 \rceil$		
Name	n	m	z	r	n_s	r_z	T	$V(b_0)$	σ	T	$V(b_0)$	σ	T	$V(b_0)$	σ	T	$V(b_0)$	σ
Aloha-10	30	9	3	64	25	10	1.3	106.0	1.0	0.6	105.8	0.64	0.3	101.1	0.36	0.18	98.3	0.18
Aloha-30	90	29	3	128	27	30	82.0	787.4	1.0	34.4	787.3	0.83	13.5	784.5	0.38	7.6	769.1	0.19
Fourth	1052	4	28	256	3	1052	186.4	-60.5	1.0	187.3	-60.5	1.00	183.4	-60.5	1.00	87.3	-60.5	1.00
Hallway2	92	5	17	128	88	88	80.6	0.28	1.0	25.4	0.26	0.34	7.9	0.23	0.10	3.3	0.16	0.03
Rock Sam.	12545	13	2	512	1	256	142.0	-147.1	1.0	71.9	-148.0	0.34	50.2	-145.3	0.10	42.7	-146.9	0.04
Tag	870	5	30	256	5	841	158.7	-25.8	1.0	131.4	-27.7	0.33	131.9	-30.6	0.10	118.8	-30.2	0.03
Tiger Grid	36	5	17	64	5	36	5.04	-0.79	1.0	2.32	-1.06	0.99	0.85	-1.09	0.78	0.48	-1.11	0.69

TABLE I

COMPUTATION TIME T (IN SECONDS) FOR $h{=}50$, INITIAL BELIEF'S VALUE $V(b^0)$, AND σ AVERAGED OVER 10 TRIALS FOR EACH DOMAIN.

improved by the user, in terms of time or quality, using the *tunable* parameter $\hat{r}_z$.

Overall, there is a clear trend that larger domains benefit more from this than smaller domains. This is due in part to large spread out belief vectors being relatively rare after expand steps; most reachable beliefs in large domains are actually dense with a few near-zero belief values. Thus, these introduce very small overall error when approximated with smaller belief vectors. Additionally, more complex expand steps (e.g., PEMA) might improve the standard PBVI beliefs, but recall that we are still σ-approximating those beliefs. Thus, the σ-approximation result will also further improve. In summary, our σ-approximation worked well in large domains, introducing low error for greatly reduced computation time.

B. POMDP Navigation and Localization on a Real Robot

We construct a real robotic navigation and localization experiment similar to those found in the few previous real applications of POMDPs [21], [22], [23]. Here, we define a 56 state POMDP: an 8-by-7 abstracted grid. There are 9 actions: all 8 neighboring cells and a stop action. Furthermore, there are 2 observations: "bump" or "no bump". Note that this results in the POMDP's actions and observations allowing for both navigation and localization. The probability of successful forward motion is 0.9, with a slight uniform chance of deviating left and right, as well as not moving. The probability of observing a "bump" is proportional to the average number of obstacles over all possible successor states. The reward is a small 0.05 for non-goal states and 0.0 for the goal. Belief is therefore over the location of the robot as it moves around the world. We assign the initial beliefs to be collapsed with 1.0 probability mass over each state and perform original PBVI expansions afterward selecting maximally "distinct" beliefs [5]. The σ-approximation is applied on these beliefs.

Figure 2 shows the real world execution of σ-PBVI ($k{=}4$) and PBVI in a maze on a robot platform: the base Kobuki made by Yujin Robot Co., Ltd. with an Nvidia Jetson TX1 made by Nvidia Corporation. As we observe, the actual real-world performance is quite similar. The maze itself was designed to spread belief over the straight "hallways" prior to entering each "room". In practice, the belief spreads out over much more than $k{=}4$ states; however, as observed, the final performance is quite similar.

V. CONCLUSION

We provide an approximation algorithm that compresses the non-zero values in belief vectors, solving larger problems faster with bounded additional error. We provide two propositions, and two related lemmas, proving that our σ-approximation is optimal and has bounded error. This is demonstrated in our experiments on six standard domains. Additionally, we implement a POMDP on a real robot in a simultaneous navigation and localization domain, comparing σ-PBVI and PBVI, showing only minor policy differences.

The main contribution of the σ-approximation its applicability to *all algorithms that operates over beliefs*. We envision its use in many other algorithms beyond σ-PBVI, including σ-HSVI2 and σ-Perseus. Also, the σ-approximation is much simpler to implement over other approaches, such as value directed belief state compression [13] or E-PCA methods [14]. We plan to explore broader use of σ-approximation in future work with this foundation established. Finally, we will provide our source code so that others could easily build faster approximate POMDP solvers.

Acknowledgments: We thank Dirk Ruiken and Samer Nashed for their help with our robot in the experiments.

REFERENCES

[1] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1, pp. 99–134, 1998.

[2] A. Castelletti, F. Pianosi, and R. Soncini-Sessa, "Water reservoir control under economic, social and environmental constraints," *Automatica*, vol. 44, no. 6, pp. 1595–1607, 2008.

[3] K. H. Wray and S. Zilberstein, "Multi-objective POMDPs with lexicographic reward preferences," in *Proceedings of the 24th International Joint Conference of Artificial Intelligence (IJCAI)*, July 2015, pp. 1719–1725.

[4] K. H. Wray, S. J. Witwicki, and S. Zilberstein, "Online decision-making for scalable autonomous systems," in *Proceedings of the 26th International Joint Conference of Artificial Intelligence (IJCAI)*, August 2017.

[5] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for POMDPs," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 3, 2003, pp. 1025–1032.

[6] T. Smith and R. Simmons, "Heuristic search value iteration for POMDPs," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2004, pp. 520–527.

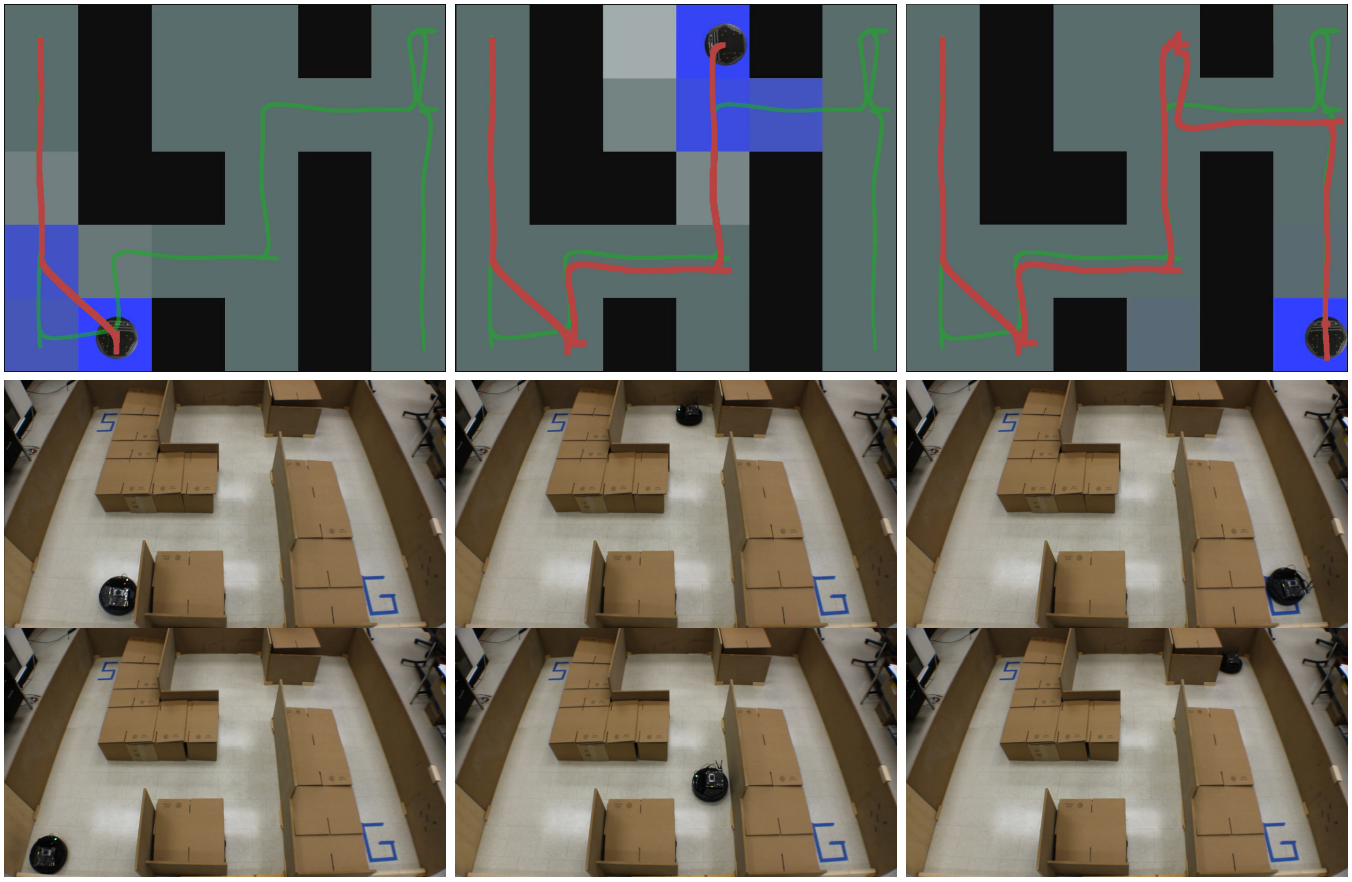

Fig. 2. Demonstration of our σ-approximation used on a real robot. Each column of images denotes the ROS output (top) and corresponding real world pictures for σ-PBVI (middle) and normal PBVI (bottom) over time (left to right). The black circle is the robot. Blue and white denote log-probability belief regarding the robot's physical location. Blue visually highlights only the top three highest weights for reference. The red line denotes the σ-PBVI path. The green line denotes the normal PBVI path. (Both paths are from *odometry*.) The start and goal are marked as "S" and "G", respectively. Note the localization attempts in the paths in which the robot intentionally "bumps" the wall to confirm its location and collapse belief.

[7] M. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[8] J. Pineau, G. Gordon, and S. Thrun, "Anytime point-based approximations for large POMDPs," *Journal of Artificial Intelligence Research*, vol. 27, pp. 335–380, 2006.

[9] G. Shani, R. I. Brafman, and S. E. Shimony, "Forward search value iteration for POMDPs," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007, pp. 2619–2624.

[10] P. Poupart, K. Kim, and D. Kim, "Closing the gap: Improved bounds on optimal POMDP solutions," in *Proceedings of the 21st International Conference on Automated Planning and Scheduling (ICAPS)*, 2011, pp. 194–201.

[11] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems*, vol. 27, no. 1, pp. 1–51, 2013.

[12] S. Ross, B. Chaib-draa, and J. Pineau, "Bayes-adaptive POMDPs," in *Proceedings of Advances in Neural Information Processing Systems 20 (NIPS)*, 2008, pp. 1225–1232.

[13] P. Poupart and C. Boutilier, "Value-directed compression of POMDPs," in *Proceedings of Advances in Neural Information Processing Systems 15 (NIPS)*, 2003, pp. 1579–1586.

[14] N. Roy, G. J. Gordon, and S. Thrun, "Finding approximate POMDP solutions through belief compression," *Journal of Artificial Intelligence Research (JAIR)*, vol. 23, pp. 1–40, 2005.

[15] E. Hansen, "Sparse stochastic finite-state controllers for POMDPs," in *Proceedings of the 24th Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008, pp. 256–263.

[16] W. S. Lee, N. Rong, and D. J. Hsu, "What makes some POMDP problems easy to approximate?" in *Proceedings of Advances in Neural Information Processing Systems 20 (NIPS)*, 2008, pp. 689–696.

[17] G. Shani, P. Poupart, R. I. Brafman, and S. E. Shimony, "Efficient ADD operations for point-based algorithms," in *Proceedings of the 18th International Conference on Automated Planning and Scheduling (ICAPS)*, 2008, pp. 330–337.

[18] W. S. Lovejoy, "Computationally feasible bounds for partially observed Markov decision processes," *Operations Research*, vol. 39, no. 1, pp. 162–175, 1991.

[19] G. Shani, "Evaluating point-based POMDP solvers on multicore machines," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 4, pp. 1062–1074, 2010.

[20] K. H. Wray and S. Zilberstein, "A parallel point-based POMDP algorithm leveraging GPUs," in *AAAI Fall Symposium on Sequential Decision Making for Intelligent Agents (SDMIA)*, November 2015, pp. 95–96.

[21] A. Brooks, A. Makarenko, S. Williams, and H. Durrant-Whyte, "Parametric POMDPs for planning in continuous state spaces," *Robotics and Autonomous Systems*, vol. 54, no. 11, pp. 887–897, 2006.

[22] M. T. Spaan and N. Vlassis, "A point-based POMDP algorithm for robot planning," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, 2004, pp. 2399–2404.

[23] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 271–281, 2003.