Partially Observable Multi-Target Tracking For the mWidar System

Ryan Draves
University of Colorado Boulder
Smead Aerospace Engineering Sciences
Boulder, CO

Maggie Wussow
University of Colorado Boulder
Smead Aerospace Engineering Sciences
Boulder, CO

***Index Terms*—Multi-Target Tracking, POMDP, POMCP, POMCPOW**

***Abstract*—This project explored the optimization of the configuration over time of a novel microwave imaging system (mWidar) to adaptively adjust the visible area given target positions. This project experimented with using a Partially Observable Decision Process (POMDP) to plan over reconfigurations of the mWidar system to observe targets with unknown positions in the environment. Continuous and discretized formulations of the underlying dynamics model were implemented, with heuristic and POMCP policies used on the discretized model and POMCPOW used on the continuous model. The results demonstrate how POMDP formulations can be used to plan over information-gathering actions in multi-target tracking scenarios.**

I. Introduction

The mWidar sensor[1] is a novel microwave imaging system. It's environment reconstruction allows for the detection of objects (such as people) in an environment through walls and other microwave-permable materials, framing its use an advanced sensor. Its aperture can be adusted by changing the configuration of its transducers, allowing for different modes for its field-of-view and imaging resolution. This creates a decision-making problem around when it may be desirable to reconfigure the sensor.

In a multi-target tracking scenario, this decision-making problem is defined by finding the set of configurations that allows for the best estimates of the most targets in an environment. "Best" and "most" are competing goals, leading to a tradeoff of exploring the environment with different aperture settings and exploiting the current configuration to capture a better estimate of the targets in the field-of-view.

II. Related Work

mWidar is a new technology for target detection and tracking; however, previous methods of task scheduling for target tracking can be applied to this problem. Specifically, task scheduling for phased arrays has been a relevant problem for target detection as the phased array requires the adjustment of its antennae to change the radar's parameters such as transmission strength and direction. Throughout the mission, these antennae have to be changed to different configurations to maximize the usefulness of the overall array so it can point and observe the object it is required to at the correct time. In Li et al.[2], the authors propose a task scheduling algorithm which is based on a three-way decision. This algorithm makes decisions on which configuration to be in based on the three actions which correspond to different areas. Their algorithm looks at scheduling tasks temporally and maximizing reward by completing the highest priority task. For the mWidar the field-of-view (FOV) of the radar is based on its current geometry. This allows the geometric configurations to be also formulated into three main actions which correspond to a specific observation area.

In Miller et al.[3], POMDPs are used in a multi-target tracking scenario where a centralized planner coordinates a swarm of UAVs to collect measurements and optimize the track quality of the targets in the environment. While our environment more closely resembles the single UAV case, the goal of optimizing track estimates remains the same. The authors formulate a novel approximation method to solve their constructed POMDP, creating Nominal Belief Optimization (NBO) to optimize a policy on a simplified belief propation. Our work differs in that we examine using online POMDP solvers to find policies, rather than approximate methods.

III. Problem Formulation

For this project the environment was modeled as a Partially Observable Markov Decision Process (POMDP) in which the mWidar is tasked with tracking targets in a defined area. Each target's position is only observed if that target has entered the FOV of the mWidar system's current configuration. The problem formulation has been separated into three parts; an underlying dynamics environment that represents the "physical world" an mWidar system may plan over and two POMDP models (discrete and continuous) that simplify the underlying environment into one tractable for online planners.

A. Environment Dynamics

The underlying environment is modeled by the independent dynamics of each target and the current configuration of the mWidar. The environment area is a $60m \times 120m$ continuous 2D space with the mWidar was located at (30,30). The environment is setup to have a set of *N* targets that spawn randomly in a uniform distribution across the perimeter of the environment area. The spawn times of the targets are also assigned randomly from a uniform distribution of the simulation duration. When each target spawns, they are given a random constant speed (v) between $[1, 2]$ m/s and a random heading (ψ) facing into the arena (with respect to its initial placement on the arena perimeter). The dynamics of the targets are given in Equation 1 which are used to propagate them through the environment. Once they exit the environment space laid out above, the targets can no longer be observed.

$$\begin{aligned} \dot{x} &= v\cos(\psi)\Delta t \\ \dot{y} &= v\sin(\psi)\Delta t \end{aligned} \tag{1}$$

The mWidar system in this problem was defined to be configurable into three states: line, arc, and circle. This also defines the action spaces, which has a deterministic transition. These configurations each create a different field of view (FOV) of the mWidar and allow it to observe different parts of the environment, shown in fig. 1. The line has the longest field of view being able to see much further into the environment but also has the smallest area it can observe at 1800 m^2. The circle allows the mWdiar to observe behind it and around it with the greatest area of the configurations at 2827.4 m^2. The arc configuration is a median between the two with a much wider observation field of view than the line but a further observation distance than the circle and an observable area in between the two at 2000 m^2.

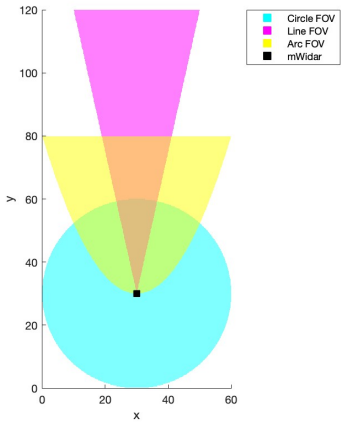

Fig. 1. mWidar Configurations FOV

Observations in this environment are given by a simple model; any target within the FOV of the current sensor configuration have their positions perfectly observed in continuous space. As a simplification of the model, perfect data association is also provided; i.e. the ID of a target is known for each observation. Apart from the data association, this is an approximation of the information that could be extracted from the mWidar's image outputs.

The rewards for each action are based on the true state of the environment. There is a small negative reward to penalize changing the configuration of the system as that takes power. There is a positive reward for each target observed for every time step and a larger reward for seeing new targets. This incentivizes exploration instead of continual exploitation. The reward for seeing new targets is high because the mWidar system gains a larger amount of information viewing a new target and decreasing its positional uncertainty compared with observing a target it previously saw and therefore has higher positional certainty about. This culminates in eqn. 2 which shows the complete reward function.

$$R(s,a) = \begin{cases} +1 & \forall \text{ target } t \in \text{FOV} \\ +2 & \forall \text{ target } t \notin \text{prev\_observations} \\ -0.25 & \text{if } a \neq \text{configState} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Lastly, a discount factor of 0.95 was configured.

This environment simulates a more complex physical environment than might be desired to solve with a POMDP. While the observation model is simplified, the state space consists of each target's position, heading,

speed, and ID. The next sections detail how this simulated environment is simplified to create tractable state spaces for POMDP solvers.

B. Discretized POMDP

Our discretized POMDP $(S, T, A, R, O, Z, \gamma)$ is defined as follows:

The environment is discretized in $1m^2$ grid cells. Rather than estimating the speed and heading of each target, only the position is captured in this state space. This makes the combined state space given by the current mWidar configuration and each target's position, or $\{1, 3\} \times (\{1, 40\} \times \{1, 120\})^N$, where N is the (known a priori) number of targets that will appear in the environment. It is worth noting two additional flags for each target: if it $has\_spawned$ or if its position is $nothing$, indicating that the target has left the arena.

Since the true state is updated by the underlying environment, our "physical world," the discretized POMDP transition probability distribution approximates the underlying environment dynamics so a belief planner can predict the discretized state into future time steps. The mWidar configuration is deterministically updated given the action, but the targets instead adopt a more complicated approximation. Since only the positions are captured in the state space, a 10% probability of spawning randomly along the perimeter is allotted to targets that have not yet spawned. Targets that have spawned are modeled as talking a random walk across neighboring grid cells (up/down/left/right).

The action space matches the underlying environment (the mWidar may be configured into either the Line, Arc, or Circle states).

The reward function, similar to the transition distribution, is an approximate model used by the belief planner. We model the same small cost (0.25) for changing the mWidar configuration and a reward (1.0) for each target that is observed, however the reward for observing new targets is not captured, as it's not in the discretized state space.

The observation space is given by the same discretization as the state space; i.e. $(\{1, 40\} \times \{1, 120\})^N$. Similar to the state space, there's the additional state for each target to denote if it was not observed.

The observation probability distribution matches the underlying environment, except for being discretized. If a target is within the current FOV, its discretized position and ID are observed.

Lastly, the discount factor matches the underlying environment and is set to 0.95.

C. Continuous POMDP

The continuous POMDP $(S, T, A, R, O, Z, \gamma)$ was formulated as follows:

The state space captured the same information as the discreteized POMDP, except for the target positions being continuous. This defines the state space as $\{1, 3\} \times ([1, 40] \times [1, 120])^N$, with the addition of the same two modes for each target.

The transition probabilities are constructed akin to the discrete case and similarly used by a belief planner. Targets are modeled to perform a random walk in the continuous space using a Gaussian distribution with a mean at the estimated target position and a covariance matrix of $1.5I$ which accounts for the targets moving at speeds of 1-2 meters per second. Target spawning was modeled with the same 10% chance as the discretized transition probabilities, albeit with spawning continuously along the perimeter, and the mWidar configured is similarly modeled to be deterministic.

The action space matches the underlying environment (the mWidar may be configured into either the Line, Arc, or Circle states).

The reward function is the same as the discrete case, except that the FOV check is continuous instead of discretized.

The observation space and observation probability distribution match the discretized POMDP. This discretization was used to get POMCPOW to bin similar states by associating states that map to the same discrete observation.

Lastly, the discount factor matches the underlying environment and is set to 0.95.

IV. Solution Approach

A. Particle Filtering

In order to run POMDP planners, a belief must first be constructed over the state space of our POMDP models (discrete and continuous). Since the state spaces are so large, we used particle filters to track our beliefs and model the dynamics through time. Importantly, since the targets propagate independently, the state space can be accordingly factorized and an independent particle filter for each target used.

Per-target particle filters for both the discrete and continuous models were used. The particle filter logic was provided by the ParticleFilters.jl package and we implemented the corresponding dynamics, likelihood, and post-processing methods for it. The dynamics and likelihood methods were computed per-target using the

same transition probabilities as described in III for each model. The post-processing method re-sampled particles when they all decayed. In the event where none of the particles explain the observation from the underlying environment, we re-sampled all particles from the observation.

For both models, we constructed per-target particle filters with 10,000 particles.

B. Heuristic Approaches

To establish a baseline, we implemented two simple heuristics: pick a random action and always pick the Line configuration.

For a more advanced heuristic, we implemented a greedy strategy that takes the discrete particle filter beliefs and computes an expected reward from each particle and its corresponding weight. This expected reward only considered target observation rewards and did not model the cost of changing configurations or the reward of seeing new targets.

C. POMCP

For the discrete POMDP model, we used the online POMDP solver POMCP[4] implemented by BasicPOMCP.jl. We parameterized the solver with $max\_depth = 10$, $tree\_queries = 1000$, and $c = 5.0$. For the estimate value, we implemented a fully observable rollout policy that greedily selected an action given the state, similar to the heuristic policy.

An important consideration for the POMCP planner is the construction of the belief to pass into the planner. Since the discretized POMDP is defined by the combined states of the targets and mWidar configuration, we must combine the per-target particle filters and construct a weighted particle belief over the combined state space. Since we had $10,000$ particles and $N = 20$ targets, our full combined space belief could be all combinations of particles, or $10,000^{20}$. To make this tractable, we simply squashed the particles for each together to get $10,000$ combined state belief "particles." This means that every combined "belief particle" consisted of a single particle from each of the target particle sets combined.

D. POMCPOW

For the continuous POMDP model, we used the online POMDP solver POMCPOW[5] implemented by POMCPOW.jl. This was parameterized by $max\_depth = 10$, $tree\_queries = 1000$, $k\_observations = 5.0$ (widening parameter), and $alpha\_observation = 0.5$ (widening exponent). An analogous continuous-space greedy rollout policy was used.

The same method of squashing the per-target particle filters was used to construct a belief over the combined continuous state space for a total of $10,000$ belief "particles."

V. Results

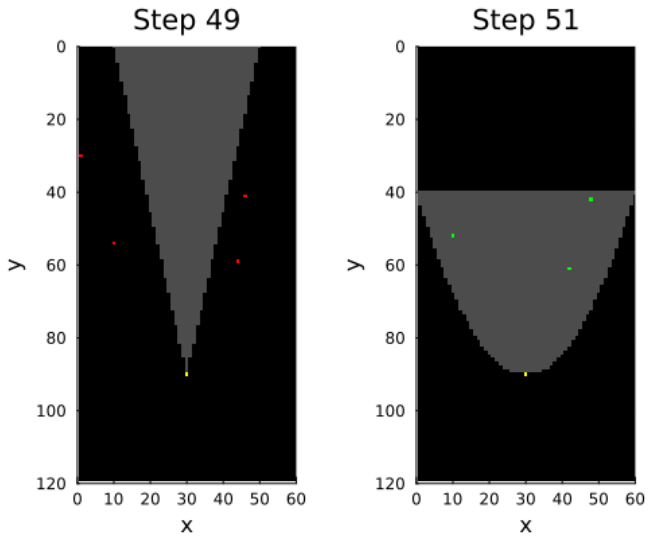

Fig. 2. POMCP exploring the Arc configuration as targets leave the observability of the Line configuration.

An environment was constructed with 20 targets to balance the amount of targets so they were not too sparse while also ensuring that there were not too many targets which would incentivize static behavior with exploitation. The five implemented policies were run to compare which policy maximized the reward and thus best balanced exploration and exploitation. The cumulative rewards are shown in Figure 3 and Table I.

Policy	Line	Random	Heuristic	POMCP	POMCPOW
	10.6	35.0	68.0	61.5	63.7

TABLE I
TOTAL REWARDS OF EACH POLICY

The simple heuristic policy which always returns the action Line performs the worst by far, performing under one-third of the next best policy. This is due to it's static nature not allowing it to explore and catch targets outside its FOV and pulling in few new targets. The second worst policy is the simple heuristic random policy where actions are drawn from a uniform random distribution. This has the opposite problem where its dynamic behavior means it often loses reward due to changing configuration. But its dynamic changes allow

it to catch more new targets gaining that bonus reward per new target.

The POMCP formulation outperforms the simple heuristic policies. Since it was fed a tractably sized belief, POMCP was able to compute a tree of decisions and discretized beliefs and find a reasonably performant best action. Figure 2 shows POMCP's chosen configuration over two frames. As targets leave the observability of the Line configuration, it computed a stronger belief over states that would benefit the Arc configuration and chose it.

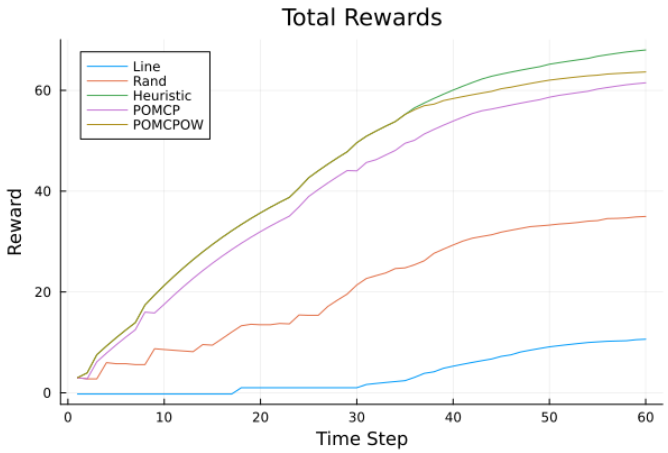

Fig. 3. Cumulative rewards over the simulation across each tested policy.

The second best performing policy is the POMCPOW policy, doing better than POMCP but worse than the advanced heuristic. We theorize that planning on a continuous state space with an according variation in its transition probabilities created better beliefs that POMCPOW was able to exploit.

The policy that had the highest reward was the advanced heuristic. Its plans greedily but the main difference between it and the other planning algorithms is how it deals with the belief estimate. We suspect that computing an expected reward over each weighted particle was more valuable in greedy planning than the combined beliefs were in short-term horizon planning.

An interesting note is that POMCP and POMCPOW policies perform similarly even though the POMCP optimizes over the entire discritized state space while the POMCPOW progressively widens the tree it optimizes over in the continuous space. This observation leads to a hypothesis that the discretizations are sized so there are no issues with precision which would affect a policy's decision. Each target has a $1m^2$ uncertainty within that discretized space where it could be located, but it doesn't significantly change the model or behavior from the true environment dynamics. This is because the imprecision of the target's location compared to its true location is only relevant with positions along the edges of the FOV. This means that the majority of the time for the majority of the targets, they spend no more than a second or two in a state where the precision matters. Thus the loss of slight precision of the positions of the targets does not significantly impact the behavior of the policies.

VI. Conclusion

Our results show that online POMDP planners can be used to select observation actions in multi-target tracking scenarios. We were unable to show that such policies could outperform a greedy heuristic policy, suggesting a range of possible issues, from reward shaping, model dynamics approximations, to the efficacy of the solvers in this problem domain.

We suspect, with insubstantial evidence, that the model approximations benefited a greedy strategy over the online planner. One of the largest issues with these models for the POMDP solvers was the formulation of the belief state since the belief modeling was extremely uncertain. Until the mWidar views the target it has a probability distribution that incudes targets spawning along the perimeter, targets outside the bounds of the environment, and that the targets could be inside the environment but outside the FOV of the mWidar. The belief state could be made more complex to include a temporally based probability of the targets spawning as time goes on. Refining the belief modeling to take into account more of the dynamics and constraints of the system would be a logical next step to improve the quality of the planner policies.

VII. Contributions & Release

Ryan developed the simulated target dynamics, discretized model, discretized belief updates, and heuristic policies.

Maggie developed the reward function, mWidar observation dynamics, continuous model, continuous belief updates, and plotting.

The authors grant permission for this report to be posted publicly.

Acknowledgment

This work was completed in parallel with another group (Anthony La Barca and Mark Boyer) who looked at the same system approached as an MDP. Some of the plotting functions were shared as well as resources on

the mWidar system itself. The dynamics, models, and methods were developed independently.

REFERENCES

[1] F. C. S. Da Silva, A. B. Kos, G. E. Antonucci, J. B. Coder, C. W. Nelson, and A. Hati, "Continuous-capture microwave imaging," *Nature Communications*, vol. 12, no. 1, p. 3981, Jun. 2021.

[2] B. Li, L. Tian, D. Chen, and Y. Han, "A task scheduling algorithm for phased-array radar based on dynamic three-way decision," *Sensors*, vol. 20, no. 1, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/153

[3] S. A. Miller, Z. A. Harris, and E. K. Chong, "A POMDP Framework for Coordinated Guidance of Autonomous UAVs for Multitarget Tracking," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, no. 1, p. 724597, Dec. 2009.

[4] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010.

[5] Z. Sunberg and M. J. Kochenderfer, "POMCPOW: an online algorithm for pomdps with continuous state, action, and observation spaces," *CoRR*, vol. abs/1709.06196, 2017. [Online]. Available: http://arxiv.org/abs/1709.06196