Multi-Robot Expansive Planning and Trajectory Evaluation for Tracking and Localization of Marine Life

Kehlani A. Fay
Pomona College
Claremont, California
knfa2018@mymail.pomona.edu

Michael Giordano
Harvey Mudd College
Claremont, California
mgiordano@hmc.edu

Alberto Soto
Harvey Mudd College
Claremont, California
alsoto@hmc.edu

Christopher M. Clark
Harvey Mudd College
Claremont, California
clark@hmc.edu

Emily Spurgeon
California State University Long Beach
Long Beach, California
emily.spurgeon@student.csulb.edu

James Anderson
California State University Long Beach
Long Beach, California
james.anderson@csulb.edu

Christopher G. Lowe
California State University Long Beach
Long Beach, California
chris.lowe@csulb.edu

ABSTRACT

Traditional techniques for marine life tracking use stationary receivers that detect and obtain measurements from tagged animals. Recently, such static systems have been replaced by multiple mobile robots, e.g., autonomous underwater vehicles (AUVs), equipped with omni-directional hydrophones that can accurately localize marine life. In this paper, the application of homogeneous multi-AUV systems to track and localize marine life is used as a motivating example to develop new MRMP (Multi-Robot Motion Planning) algorithms. These algorithms generate trajectories that maximize a new fitness function that incorporates 1) probabilistic motion models generated from historical data of live sharks, and 2) ideal AUV formations for observing a shark from multiple sensor vantage points. The two expansive RRT variants, named Independent State Expansion *(ISE)* planning and Joint State Expansion *(JSE)* planning, differ in how new samples are randomly generated during the algorithm's random search. The fitness function was developed to quantify how accurately the positioning of AUVs would trilaterate the target animal. Through simulation, it was found that the Joint planner was 70% faster with respect to run time than Independent planner, while both could produce similar mean fitness function values. The fitness for these variants was also measured for simulations where different target motion models were used when calculating the fitness function, highlighting the improved performance when using actual target motion motion models.

CCS CONCEPTS

• **Computer systems organization** → **Robotic autonomy**; **Robotic control**; **Sensor networks**;

KEYWORDS

Multi-robot Systems, Autonomous Robots, Cooperative Robots, Decentralized Algorithms

ACM Reference Format:
Kehlani A. Fay, Michael Giordano, Alberto Soto, Christopher M. Clark, Emily Spurgeon, James Anderson, and Christopher G. Lowe. 2023. Multi-Robot Expansive Planning and Trajectory Evaluation for Tracking and Localization of Marine Life. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27-April 2, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3555776.3577749

SAC '23, March 27-April 2, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/10.1145/3555776.3577749

1 INTRODUCTION

Prior work in the subclass of motion planning for cooperative, dynamic target tracking often utilizes a variety of probabilistic models but fails to utilize historical behavioral data to inform probabilistic or predictive motion plans [2] [17] [25]. To the best of our knowledge, this is the first work which combines multi robot motion planning, probabilistic motion models, and historical behavioral models to generate motion plans. The result is a motion plan which incorporates predictive animal motion and responds according to observations. This work utilizes predictive marine life tracking, specifically white shark tracking, as a motivating example. However, the approach taken is agnostic of the creatures tracked and can be applied to other tracking targets by updating the likelihood function to the collected target data, desired distance function, and robot dynamics. The MRMP algorithms presented use an extension of Rapidly Exploring Random Tree (RRT) to take advantage of the ability to rapidly explore high degree of freedom configuration spaces and incorporate nonlinear system constraints.

Tracking marine life provides essential behavioral and ecological data for understanding aquatic animals, evaluating marine species' health, and discovering anthropomorphic affects on marine life. A wide variety of tracking techniques exist to accomplish marine life data collection including satellite transmitters, stationary receivers, and manual boat based tracking. However, each of these techniques are limited by the requirement of satellite transmitters

to surface [28], fixed localization area and accuracy [10], or human intensive labor [8]. More recent studies have demonstrated the use of hydrophones on AUVs to successfully track tagged marine life to address these issues [8][6][13]. AUVs also enable tracking in larger areas, increase tracking accuracy, and improve temporal resolution [13]. These studies have experimentally validated the use of AUVs in marine life tracking, but with limited work on optimized AUV formations for localization.

Prior work on multi-AUV marine life tracking includes the development of state estimation algorithms for localizing tagged sharks [13]. However, this previous work does not address the issue that shark state estimate representations are multi-modal probability distributions and lacks considering such representations into motion planning. Related work using acoustic beacon localization for a group of AUVs uses an information theoretic approach but does not account for nonlinear movements of the tracked AUVs or use predictive behavioral models [23]. This work builds on the prior work by creating MRMP algorithms that construct trajectories which are optimal with respect to the expected shark motion generated from probability distributions that represent real shark motion behaviors.

The main contributions of this work include:

- A novel multi-AUV RRT motion planner that generates kinematically feasible trajectories for tracking a common target. The planner has two variants and an expansion modification that increases planner success and trajectory fitness.
- A fitness function for multi-AUV trajectory optimization.
- New conditional probability models for shark behavior that can be queried by the fitness function.
- Simulation results that highlight algorithm performance.

The paper is organized as follows. Section II discusses prior work on motion models, robot target tracking, and localization. Section III introduces the problem formulation. Section IV describes the new shark motion models. Section V describes the MRMP algorithm. Section VI describes simulation results.

2 BACKGROUND

Acoustic localization is the process of finding an object's position in space relative to the surrounding environment through sound. For shark tracking, acoustic tags have been used to relay animal motion information to nearby receivers. Hydrophone arrays mounted to AUVs have been used as these receivers to successfully track and gather population estimates on a variety of marine life [7][11]. When an acoustic tag emits a signal, positions can be estimated by the relative receiving time between each hydrophone. The degree of localization accuracy is partially determined by distance from the object and, in the case of multiple AUVs, is also determined by the relative headings and relative distance. However, uncertainty can be sourced from ambiguous source angle due to symmetry in time of flight. Prior work in single robot and multi robot shark tracking has shown estimation, communication, and reducing this uncertainty is feasible with multiple robots [8][13].

Active target tracking (ATT) involves dynamically following a target with a robotic agent. In the case of multi-agent ATT, robots need to avoid both obstacles and other robots. Current multi-robot ATT techniques include, but are not limited to, inter-agent repulsion, elliptical orbits, leader follower approaches, and Policy Gradients [16][4][3]. A large body of work has explored optimal robot configurations for ideal sensor measurements and localization along with uncertainty in target dynamics [17] [22] [24]. In the case of robot soccer where agents triangulate positions around a moving ball, a Nonlinear Model Predictive Controller used comprehensive modelling of all system components and agents to predict the future state and optimize robot to target formations while minimizing uncertainty [19]. This decentralized approach couples each robot's position through its cost function in order to maintain a desired geometry, but exclusively uses a nonlinear predictive controller in order to avoid using a path planner.

While the algorithms used for robot tracking typically do not include actual animal motion models, related work has been conducted in developing such models. Position based animal motion models typically fall into three main categories: Hidden Markov Models (HMMs), State Space Models (SSMs), and Diffusion Processes [21]. Prior work in ecological research have used HMMs when state dynamics are unobservable and sampling frequency is constant [18]. In marine life research, HMMs have successfully been applied to label behavioral phases of bluefin tuna [20] and tag recovery on white sharks [9]. When applying these models to hydrophone sensors, a probabilistic model of the shark's position and heading is used. This accounts for the dual symmetrical localization of hydrophone measurements and a low sampling rate of once per minute. The shark's behavioral model is predicted and updated as either stationary, constant velocity, or a Hidden Markov Model (HMM).

Multi Robot Motion Planning (MRMP) remains a fundamental NP-hard problem in robotics. Approaches to MRMP aim to create dynamically feasible and collision free trajectories for any given number of robots and paired start to goal locations. Challenges within solving MRMP problems include scalability, robustness, uncertainty, communication, and optimization of multi robot paths for tasks [27]. MRMP problems have typically been sub divided in control strategies of centralized and decentralized. Centralized approaches use full state information of all robots for more optimal solutions [14], but are often difficult to deploy in field environments with limited communication and struggle with scalability. Decentralized approaches trade guarantees on optimal paths for scalability and relax reliance on communication. Due to the difficulty of underwater communication [12], this paper utilizes a decentralized approach.

Prior work in decentralized Multi Robot Motion Planning (MRMP) utilize a variety of path evaluation and path planning techniques with a large focus in search based, deep learning, reduction, and rule based algorithms [15][30]. Search based approaches have extended traditional path planning algorithms with leading algorithms including variants of ORCA [26]. While search based approaches have shown large promise, ongoing improvements in run time, scalability, and task relevant optimization remains ongoing.

3 PROBLEM FORMULATION

Given a set of A identical AUVs at initial states $X_{i,0}$ for $i = 0...A-1$, each equipped equipped with inter-AUV communication and a

sensor that provides position measurements $Z_{i,t}$ of a target being tracked, the goal is to generate AUV trajectories that enable the AUVs to autonomously track and follow the target in open water.

For each AUV i, a state at each time step t is defined as $X_{i,t} = [x_{i,t}, y_{i,t}, \theta_{i,t}, v_{i,t}, \omega_{i,t}]$ where (x, y) corresponds to the 2D vectored position of robot i at time t, $\theta_{i,t}$ is robot i's angular heading relative to true North, and v,ω correspond to the linear and angular velocity respectively. An AUV i's trajectory, is a sequence of states: $T_i = [X_{i,0}, ..., X_{i,t_{max}}]$.

To represent the target's state, $X_{s,t} = [x_{s,t}, y_{s,t}, \theta_{s,t}]$ at time t and selected grid cell s with 2D (x,y) coordinates (i,j), a probability distribution, $P(X_{s,t})$, is used and updated based on received hydrophone measurements. The state of the target can be defined as $X_{s,t} = [x_{s,t}, y_{s,t}, \theta_{s,t}]$ at time t with 2D position ($x_{s,t}$,$y_{s,t}$) and heading θ. However, the belief state of the target is represented as the probability distribution $P(X_{s,t})$ which is updated based on transition probabilities and, in the case of closed loop feedback, incoming $X_{s,t}$ measurements. $P(X_{s,t})$ is defined as a target relative body frame matrix representing the likelihood of the target's state for each occupancy map grid cell. These trajectories are then to be optimized to:

- Maximize the likelihood the target stays within a circular formation created by the AUVs, thereby enabling multiple sensor vantage points for increased tracking accuracy.
- Minimize the likelihood AUVs come within some threshold distance of the target to prevent interference with target motion behavior.
- Minimize the likelihood AUVs leave some threshold distance of the target to maintain reliable hydrophone readings of the shark's acoustic tag.

The optimization of these trajectories is enabled by finding a trajectory set $\mathbf{T} = [T_0 ... T_{A-1}]$ that maximizes the following fitness function:

$$F(\mathbf{T}) = \sum_{t=0}^{t_{max}} \frac{\rho t}{t_{max}} f(T_t) \tag{1}$$

$$f(T_t) = \sum_{i,j \in P} \sum_{u,v \in R} p_{i,j,t} \; C(\theta_{i,j,u,v}) \; min\left(D(q_{i,j,u}),\; D(q_{i,j,v})\right) \tag{2}$$

This fitness function in Eq. (1) sums the expected fitness over all times across a horizon of t_{max} seconds. The ratio $\frac{\rho t}{t_{max}}$ linearly biases the function to fitness gained at the end of trajectory where ρ is k constant. In Eq. (2), the expected fitness for time step t includes a sum over all cells i, j in a probability matrix P, and over all AUV pairs in the trajectory set R. The term to the right of the summations includes a product of the likelihood $p_{i,j,t}$ of the shark being located at cell i, j at time step t, a AUV pair angle function $C()$, and an AUV pair distance function $D()$. $\theta_{i,j,u,v}$ represents the relative heading between two selected AUVs, u and v in the set of all robots R, with the selected grid cell (i, j).

The distance function $D()$ enables the AUVs to maintain the target within sensor range while maintaining distance sufficient not to disturb the target. To note, $q_{i,j,u}$ is defined as the 2D euclidean distance between AUV u and cell (i, j). In Eq. (3), h is the desired upper limit between an AUV and target marine life (e.g., 50 meters), and $\gamma = 0.05$ is a smoothing factor. Fig. 1 illustrates the distance function.

$$D(l) = \left(\frac{1}{1 + e^{(l-h)/(\gamma h)}}\right) + \left(\frac{1}{1 + e^{((h/5)-l)/(\gamma h)}}\right) - 1 \tag{3}$$

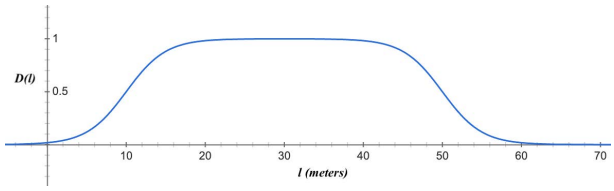

Figure 1: Graph of $D(l)$ from 0m to 70m.

The angle function $C()$ rewards the AUVs for spreading out around the target to provide multiple sensor vantage points. First, the heading angle $\beta_{r,i,j}$ from the state $X_{i,j}$ of cell (i, j) in the probability map P position to each AUV is calculated. Second, the differences $\Delta\beta_{i,j}(r, s) = \beta_{r,i,j} - \beta_{s,i,j}$ between these heading angles for each AUV pair (s, r) are calculated. Third, difference errors can be generated, (in the case where AUVs perfectly surround the target), as $\theta_{i,j}(r, s) = |\beta_{i,j}(r, s) - 2\pi/A|$. Finally, the input to $C()$ is this error averaged over all AUVs pairs to yield $\bar{\theta}_{i,j,t}$ at time t for cell (i, j).

$$C(\bar{\theta}_{i,j,t}) = \left(\frac{1}{1 + \bar{\theta}_{i,j,t}}\right)^3 \tag{4}$$

4 SHARK MOTION MODELING

In this work, fitness is calculated as a function of the predicted state of a shark — which is represented as a 2D time-dependent probability distribution P, where the likelihood of being in cell (i, j) of a 2D spatial grid is $p_{i,j,t} \in [0, 1]$ at time t. Notably $\sum_{i,j} p_{i,j,t} = 1$ $\forall\, t$. The probabilities $p_{i,j,t}$ are generated by Markovian expansions, in which:

$$p_{i,j,t} = \sum_{k,l} p(i, j, t|k, l, t-1) p_{k,l,t-1} \tag{5}$$

In Eq. (5), the likelihood of a shark being located at cell(i, j) at time t is a sum of the likelihoods of being located at all other cells (k, l) at the previous time step $t - 1$, multiplied by the transition probability $p(i, j, t|k, l, t - 1)$.

To obtain transition probabilities $p(i, j, t|k, l, t - 1)$, 22 juvenile white sharks were acoustically tracked off the coast of southern California [1]. Sharks were fitted with depth sensing transmitters, allowing for 3D positioning of movements based on triangulation within the 5 km^2 receiver array. This yielded 30,000 data points. Normalizing the sharks headings, the 2D shark translations were extracted and modeled as transition probabilities. Due to the large majority of shark trajectories maintaining a depth between 0 to 1 meters, the algorithm variants are deployed in 2D but can be easily extended to 3D. Notably, the transitions are conditional on shark depth (Fig 2), and on shark velocity (Fig. 4).

When extracting the transition probabilities $p(i, j, t|k, l, t - 1)$ to calculate a shark location probability for any time time t, the planner can use the most recent shark state measurement $X_{s,0}$,

and then translate and rotate the transition probabilities (e.g. those shown in Figures 2 and 3) to be centered on the state $X_{s,0}$.

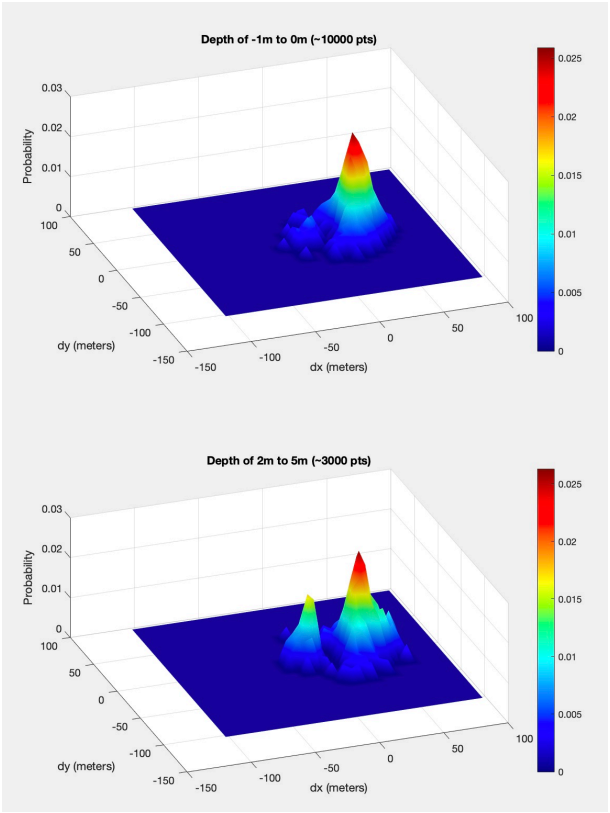

Figure 2: Probability distributions modeling shark transitions over 1 minute, starting at $(0, 0)$ and pointing in the positive x direction. The shark is at the surface in (a), and the sea floor in (b).

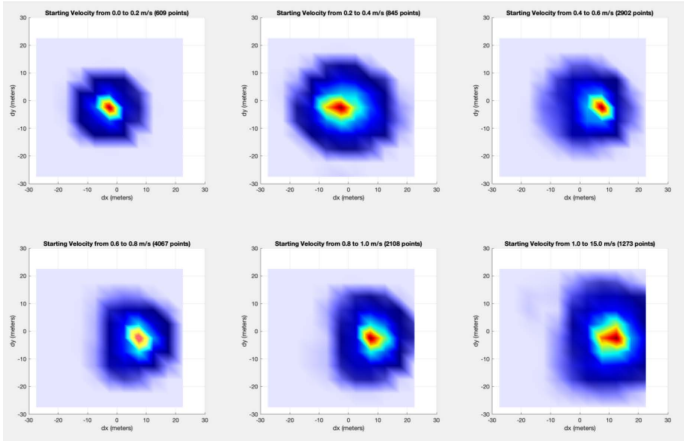

Figure 3: Probability distributions from shark data conditional on current velocity, i.e. each subimage corresponds to a different velocity range. The color legend is the same as in Figure 2.

5 MRMP ALGORITHMS

This section introduces two variants of a multi-AUV RRT motion planner that generates kinematically feasible, collision-free trajectories. Given an initial state $X_{i,0}$ for each AUV $i = 0...A-1$, and initial shark state estimate $X_{s,0}$, both planners create a set of A collision free trajectories, one trajectory for each AUV.

Algorithm 1 Expansive RRT Planner with A AUVs

1: $G \leftarrow InitializeTree(n_0)$

2: **while** planTime < planTimeBudget **do**
3: $n_{random} \leftarrow NodeSelection(G)$
4: $n_{new} \leftarrow NodeExpansion(n_{random})$

5: **if** $NoCollision(n_{random}, n_{new})$ and $HighFitness(n_{new})$ **then**
6: $G \leftarrow n_{new}$
7: **if** $InEndGame(n_{new})$ and $Fitness(n_{new}) > F^*$ **then**
8: $n^* \leftarrow n_{new}$
9: $F^* \leftarrow Fitness(n_{new})$
10: **end if**
11: **end if**
12: **end while**

13: **T** $\leftarrow TrajectoryFromNode(n^*)$
14: **return T**

As with traditional RRT algorithms, a tree G of nodes and edges is grown from an initial node n_0 through sampling that generates random edges to new nodes. The two Independent State Expansion and Joint State Expansive planner variants, named *ISE* and *JSE* hereafter, differ by the coordination strategies used during the RRT expansion step in which new nodes are created. For the ISE planner, the j^{th} node n_j is defined by the state of all A AUVs at some time step t, the fitness accrued following tree edges to the node, and the node's parent node in the tree, i.e. $n_j = [t, X_{0,t}, ..., X_{A-1,t}, F(n_j), n_{parent}]$. For the JSE planner, node n_j is defined by the AUV states, the fitness accrued, the parent node, and the 2D position X_C of the center C of the AUV cluster, i.e. $n_j = [t, X_{0,t}, ..., X_{A-1,t}, F(n_j), n_{parent}, X_C]$.

Both planners utilize five main steps: initialization, edge expansion, collision checking, fitness evaluation, and end-game checking. As shown in Algorithm 1, line 1, the planners *initialize* by creating a tree G with the single node n_0 that incorporates the initial states of all AUVs and becomes the root node of the tree.

On line 3 of Algorithm 1, a while loop is used to iteratively add more edges to the tree G, until the planners time budget has elapsed. For each iteration of the while loop, the tree G is first sampled for a node n_{random} to expand from. This node selection step is accomplished by randomly sampling a point in space, and then finding the closest node n_{random} in tree G. The closest node is defined here as the minimum distance from an existing node in tree G to the furthest robot's prior distance. By taking the minimum of the max change in robot position and orientation, this allows workload to be distributed more evenly.

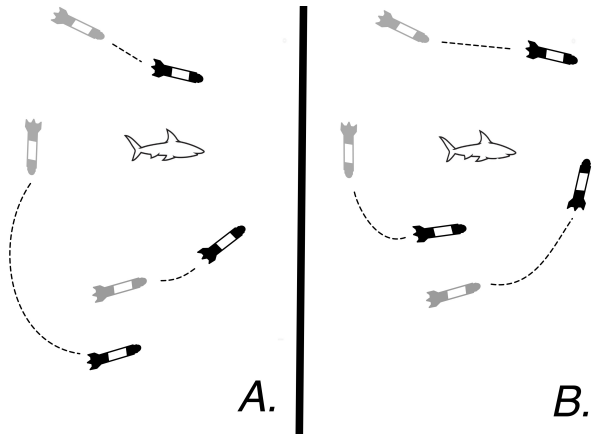

Figure 4: In this example, node B is closer than node A to the parent node's prior robot states. The prior robot states are opaque with the new generated node states in black.

Given the sampled node n_{random}, the *NodeExpansion* function on line 4 will generate a new node n_{new} to potentially be added to the tree. This new node must contain states of all A AUVs that are kinematically reachable from their parent states in n_{random}. These states are generated differently depending on whether the JSE vs. ISE planner is invoked.

Both planners start by initially generating a two component trajectory consisting of a constant velocity random arc of radius r and heading change $\Delta\theta_i$, and a constant velocity straight segment. This is accomplished by first randomly sampling a velocity $v_i \in [v_{min}, v_{max}]$, where v_{min} and v_{max} are dictated by the physical robot's capabilities. For a preset expansion time Δt, the expansion distance is calculated as $d = v_i \Delta t$. Subsequently, an expansion heading change is randomly sampled $\Delta\theta_i \in [-\Delta\theta_{max}, \Delta\theta_{max}]$. Assuming a circular arc motion, the radius of the arc for constant velocity tracking along the arc is $r = \frac{d}{\Delta\theta_i}$

The ISE planner applies the arc segment motion to each AUV i's state $X_{i,t-1}$ of n_{random}. In Eq. (6) and (7), the new state positions are a function the last state (term 1), arc geometry (terms 2,3), and straight line segment geometry (term 4).

$$\begin{aligned} x_{i,t} &= x_{i,t-1} - r\sin\theta_{i,t-1} + r\sin(\theta_{i,t-1} + \Delta\theta) + v\Delta t\cos(\theta_{i,t-1} + \Delta\theta) \\ y_{i,t} &= y_{i,t-1} + r\cos\theta_{i,t-1} - r\cos(\theta_{i,t-1} + \Delta\theta) + v\Delta t\sin(\theta_{i,t-1} + \Delta\theta) \\ \theta_{i,t} &= \theta_{i,t-1} + \Delta\theta \end{aligned} \tag{6}$$

The JSE planner applies the arc segment motion to the center state $X_{c,t-1}$ of the cluster of AUVs of n_{random}, as opposed to $X_{i,t-1}$ above. The same update equations (6),(7),(8) can be used to update the center state c instead of individual AUV i.

Once the JSE planner generates the new cluster center state $X_{c,t}$, each AUV's individual state in the cluster can be calculated. First, the new radius of the cluster is sampled $\rho_t \in [\rho_{min}, \rho_{max}]$ such that $\delta\rho_t$ does not exceed the max velocity. Then, for AUV i, the new node's updated state position is calculated via:

$$\begin{aligned} x_{i,t} &= x_{c,t} + \rho_t cos\beta_i \\ y_{i,t} &= y_{c,t} + \rho_t sin\beta_i \\ \beta_i &= \frac{2i\pi}{A} \end{aligned} \tag{7}$$

To ensure AUVs do not require velocities greater than v_{max} when charged with following the outside turning arc in cluster formation, all AUV velocities are scaled such that the fastest AUV will not exceed v_{max}. Additionally, JSE allows at each expansion for the radius from X_C to each robot to expand by +/- .5 meters and the variation from each robot from it's prior state to increase or decrease by +/- .5 meters. Final distance values are scaled for velocity constraints to ensure reachability within v_{max}. This allows for JSE formation to expand and contract around obstacles.

After each expansion, a collision checker is queried to determine if the edge from n_{random} to n_{new} is collision free. The planner leverages an iterative collision checker [5] that checks the kinematic trajectory for each AUV from the parent node state to the new node state, for collision with static obstacles. The collision checker expands on the prior through checking for each robot in the potential added node, no robot crosses another robot's path for all nodes in the tree at timestep t.

On line 5, the algorithm also checks if new node n_{new} has sufficient fitness value to be added to the tree. Note the function in Eq. 2 is used for this evaluation. This process strongly biases the tree to generating trajectories of high fitness (see results section below.

As shown on line 7, the algorithm checks if the new node belongs to the end game region, and if the trajectory from node n_0 to n_{new} has a higher fitness function than previously discovered. If so, the optimal end node n^* is set and new best fitness value recorded (lines 8 and 9). To note, a node is determined to belong to the end game region if its time stamp is greater than the preset trajectory time horizon t_{max}.

In order to deploy this algorithm in a decentralized manner, each AUV begins with the same random seed to ensure random node generation is the same across all AUVs. AUVs will communicate their updated 2D positions prior to each trajectory replan using either acoustic beacons with state estimation, token passing as executed in prior work [29], or other communication methods. Exact communication protocols are beyond the scope of this paper.

6 EXPERIMENTS AND SIMULATION RESULTS

Simulation experiments were conducted with C# implementations of the algorithm running on a Dell XPS 15. All experiments utilize AUVs that adhere to kinematic constraints, including a maximum longitudinal velocity of 2.5 m/s and maximum angular velocity of $\pi/4$ radians/second. These constraints are representative of the target robot, an L3Harris Iver3 AUV. The ISE planner initializes all AUVs to be co-located with the shark, and the JSE planner initializes AUVs evenly distributed on a circle with initial radius r centered on the shark. The shark's initial position is assumed to be known and normalized to the center of the graph as this algorithm is designed to be used post search. Baseline results were simulated using the JSE planner with 4 AUVs under Markov Expansion with a shark speed of 0.75 m/s. Unless otherwise specified, high fitness expansions were used for baseline tests with 100 runs.

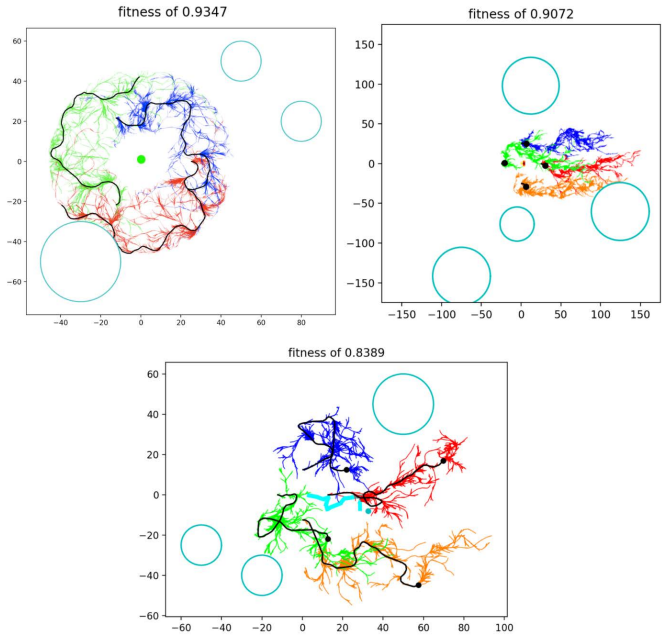

Figure 5: The ISE planner under predicted and observed stationary (top left), constant velocity (top right), and Markov models (bottom).

The planner was run in closed loop to highlight its adaptability for real-time replanning and response to target motion observations. For the closed loop simulations described below, the planner constructed new plans every 10 seconds, where each plan consisted of 30 second duration trajectory sets. This helps the planner consider future obstacles and adapt to observed shark movements. At the start of each plan cycle, the simulated shark state is updated by sampling from the probability matrix $P(X_{s,10})$ generated by the motion model in use. This shark state is provided to the planner as a measurement with zero uncertainty. The planner uses this updated state to seed the generation of probability map P, enabling fitness function calculations for potential trajectories. An example of a final ISE closed loop plan (Fig. 6) and final JSE closed loop plan (Fig. 7) are available for execution reference.

Both planner variants constructed higher fitness trajectories when employing the *high fitness* expansions than without. As shown in Figure 6, using a high fitness threshold value of 0.8 resulted in the ISE planner yielding a mean fitness of 0.78 as opposed to 0.03 without high fitness expansions. Similarly, the JSE Planner resulted in 0.82 with high fitness expansions and 0.01 without. Across variable shark speeds from 0 to 1 m/s, both planner average scores remained the same. Shark speeds above 1.5 m/s were unable to be accurately tested due to limited transition modeling data.

Both the ISE and JSE planners were tested against predicted and mispredicted belief models of the shark's motion. Under correctly matched models, the ISE planner resulted in a scores of 0.83, 0.83, 0.81 under stationary, constant velocity, and Markov models respectively. Likewise, the JSE planner resulted in an average score of 0.81, 0.84, 0.82, respectively. The minimum fitness value produced with the Markov model was greater than any other model's minimum

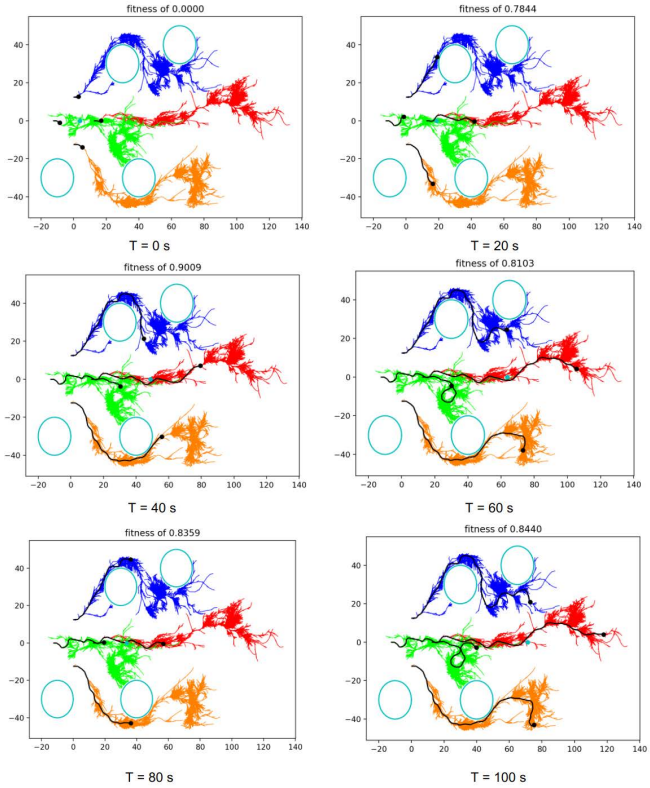

Figure 6: A 100 second trajectory plan generated by the ISE planner using 4 AUVs. The shark is both anticipated and moving by the at constant velocity of 0.75 m/s to the right. The highest probability of the shark location is represented as a blue dot with the AUV paths in black and obstacles represented with cyan circles.

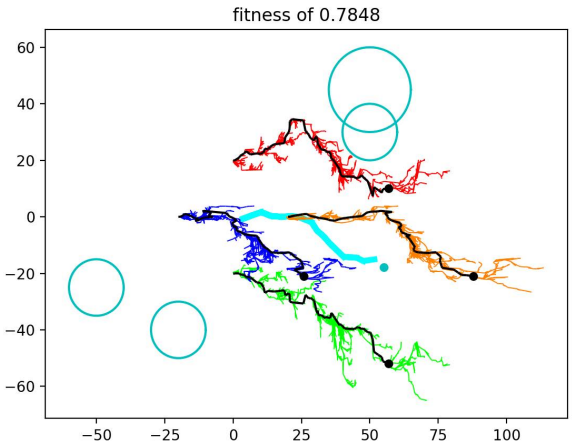

Figure 7: A snapshot at t=80 seconds for an executed JSE trajectory plan using a markov shark model and 4 AUVS. The observed shark path is in light blue with current shark position as a blue dot.

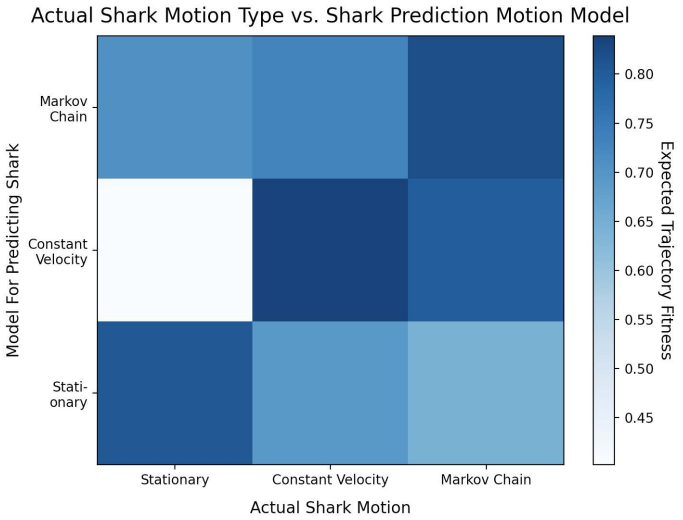

Figure 8: ISE planner Scores under varying Motion Models

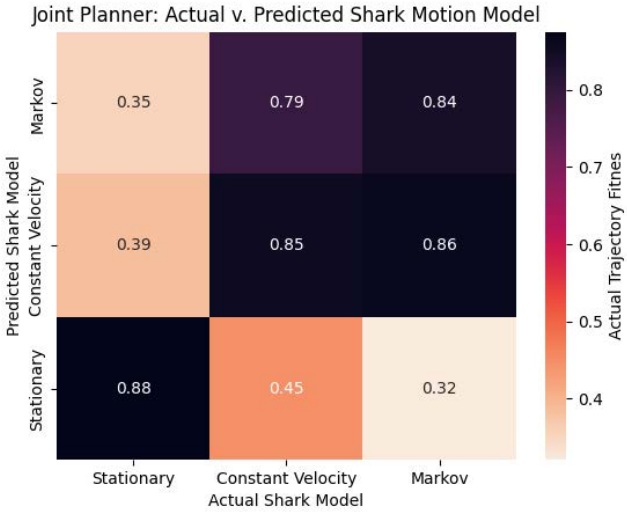

Figure 9: JSE planner Scores under varying Motion Models

fitness. The Markov model also predicted best against itself, despite not being deterministic. In summary, the Markov model not only better represents the actual motion of white sharks being tracked, but it proved robust to different models employed by the simulated sharks. An example of plans under varying shark motion models and actual shark motion models (Fig. 5) illustrates how the planner changes under varying models and behaviors. Score representations of mismatched and matched motion models show the ability of both planners to perform well under matched and mismatched model conditions (Fig 8. and Fig. 9).

With both planner variants, each additional AUV caused the overall fitness score to increase until a plateau in fitness was reached.

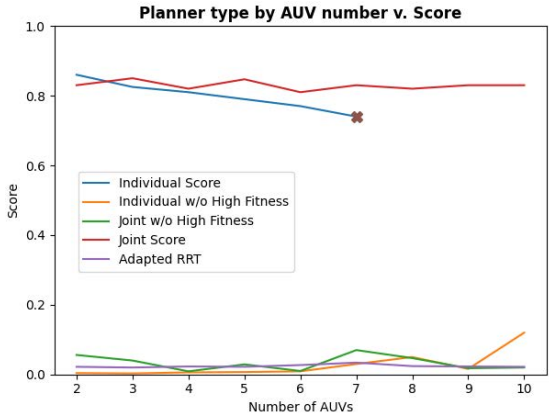

Figure 10: A score comparison to number of AUVs for ISE, JSE, and adapted baseline RRT planner.

Specifically, with each added AUV, the resulting increase in score had diminishing returns with little return occurring after 6 AUVs. The ISE Planner repeatedly timed out after 7 AUVs represented by the 'X' in Figure 10. Note the significantly improved fitness of both planners' trajectories when run using the high fitness expansion.

The main trade-off between the ISE Planner and the JSE planner is run time. Run time for the ISE Planner proved to increase exponentially as the number of robots increased while the JSE planner increased linearly. This is due to recomputing the trajectories for each robot in the ISE Planner while the JSE planner only computes the center trajectory and extrapolates robots positions from the updated point. This effectively removes the need for inter-robot collision-checking, and therefore leads to many more successful expansions. Between 2 and 4 robots, the ISE Planner is able to compute robot trajectories with higher fitness scores than the JSE planner, while avoiding timeout. Above this threshold, the ISE Planner frequently times out.

To provide a relative baseline for the ISE and JSE planners, a traditional RRT algorithm was adapted to work for multiple robots and evaluation functions. Namely, the traditional RRT planner has been adapted by changing a node to represent a collection of robots for score evaluation, where the minimum distance from a random configuration to the closest node is redefined as the minimum average of each robot's position to each robot's position in the random configuration state. Additionally, a max angular velocity and linear velocity of $\pi/2$ and 2.5 m/s respectively have been set to match constraints placed on ISE and JSE planners. High fitness was not implemented for this planner and resulted in an average score of .022 across variable robots.

7 CONCLUSION

This work presents two MRMP algorithms for multi robot target tracking using predictive models to optimize robot formations in simulation. When using correctly predicted motion models for fitness function calculation, the algorithm generated average fitness values of .78 and .82 out of the theoretical maximum of 1.0 for

ISE and JSE planner variants respectively. When applied to multi-AUV shark tracking, and incorporating a probabilistic model of the shark's position and future positions, robot formations are optimized for stable, long term, trajectory planning. Against a baseline RRT planner adapted for multiple robots and score evaluation, this planner scored an average of .022 out of 1.0. Both planners with high fitness outperformed this baseline.

In future work, one could deploy this simulated algorithm using multiple Iver3 AUVs. Additionally, the current algorithm assumes perfect communication and localization methods to accommodate communication loss could help AUVs recover from mismatched motion plans. The algorithms presented naturally extend to tracking swarms as the probability distributions can become products over multiple sharks and may be deployed in future work. Real world comparisons in performance between tracking multiple and single sharks with both ISE and JSE planners can be experimented with. The planning approach outlined in this paper may be applied to other multi agent active tracking problems including surveillance, wildlife tracking, and search and rescue.

ACKNOWLEDGMENTS

This material is supported by the National Science Foundation (NSF) under Grant No. 1952616. It does not represent the views or positions of the NSF.

REFERENCES

[1] James Anderson, Emily Spurgeon, Brian Stirling, Jack May, Patrick Rex, Bobby Hyla, Steve McCullough, Marten Thompson, and Christopher Lowe. 2022. High resolution acoustic telemetry reveals swim speeds and inferred field metabolic rates in juvenile white sharks (Carcharodon carcharias). *PLoS ONE* 17, 6 (06 2022).

[2] Haluk Bayram, Nikolaos Stefa, Kazim Selim Engin, and Volkan Isler. 2017. Tracking Wildlife with Multiple UAVs: System Design, Safety and Field Experiments. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, Los Angeles, CA, USA, 97–103.

[3] Chandreyee Bhowmick and Laxmidhar Behera. 2015. Tracking of a randomly moving target by flock of multiple agents using leader-follower approach. In *2015 39th National Systems Conference (NSC)*. IEEE, Greater Noida, India, 1–5.

[4] Aseem Borkar and Girish Chowdhary. 2021. Multi-agent Aerial Monitoring of Moving Convoys using Elliptical Orbits. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Xi'an, China, 8999–9005.

[5] Christopher M. Clark, Stephen M. Rock, and Jean-Claude Latombe. 2003. Dynamic Networks for Motion Planning in Multi-Robot Space Systems. In *Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space*. i-SAIRAS, NARA, Japan.

[6] John Eiler, Thomas Grothues, Joseph Dobarro, and Michele Masuda. 2014. Comparing Autonomous Underwater Vehicle (AUV) and Vessel-based Tracking Performance for Locating Acoustically Tagged Fish. *Marine Fisheries Review* 75, 6 (02 2014).

[7] John H. Eiler, Thomas M. Grothues, Joseph A. Dobarro, and Rahul Shome. 2019. Tracking the Movements of Juvenile Chinook Salmon using an Autonomous Underwater Vehicle under Payload Control. *Applied Sciences* 9, 12 (2019).

[8] Christina Forney, Manii Esfandiar, Michael Farris, M. Moline, Lowe Christopher, and Christopher Clark. 2012. Tracking of a tagged leopard shark with an AUV: Sensor calibration and state estimation. In *Proceedings - IEEE International Conference on Robotics and Automation*. IEEE, St. Paul, MN, USA, 5315–5321.

[9] Richard Grainger, David Rauberheimer, Victor M. Peddemors, Paul A. Butcher, and Gabriel E. Machovsky-Capuska. 2022. ntegrating Biologging and Behavioral State Modeling to Identify Cryptic Behaviors and Post-capture Recovery Processes: New Insights From a Threatened Marine Apex Predator. (Jan. 2022).

[10] Neil Hammerschlag, Austin J Gallagher, and D.M. Lazzarre. 2011. A review of shark satellite studies. *Journal of Experimental Marine Biology and Ecology* 398, 1-2 (Feb. 2011).

[11] Elizabeth T. Küsel, Tessa Munoz, Martin Siderius, and David Mellinger. 2017. Marine mammal tracks from two-hydrophone acoustic recordings made with a glider. *Ocean Science Discussions* 13, 2 (April 2017).

[12] Marco Lanzagorta. 2012. *Underwater Communications*. U.S. Naval Research Laboratory and Morgan Claypool Publishers, Hershey, PA, USA, Synthesis lectures on communications Underwater Optical Communications: System Performance, 1–129. https://doi.org/10.2200/S00409ED1V01Y201203COM006

[13] Yukun Lin, Jerry Hsiung, Richard Piersall, Connor White, Christopher G. Lowe, and Christopher M. Clark. 2016. A Multi-Autonomous Underwater Vehicle System for Autonomous Tracking of Marine Life. *Journal of Field Robotics* 34, 4 (Aug. 2016).

[14] Ryan Luna and Kostas E. Bekris. 2011. Efficient and complete centralized multi-robot path planning. In *IEEE International Workshop on Intelligent Robots and Systems (IROS)*. IEEE, Francisco, CA, USA, 3268–3275.

[15] Hang Ma. 2022. Graph-Based Multi-Robot Path Finding and Planning. *Current Robotics Reports* 3 (2022).

[16] Lili Ma. 2018. Cooperative Target Tracking using a Fleet of UAVs with Collision and Obstacle Avoidance. In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, Sinaia, Romania, 652–658.

[17] Siddharth Mayya, Ragesh K. Ramachandran, Lifeng Zhou, Gaurav S. Sukhatme, and Vijay Kumar. 2022. Adaptive and Risk-Aware Target Tracking with Heterogeneous Robot Teams. *IEEE Robotics and Automation Letters* 7, 2 (April 2022).

[18] B. McClintock, R. Langrock, O. Gimenez, E. Cam, D. Borchers, R. Glennie, and T. Patterson. 2021. Uncovering ecological state dynamics with hidden Markov models. *Ecology Letters* 24, 5 (2021).

[19] Tiago P. Nascimento, Andre G. S. Conceicao, and Ant´onio Paulo Moreira. 2014. Multi-Robot Systems Formation Control with Obstacle Avoidance. In *Proceedings of the 19th World Congress The International Federation of Automatic Control*. IFAC, Cape Towen, South Africa, 6648–6653. https://doi.org/10.3182/20140824-6-ZA-1003.01848

[20] Toby A. Patterson, J. Paige Eveson, Jason R. Hartog, Scott Cooper Karen Evans, Alistair J. Hobday Matt Lansdell, and Campbell R. Davies. 2018. Migration dynamics of juvenile southern bluefin tuna. *Scientific Reports* 8 (May 2018).

[21] Toby A. Patterson, Alison Parton, Roland Langrock, Paul G. Blackwell, Len Thomas, and Ruth King. 2017. Statistical modelling of individual animal movement: an overview of key methods and a discussion of practical challenges. *AStA Advances in Statistical Analysis* 101 (2017).

[22] Huy X. Pham, Hung La, David Feil-Seifer, and Matthew Deans. [n. d.]. A Distributed Control Framework for a Team of Unmanned Aerial Vehicles for Dynamic Wildfire Tracking. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), publisher = "IEEE", address=Vancouver, BC, Canada, year=2017, month=September, pages=6648-6653, url = https://doi.org/10.1109/IROS.2017.8206579, doi=10.1109/IROS.2017.8206579.*

[23] Anwar Quraishi and Alcherio Martinoli. 2021. Coordinated Path Planning for Surface Acoustic Beacons for Supporting Underwater Localization. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Prague, Czech Republic, 4343–4349.

[24] Ragesh K. Ramachandran, Nicole Fronda, and Gaurav S. Sukhatme. 2020. Resilience in multi-robot target tracking through reconfiguration. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Paris, France, 4551–4557.

[25] Cyril Robin and Simon Lacroix. 2022. Multi-robot target detection and tracking: taxonomy and survey. *Autonomous Robots* 40, 4 (May 2022).

[26] Jur van den Berg, Stephen J Guy, Ming C Lin, and Dinesh Manocha. 2011. *Reciprocal n-Body Collision Avoidance*. ISRR, Robotics research Reciprocal n-Body Collision Avoidance, 3–19. http://dx.doi.org/10.1007/978-3-642-19457-3_1

[27] Janardan Kumar Verma and Virender Ranga. 2021. Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope. *Journal of Intelligent Robot Systems* 10 (April 2021).

[28] Frederick A. Voegeli, Malcolm J. Smale, Dale M. Webber, and Yanko Andrade. 2001. Ultrasonic Telemetry, Tracking and Automated Monitoring Technology for Sharks. *Environmental Biology of Fishes* (Feb. 2001).

[29] Jane Wu, Russel C. Blingham, Samantha Ting, Kolton Yager, Zoe J. Wood, Timmy Gambin, and Christopher M. Clark. 2018. Multi-AUV motion planning for archeological site mapping and photogrammetric reconstruction. *Journal of Field Robotics* 36 (2018).

[30] Yang Yang, Juntao Li, and Lingling Peng. 2020. Multirobot path planning based on a deep reinforcement learning DQN algorithm. *CAAI Transactions on Intelligence Technology* 5, 3 (2020).