RRT* Trajectory Scheduling Using Angles-Only Measurements for AUV Recovery

Xuezhi Wang[1], Simon Williams[2], Daniel Angley[2], Christopher Gilliam[1], Trevor Jackson[3], Richard Ellem[3], Amanda Bessell[3], Bill Moran[2]

[1] School of Engineering, RMIT University, Australia
[2]Dept. Electrical & Electronic Engineering, University of Melbourne, Australia
[3] Maritime Division, Defence Science and Technology Group, Australia

***Abstract*—Sensor trajectory optimisation involves extensive search over the sensor motion space against an optimisation criterion. The search under dynamic programming or fixed grid is often computationally nontrivial even for a myopic search scenario. In this paper, we study the problem of an autonomous underwater vehicle planning its return route to a moving recovery vessel. To complicate the issue, the AUV needs to localize the vessel using angle-only measurements. Accordingly, we propose a random sampling based trajectory planning algorithm that incorporates both a dynamic goal and the need to localize that goal. More precisely, we incorporate an information theoretic cost into a rapid-exploring random tree trajectory planning framework thus allowing the AUV to both localize and reach the recovery vessel. Our experimental results show that the proposed method may achieve the same trajectory optimisation performance as that under dynamic programming method but with greater computational efficiency.**

***Index Terms*—Trajectory optimisation, path planning, angles-only tracking, RRT*, AUV**

I. Introduction

Autonomous Underwater Vehicles (AUV) are expected to play an increasingly important role in a wide range of undersea applications such as conducting surveys to map the seabed and locate bottomed objects, sensing and characterising the undersea environment, and monitoring the movements and behaviours of surface vessels, underwater vehicles, other sub-surface objects and/or marine life [1]. Vehicles operating in the underwater environment often rely on sonar as the primary method for sensing and communicating over extended ranges due to the significant attenuation of electromagnetic signals. Fig. 1 illustrates the scenario we used to explore a passive sonar based application, an AUV operating in cooperation with a surface vessel that supports deployment and recovery of the AUV. The AUV relies on knowledge of recovery vessel state, which is estimated based on an angles-only tracker using onboard passive sonar measurements, to steer itself back to the safe recovery region.

Tracking of a moving target from a single angles-only sensor requires the sensor to perform maneuvers during the process of measurement to acquire observability required for estimating the kinematic state of the underlying target.

This work is partially supported by the Defence Science and Technology Group Australia under research agreement ID7966 (SW394606).

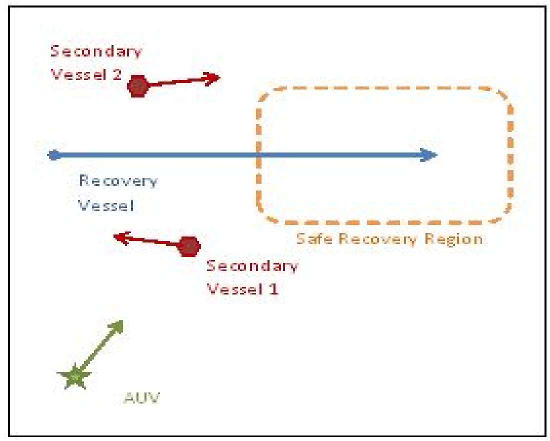


Fig. 1. Recovery vessel transit between AUV and multiple secondary vessels

The trajectory of sensor motion is critical for the reduction of estimation error and a trajectory optimisation process is generally necessary for driving the sensor movement in a way to allow the tracker to maintain the minimal estimation error. This operation under traditional dynamic programming implementation is based on a N-step ahead fixed grid search strategy. While the expected precision can be achieved by choosing $N > 1$ steps with a fine grid of sensor motion hypotheses, the associated computational complexity is usually beyond acceptable as the number of hypothesis to be dealt with grows exponentially with the number of steps involved.

Alternatively, trajectory optimisation can be performed using random sampling based techniques which can handle a large N steps with computational complexity linear to the number of samples used. The performance of these techniques can approach the optimal solution in the limit where the number of samples becomes large. A representative sampling based approach is the rapid-exploring random tree (RRT) algorithm. It was originally proposed by LaValle in [2] and the standard RRT was later extended to the RRT* algorithm in [3]. Several variations appeared in the literature including the Linear Quadratic Regulation RRT* [4] and Incremental Sampling-based Methods [5]. The RRT algorithm for trajectory optimisation with unpredictable obstacles was reported in [6]. A summary of recent sampling based techniques is presented in [7].

The sensor trajectory optimisation problem is also known as trajectory scheduling or path planning in the literature [8]. This is cast as a partially observed Markov decision process (POMDP) [9], [10], where the decision process is carried out by minimising the cost or maximising the reward against a measurable criterion that is related to the Fisher information [11]–[13] or mutual information [14], [15].

In this work, we propose a RRT* based algorithm with a modified sampling picking procedure for optimising AUV trajectory in its recovery process. For angles-only tracking, a multi-step ahead trajectory optimisation is performed between every adjacent sampling points, but the AUV is only driven one step at every epoch. Therefore, a biased tree growing strategy which draws more samples in the vicinity of the sensor/AUV location to cover all possible kinematic state hypotheses of the AUV is preferable. We show by simulation that computational complexity is substantially reduced by growing the tree branches from the AUV using distance ordered samples in the AUV trajectory optimisation process.

The angles-only tracking problem is described in Section II. We propose the progressive RRT* approach in Section III. Experiment setup and results discussion are presented in Section IV which is followed by the conclusions in Section V.

II. The Angles-Only Tracking Problem

Angles only tracking problem can be illustrated by a 2D example shown in Fig. 2, where the sensor with initial position $\mathbf{x}_0 = (x_0, y_0)'$ is used to estimate the position $\mathbf{x} = (x, y)'$ of a static target by observing the line of sight angle between the target and itself. In order to observe $\mathbf{x}$, the sensor must take another measurement at a different position $\mathbf{x}_1 = (x_1, y_1)'$. Therefore, we have the following measurement model:

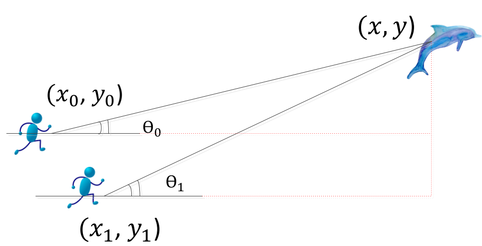


Fig. 2. Measurement of angles-only tracking.

$$\mathbf{z} = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{y-y_0}{x-x_0}\right) \\ \arctan\left(\frac{y-y_1}{x-x_1}\right) \end{bmatrix} + \omega, \quad (1)$$

where ω is assumed to be a Gaussian zero-mean random variable with standard deviation σ to signify observation noise.

Remarks:

- For moving targets, future target states are approximated from the tracker prediction.
- To solve $\mathbf{x} = (x, y)'$, the necessary condition is $\Delta\theta = \theta_1 - \theta_0 \neq 0$. This requires the sensor moving to $\mathbf{x}_1 = (x_1, y_1)'$ at a non-zero correction angle φ. Let r be the sensor travel distance from $\mathbf{x}_0 = (x_0, y_0)'$ to $\mathbf{x}_1 = (x_1, y_1)'$, and d the distance between $\mathbf{x}_0 = (x_0, y_0)'$ and $\mathbf{x} = (x, y)'$. We can show that the optimal sensor heading under the maximum determinant of Fisher information matrix on (1) is given by

$$\varphi_{opt} = \pm \arctan\left(\frac{|r^2 - d^2|}{2rd}\right) \quad (2)$$

  Based on (2), the optimal sensor course correction φ_{opt} vs. the distance r that the sensor is able to move in one step is plotted in Fig. 3.
- Fig. 3 indicates that the optimal sensor course correction is $\varphi_{opt} \leq \pi/2$ while the sensor is approaching to target.

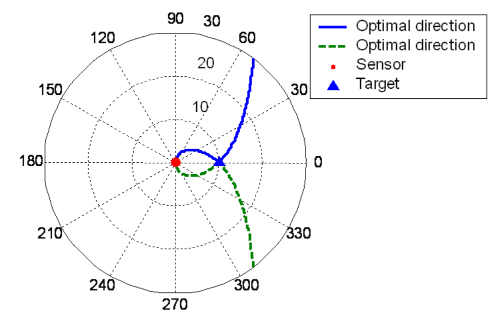


Fig. 3. Optimal sensor movement direction for given radius constraints to maximize information gain in tracking a stationary target. Two symmetric solutions are indicated in different colors.

For the problem of an AUV chasing a recovery vessel, the AUV observation angles include bearing and elevation angles between the recovery vessel (target) and itself. Without loss of generality, in this paper we only consider a 2D case, i.e., angles-only measurements in our simulation. The determinant of the Fisher information matrix $\boldsymbol{G}$ associated with the sensor observation (1) is used as the reward function in the AUV trajectory optimisation. The Fisher information matrix $\boldsymbol{G}$ for the measurement (1) is defined as

$$\boldsymbol{G} \triangleq E\Big[\big(\nabla_{\mathbf{x}} \ln p(\mathbf{z}|\mathbf{x})\big)\big(\nabla_{\mathbf{x}} \ln p(\mathbf{z}|\mathbf{x})\big)'\Big]_{\mathbf{x}\approx\hat{\mathbf{x}}}, \quad (3)$$

where $\ln p(\mathbf{z}|\mathbf{x})$ is log-likelihood, target state $\mathbf{x}$ is approximated by the posterior state estimate $\hat{\mathbf{x}}$ obtained by the tracker. Thus, during each step the AUV will move to the point $\mathbf{x}_k^s, k = 1, 2, \cdots$ such that the accumulated determinant of Fisher information matrix $\sum_{i=1}^{k} \boldsymbol{G}_i$ at $\mathbf{x}_k^s$ is maximised [16].

III. Progressive RRT* Approach

A. Procedure of the standard RRT* scheduling

The standard RRT* algorithm grows a tree rooted from the sensor location by using random samples drawn in the location search space, iteratively. The tree is defined by the set of selected node locations, denoted by $\boldsymbol{B} = \{\mathbf{q}_{s,1}, \mathbf{q}_{s,2}, \cdots\}$, where $\mathbf{q}_{s,i} = (x_i, y_i)$ for 2D case. At each iteration, the process involves

1) Draw a random sample $\mathbf{q}_r$.
2) We then find the node, denoted by $\mathbf{q}_n$, in the tree/branch which is closest to $\mathbf{q}_r$.
3) Steer from $\mathbf{q}_n$ to $\mathbf{q}_r$ reaching to $\mathbf{q}_{new}$, a node which the sensor may reach feasibly without colliding with an obstacle.
4) Update the cost/reward for the sensor moving from the parent $\mathbf{q}_n$ to $\mathbf{q}_{new}$ against the required optimisation criterion. For angles-only tracking, the accumulated determinant of the Fisher information matrix is computed as the reward.
5) Determine the parent node of $\mathbf{q}_{new}$ by selecting the node of lowest cost (or highest reward) without colliding with an obstacle from the tree in the circular boundary defined by the sensor maximum moving distance between adjacent sampling epochs centered at $\mathbf{q}_{new}$;
6) Add $\mathbf{q}_{new}$ into the tree $\boldsymbol{B}$;
7) Repeat the procedure from 1) until the number of samples is reached or the sensor achieves the intended goal of closing within a specified distance of the recovery vessel.

Remarks: In the angles-only AUV scheduling problem, the trajectory optimisation is computed over a multiple step look ahead, however the sensor only moves along the first step at each time epoch. Thus, it is important to have enough samples covering all possible sensor trajectories, in particular, in the first few steps. On the other hand, computational complexity of the RRT* algorithm is proportional to the number of samples used. This suggests the use of a biased tree in the RRT* implementation which can achieve the sensor trajectory optimisation efficiently. The random tree can be biased by increasing the probability of sampling states from a specific area. We discuss this further in the next section.

B. RRT* method discussion

As suggested in [5], several approaches can be used to generate a biased tree for our angles-only tracking problem. Here we discuss two of them: 1) Random sample picking (standard); 2) Distant progressive sample picking.

Random sample picking: draw samples uniformly over the area of interest and pick a sample randomly each time for growing the tree branch.

Progressive distance sample picking: draw samples uniformly over the area of interest and pick a sample based on the order from near to far distance between the sample and sensor in the "tree exploring" process.

An obvious drawback of the progressive distance sample picking is that samples behind a large obstacle are eliminated in the tree branch growing process, as shown in Fig. 4. Nevertheless, this method may be used for the following reasons.

1) The sensor trajectory optimisation strategy: *at each optimisation circle, sensor will move only one step along the optimised path.*

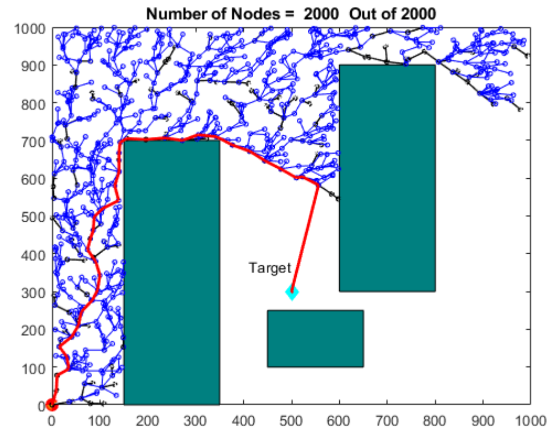


Fig. 4. Using the progressive distance sample picking method, samples behind a fully blocked obstacle will be lost. Note that this single run results in 39 steps of sensor movement while in the proposed scheduling scenario, the sensor is only allowed to move one step forward from a single run.

2) While there are many possibilities for the first step of sensor movement, the best "first step" requires a large number of samples to identify.
3) RRT* algorithm tends to be optimal for path searching/planning as the number of samples becomes large.
4) The progressive distance sample picking method enables the possibility to cover the area near the sensor with full sample density by using a lower number of samples.
5) For angles-only tracking, it is important to have a high density of samples in the proximity of the sensor so that all possible directions in the field of sensor view can be covered.

Next, we compare the two sample picking methods for the RRT* algorithm using a 2D static target tracking example. Here, the accumulative log determinant of the Fisher information matrix, obtained based on the observation model (1) between adjacent steps along each of all paths, serves as the reward. During each sampling period, the sensor is steered along the path of maximum reward.

C. Random sample picking vs. progressive distance sample picking

The sensor starts from $(0, 200)$ and moves toward a static target at $(900, 900)$ with a field of view angle of 360^o. During each sampling interval, which we call a step, the sensor is able to move a maximum distance of 100 m and receives one measurement from the target. The standard deviation of sensor measurement noise ω is assumed to be 2^o. The scenario is shown in the simulation result plot Fig. 5, where the green circles are obstacles of known locations in the experiment. In practice, they represent the areas occupied by other vessels and their locations are unknown and estimated by the tracker as well and we assume that they can be distinguished from the recovery vessel by their motion dynamics.

For the progressive distance sample picking method, we chose to use samples in a reduced area centered at the sensor with 1/16 the number of samples that were used for the random sample picking method. On the other hand, the random sample picking method uses all samples which are uniformly drawn from the area of interest. The comparison of the two methods is summarised in Table I, where "Total CPU time" is the total computational overhead for optimising the sensor trajectory from the start point to the target location. "Mean No. ahead steps" is the average number of target measurements received by the sensor during the sampling period that are used by the RRT* algorithm to plan ahead. This number becomes smaller as the sensor gets closer to the target. The comparison of optimised sensor trajectories between the two sample picking methods under different sample sizes are shown in Fig. 5 (a) and (b), respectively. These plots illustrate how the optimised trajectory varies with the number of samples and cost criterion used as well.

The above simulation analysis suggests that the RRT* algorithm with a progressive distance sample picking method achieves a better compromise between accuracy, performance and computational complexity.

TABLE I
PROGRESSIVE DISTANCE SAMPLE PICKING VS. RANDOM SAMPLE PICKING

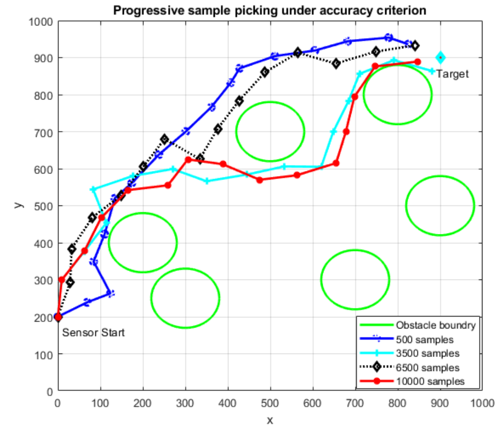
No. of Samples	500	3500	6500	10000
Progressive distance sample picking				
Total CPU time (s)	0.3204	8.5704	32.2665	61.5984
Mean No. ahead steps	4.4706	4.8000	5.1429	5.1429
Random sample picking				
Total CPU time (s)	9.2299	350.0706	1038.8	2458.8
Mean No. ahead steps	10.7778	10.3125	9.6923	9.9231

IV. EXPERIMENTAL RESULTS AND DISCUSSION

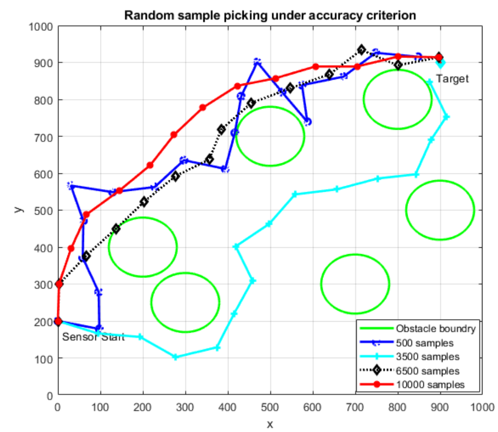
In this section, we demonstrate the performance of RRT* algorithm with the method of progressive distance sample picking for the application of an AUV chasing a moving recovery vessel using angles-only observations. The angle measurement errors are assumed to be Gaussian distributed with zero-mean and standard deviation of 2^o. As shown in Fig. 6, the AUV starts chasing from the origin at a speed no greater than 60 meters per minute and it measures the recovery vessel bearing once per minute. Initially, the recovery vessel is at the position (300, 350) meters and it is moving at a constant velocity of $[12, 8]^T$ meters per minute when the AUV chasing starts. Six other vessels are distributed over the locations as shown in the figure. The location uncertainties are denoted using circles with radius equal to 80 m. While angle measurements of the recovery vessel can be received at any location, the AUV is not allowed to enter the other vessel uncertainty regions. The chasing will stop when the AUV enters into a circular recovery area centered at the recovery vessel with radius of 60 meters.

One hundred Monte Carlo runs are performed for each of the following cases:

1) 3000/6000 samples are drawn uniformly over the area. During each sampling period, the AUV moving trajec-



(a)



(b)

Fig. 5. RRT* method analysis: Optimised sensor trajectory in different sample sizes: (a) Progressive distance sample picking using 1/6 of total number of samples. (b) Random sample picking using full number of samples.

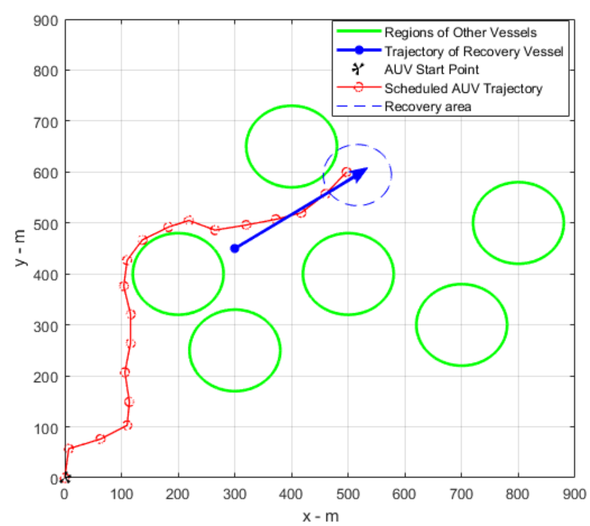


Fig. 6. Illustration of the scenario of an AUV chasing a recovery vessel, where a typical AUV moving trajectory scheduled via RRT* algorithm is given.

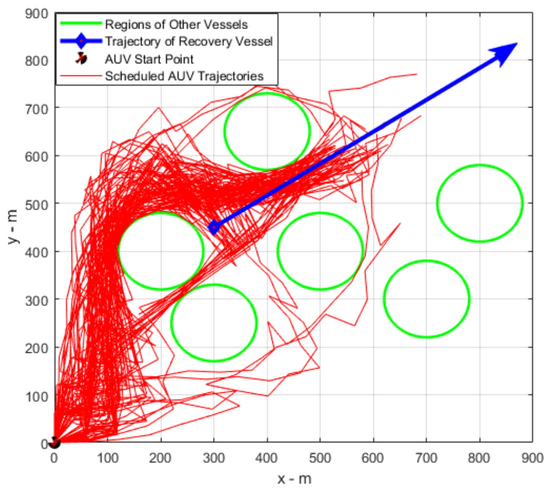


Fig. 7. AUV moving trajectories optimised by the RRT* with random sample picking which uses all 6000 samples drawn uniformly over the entire area of interest.

tory is optimised using the RRT* algorithm with a random sample picking method (standard RRT* procedure).

2) 3000/6000 samples are drawn uniformly over the area. During each sampling period, the progressive distance sample picking method is used to select 1/6th of the total 3000/6000 samples located in the vicinity of the estimated recovery vessel location.
3) 10000 samples are drawn uniformly over the area. During each sampling period, the progressive distance sample picking method is used to select 1/6th of the total 10000 samples located in the vicinity of the estimated recovery vessel location.

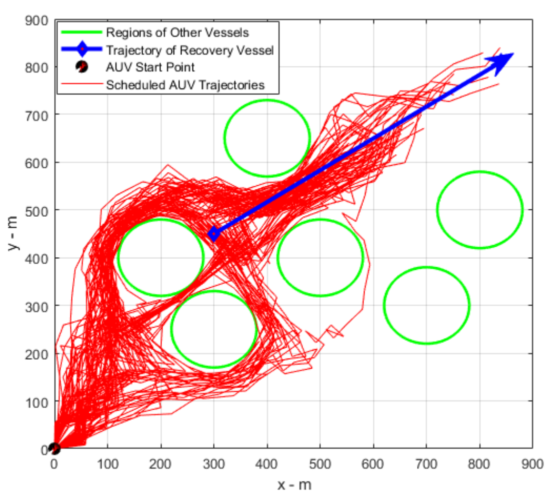


Fig. 8. AUV moving trajectories optimised by the RRT* with progressive distance sample picking which uses 1/6 of total 6000 samples drawn uniformly over the entire area of interest.

Simulation results are shown in Figures 7, 8, 9 and Table II, respectively. Red lines are the optimized UAV trajectories from 100 Monte Carlo runs. These results show:

- **Total CPU time** - which shows the actual computational complexity for the underlying case.
- **Number of steps ahead scheduled in the RRT*** to determine the AUV first step movement, where a large number signifies a better optimisation strategy but a big computational overhead.
- **The total number of measurements taken** during the AUV chasing process, which reflects the total time taken to complete the chase.

TABLE II
RANDOM SAMPLE PICKING

No. of Samples	3000	6000	10000
Random sample picking			
Total CPU time (s)	334.5	1283.6	–
Total No. of meas. taken	21	20	–
No. steps scheduled for taking 1st meas.	16.9	16.3	–
Progressive distance sample picking			
Total CPU time (s)	8.7	28.6	72.3
Total No. of meas. taken	28.5	25.8	23.5
No. steps scheduled for taking 1st meas.	10	10	10.2

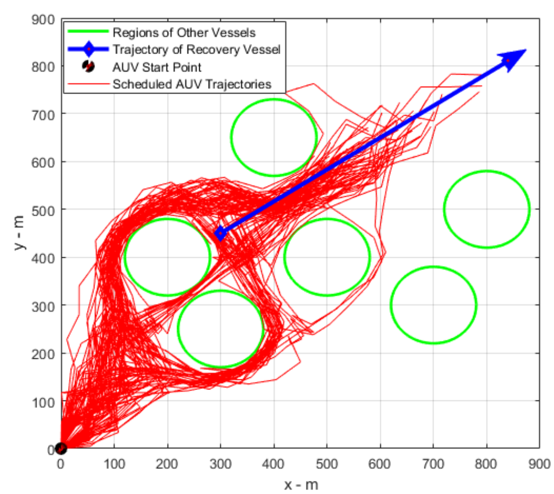


Fig. 9. AUV moving trajectories optimised by the RRT* with progressive distance sample picking which uses 1/6 of total 10000 samples drawn uniformly over the entire area of interest.

From the simulation results, we observed that

- the RRT* scheduling performed on a full scale (i.e., Random sample picking method) is the ideal approach and approaches the optimal solution as the number of samples becomes large. As shown in Fig. 7, scheduling based on the standard RRT* approach with 6000 samples drawn, results in the lowest number of measurements (21). However, the actual computational overhead associated is beyond practical (i.e., more than 21 minutes).
- when the scheduling is done using the RRT* with the progressive distance sample pick method, which uses only

those samples in the vicinity of AUV, the computational overhead is substantially reduced. As shown in Figures 8, 9 and Table II, while more efficient, the required chasing time, or in other words, the average number of measurements taken is increased from 21 to 28.5 with 3000 samples, 25.8 with 6000 samples and 23.5 with 1000 samples.
- Fisher information approximated with the posterior estimate of recovery vessel is chosen as the reward function. This will maximise knowledge of recovery vessel location.

This result suggests that for the AUV chasing recovery vessel application, the RRT* algorithm with progressive distance sample picking may provide a better trade-off between feasibility, chasing time and number of samples drawn.

V. Conclusions

In this paper, we implement the RRT* algorithm with a biased random tree for the application of an AUV trajectory scheduling approach for chasing a recovery vessel using angles-only measurements. The AUV is required to chase the recovery vessel and reach it as soon as possible whilst simultaneously steering its trajectory in order to minimise the estimation error of the recovery vessel target state and to avoid entering the areas surrounding other vessels. Our simulation results show that for the autonomous AUV recovery application, the proposed method can provide a better trade-off between feasibility, chasing time and number of samples drawn. On the other hand, running the standard RRT* procedure may have issues associated with either lack of samples or excessive computational resource requirements.

References

[1] G. Best and S. Anstee, "Motion planning for autonomous underwater vehicle supervision," in *Proceedings of the (2014) Australasian Conference on Robotics and Automation, ACRA*. Melbourne, Australia, December 2014.

[2] Steven M. LaValle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning", *Technical Report (TR 98-11)*, Computer Science Department, Iowa State University, 1998.

[3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[4] R. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "LQR-RRT*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *IEEE International Conference on Robotics and Automation*, 2012.

[5] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *49th IEEE Conference on Decision and Control (CDC)*, Dec 2010, pp. 7681–7687.

[6] M. W. Otte and E. Frazzoli, "Rrtx: Real-time motion planning/replanning for environments with unpredictable obstacles," in *International Workshop on the Algorithmic Foundations of Robotics (WAFR)* , Istanbul, Turkey, 2014.

[7] Short, A., Pan, Z., Larkin, N. and van Duin, S, "Recent progress on sampling based dynamic motion planning algorithms", *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp.1305–1311, 2016 USA.

[8] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, March 2016.

[9] V. Krishnamurthy, "Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors", *IEEE Trans. Signal Processing*, vol. 50, no. 6, pp. 1382-1397, June 2002.

[10] J.-P. Le Cadre, and O. Trémois, "Optimization of the Observer Motion Using Dynamic Programming", *Proc. ICASSP'95*, vol. 5, pp. 3567–3570, Detroit, Michigan, 1995.

[11] J. A. Fawcett, "Effect of course maneuvers on bearings-only range estimation", *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. 36, no. 8, pp. 1193–1199, August 1988.

[12] J.-P. Le Cadre, and S. Laurent-Michel, "Optimizing the receiver maneuvers for bearings-only tracking", *Automatica*, vol. 35, pp. 591–606, 1999.

[13] J. P. Helferty, D. R. Mudgett, and J. E. Dzidlski, "Trajectory optimization for minimum range error in bearings-only source localization", *Proc. Oceans' 93 Engineering in Harmony with Ocean*, pp. 229–234, 1993.

[14] A. Logothetis, V. Krishnamurthy, J. Holst, and A. Isaksson, " Modal State Estimation of a Maneuvering Target in Clutter", *Proc. 36th IEEE Conf. on Decision and Control*, San Diego, pp.5024–5029, Dec. 1997.

[15] A. Logothetis and V. Krishnamurthy, "Modal Trajectory Estimation for Jump Markov Linear Systems via the Expectation Maximization Algorithm", *Proc. 36th IEEE Conf. on Decision and Control*, San Diego, USA, pp.1700-1705, Dec. 1997.

[16] X. Wang, Y. Cheng, and B. Moran, "Nonlinear parameter estimation in statistical manifolds," in *Proceedings of the Eighth IEEE Sensor Array and Multichannel Signal Processing Workshop*, Coruna, Spain, June 22-25 2014.