Smoothing optimal RRT* trajectories for recovery of an AUV by a moving surface vessel.

Simon Williams[a], Xuezhi Wang[b], Daniel Angeley[a], Christopher Gilliam[b],
Bill Moran[a], Trevor Jackson[c], Richard Ellem[c], Amanda Bessell[c]
[a] University of Melbourne, [b] RMIT University, [c] DST Group

ABSTRACT

In this paper we will show how the use of a kinematic constraint (a Dubins Path) can smooth optimal paths generated by a probabilistic algorithm (RRT*) for generating optimal paths. We have extended the original algorithm to take account of moving objectives, extended the cost functions to plan trajectories, improve the accuracy of sensor measurements, and incorporate position estimates and covariances into the sampling mechanism to focus the trajectories.

The main drawback of these methods is that the optimality is only probabilistic. Consequently, any finite implementation produces variable and disjointed trajectories that are not suitable in reality. Several smoothing techniques were tried and the Dubin's path produced the best results.

Keywords: sonar, trajectory optimsation, AUV

1. 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. Vehicles operating in the underwater environment typically rely on sonar (i.e. underwater acoustics) as the primary method for sensing and communication over extended ranges due to the significant attenuation of electromagnetic signals. Here we consider using a passive sensor on an AUV operating in cooperation with a surface vessel that supports deployment and recovery of the AUV.

The problem of path planning for a mobile surface vessel that maintains supervisory contact (using an acoustic communications link) with an AUV executing a pre-planned mission was considered by Best.[1] Here we consider an AUV that conducts survey missions in a more autonomous manner, and on completion of the mission or when signalled by the support vessel via the communication link, uses its on-board sensors and optimal trajectory path planning to navigate safely back towards the support vessel to rendezvous and surface within a safe standoff distance for subsequent recovery. The support vessel is assumed to be mobile and is expected to transit along a straight line trajectory at a constant relatively slow speed towards a safe region (away from other vessels) to prepare for recovery of the AUV. Signals transmitted via the acoustic communications link can be used to assist the AUV with the detection and identification of the support vessel but are not used to control or program the trajectory or destination of the AUV. The final recovery location will not be known to the AUV or the support vessel in advance and will depend on how long it takes for the AUV to navigate to a safe recovery location.

The AUV is equipped with a forward facing high-frequency sonar sensor array that is used for sensing and navigation, which operates primarily as a passive acoustic sensor. The sensor array receives the acoustic energy emitted from all nearby vessels or objects located within the sensor's field of view, defined by the angular sector containing bearing and elevation angles within $\pm 60^\circ$, with a finite angular beam resolution of $\pm 3.5^\circ$. An on-board sonar processing system determines the direction of arrival of the received acoustic signals and uses this to estimate and track the positions and velocities of all observable signal sources over time. The estimated state information associated with each track is then used for optimal trajectory path planning to enable the AUV to

Further author information (Send correspondence to S.W.)
S.W.: email simon.williams@unimelb.edu.au

safely navigate towards the recovery vessel in the shortest amount of time while avoiding any other vessels or objects in the area.

The presence of other vessels or objects will impose sensor measurement and physical space constraints within the optimal trajectory path planning algorithms. Sensor measurement constraints result from reduced detectability of the support vessel caused by interference from other noise sources within the same angular sector or increases in overall received noise levels from noise sources in close proximity to the sensor. Physical space constraints result from maintaining minimum safe separation from other vessels or objects, and, from specified non-navigable areas such as shallow water areas, shorelines, islands, physical structures in the water, shipping channels or other restricted areas. These latter 'non-navigable' physical constraints are implemented as pre-programmed geographic regions within the on-board processing system of the AUV.

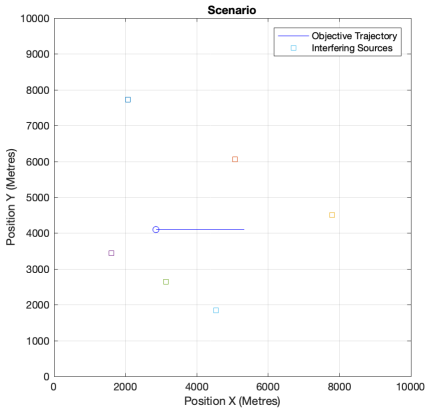


Figure 1: Example scenario for an AUV to navigate. The path of the moving objective is indicated by the blue line and the squares indicate the position of the interfering sources. Note that the AUV is initialised at the origin.

function RRT*$(o, g, \eta,$ N)
  $t \leftarrow \{o\}$
  $n \leftarrow 0$
  while $n < N$ **do**
    $x \leftarrow$SAMPLE
    $x' \leftarrow$STEER(t, x)
    $t \leftarrow t + \{x'\}$
    $t \leftarrow$REWIRE(t)
    $n \leftarrow n + 1$
  end while
  return t
end function

Figure 2: Original RRT* algorithm

1.1 Randomly Exploring Random Trees

Figure 1 shows a scenario where an AUV needs to navigate a fleet of interfering sources using its passive sonar to rendezvous with a moving recovery vessel.

Previous work[2,3] was based on $n-$step lookahead on the tracker processing. The new algorithm is based on the Randomly Exploring Random Trees (RRT*) algorithm developed in the robotics literature[4,5] for motion planning. The probabilistic nature allows soft constraints to be implemented as required.

The RRT* algorithm is provably optimal for a fixed objective and arbitrary allowable manoeuvres at each time step, which does not hold in our case. Figure 2 shows the original RRT* algorithm.

It begins with picking a point at random in the region of interest (SAMPLE). Then you move from the nearest point in the tree towards that point as far as is feasible (STEER) usually by giving a maximum allowable distance travelled η. Limits on feasibility are placed by speed restrictions and obstructions in the region. The cost of this step is then added to the cost recorded at the originating node. But, before this cost is encoded in the tree, a search is conducted locally to see if there is an alternative route to the new point that is lower cost (REWIRE). One of the keys to the RRT* algorithm is that this search need only proceed *locally*, within a distance r given by

$$r = \min\left\{\left(\frac{\gamma \log n}{\xi_d n}\right)^{1/d}, \eta\right\} \tag{1}$$

where d is the dimension of the space, n is the number of sample points, γ is a constant and ξ_d is the volume of the unit ball in $\mathbb{R}^d$ Figure 4 shows how, if a new lower cost link is possible then, the original link is deleted and

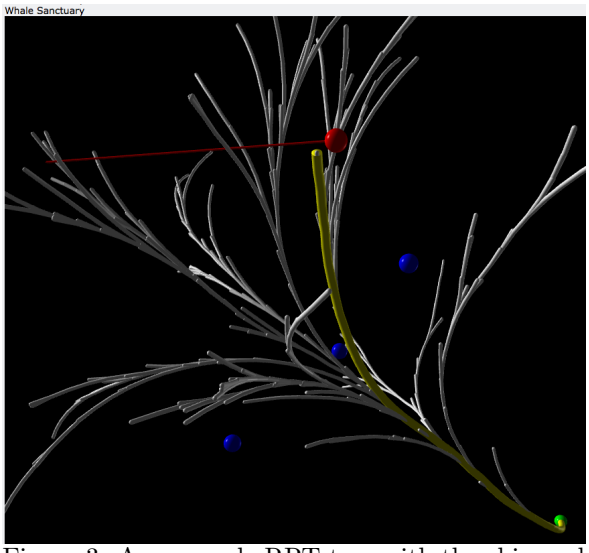


Figure 3: An example RRT tree with the ship and its motion marked in red, interfering sources in blue and the AUV's initial position in green and optimal path in yellow.

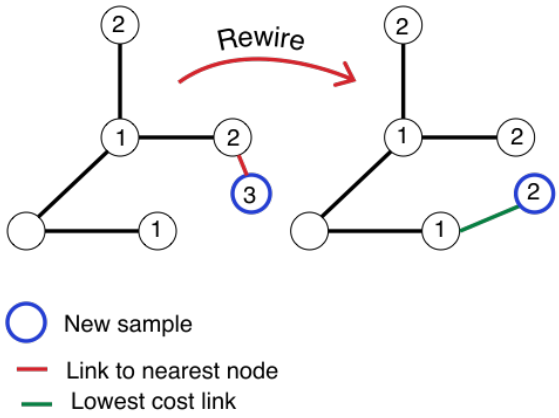


Figure 4: Diagram of rewire step in RRT* algorithm. The new sample (blue) is linked in to the tree at its nearest neigbour (red). A search, that only extends a maximum of one link length around the sample, reveals a lower cost link. The old edge is deleted and the new edge created (green).

the alternative instantiated and the new (lower) cost encoded at the new node.

This algorithm is probabilistically complete and asymptotically optimal for finding a path to a goal through an unobstructed region. Specifically the probability of the algorithm converging to the optimal path increases to 1 as the number of samples N increases.

2. MOTION PLANNING FOR TRAJECTORY OPTIMISATION

In this section we shall describe our extensions to the RRT* algorithm[6,7] to plan optimal trajectories that address the use of mobile objectives, dynamic cost functions, kinematic constarints, focussed sampling, and goal-oriented stopping criteria

2.1 Implementing RRT*-based Trajectory Optimisation

The tree shown in Figure 7 is one step in our trajectory optimisation algorithm. Ultimately only the first step of the yellow path will be used to initialise another run of the RRT* motion planner. The full trajectory optimisation algorithm is described in Figure 6. This involves generating an RRT* tree at each step and using the first edge of the tree that ends closest to the target, where the FirstStep(t, i) returns the first step in the branch of tree t at depth i which is closest to the goal.

2.2 Mobile Objective and Dynamic Costs

In order to incorporate the motion of the objective into the algorithm we take advantage of the fact that the AUV uses a tracker to keep an estimate of the position and velocities of its recovery vessel and the other vessels in its field of view. Using this information and the fact that each step in the tree corresponds to an elapsed time, we can use an updated position of the objective when calculating in the tree.

For example, if our new point x is attached to the tree at a depth k from the origin, then the cost function can be evaluated using the new objective location $l_i = l_0 + kv$ where l_0 is the original location and v is the velocity estimate in units of tree steps η.

This enables the implementation of dynamic cost functions where we penalise the alignment of an interfering source and the objective within the same received beamwidth ($\pm 3.5°$), and maintain the objective within the sensor's field of view ($\pm 60°$). This avoids the loss of detections that would result from the recovery vessel being

function COST(x_i, l_i, s)
  $c \leftarrow \eta$
  if Angle(x_i, s) - Angle(x_i, l_i) $\leq 3.5°$ **then**
    $c \leftarrow c + 50$
  else if Angle($x_i - x_{i-1}, l_i$) $\geq 60°$ **then**
    $c \leftarrow c + 25$
  end if
  return c
end function

Figure 5: Dynamic Cost function

function OPTIMISETRAJECTORY(l, s)
  $path \leftarrow [InitialNode]$
  for i in 1..l **do**
    $tree \leftarrow$RRT*(s)
    $path \leftarrow path +$ FIRSTSTEP($tree, l - i$)
  end for
  return $path$
end function

Figure 6: RRT*-based Trajectory Optimisation

obscured by interfering noise sources associated with other nearby vessels, or located outside the sensor's field of view, which would impact the ability of the tracker to provide accurate target state estimates. The form of these dynamic cost functions is shown in Figure 5, where x_i is the location of the new node which is i steps from the root, l_i is the expected location of the objective after i steps and s are the locations of the interfering sources.

2.3 Kinematic Constraints

The RRT* algorithm[4] has been modified to incorporate motion constraints for the AUV by limiting the maximum change in direction at each time step to 10°. This alters the STEER procedure so that the vector joining the new point x' to the tree always lies within 10°of the previous leg of the tree.

Figure 7 is the result of our modified RRT* algorithm. 1000 samples were taken from a uniform distribution across the region which is a 6x6x6 cube centred at (0,0,0). The AUV starts at (-3,-3,-3) pointing directly at the recovery vessel and has a maximum speed of $\eta = 0.2$ per unit time. The recovery vessel starts at (2,2,-2) moving in direction (0,0,1) at a constant speed of 0.09 per unit time.

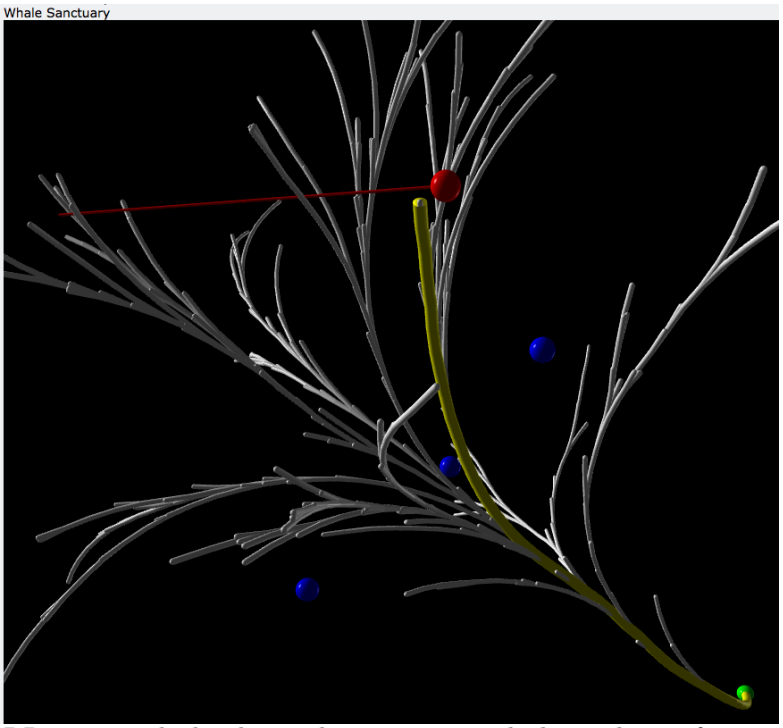


Figure 7: An example RRT tree with the ship and its motion marked in red, interfering sources in blue and the AUV's initial position in green and optimal path in yellow.

2.4 Trackers and Focussed Sampling

We model the measurements made by the AUV's sensor using a simplified sonar equation with a simplified uniform propagation model and a discrete beam pattern for each of the sensor received beam components.

The AUV's tracker is a Sequential Monte-Carlo Multi-Hypothesis Tracker (SMC-MHT) described in.[2,8] It is a multiple hypothesis tracker which detects and estimates the underlying target state with unknown measurement origin by enumerating and evaluating all possible measurement origin hypotheses. The sequential Monte Carlo sampling technique is used to approximate the posterior probabilities of the association hypotheses and make the approach computationally tractable.

It is important to be able to separate out the effect of the tracker on the trajectory optimisation algorithm. We shall see one approach by processing the data in the next section. Our approach is using what we have termed a 'Ground Truth' tracker, which outputs the actual position of the target with a simulated diagonal covariance an order of magnitude smaller than those predicted by the actual trackers. This enables repeatable runs with out interference from the random nature of the tracking algorithms.

Usually the SAMPLE procedure uses a uniform distribution over the area to be covered. Here we use the position and covariance estimates from the tracker and the parameters of a multivariate normal distribution and sample the space focussed around the objective location.

2.5 Goal oriented stopping criteria

In the original RRT* implementation the cost minimised the distance to the objective. However, a more realistic criteria for a moving observer is to use its passive sensor to approach the target, and then to terminate the approach or switch to other close-range sensing methods to avoid complex nonlinear near-field estimation effects associated with angles-only tracking problems. Accordingly the stopping criteria used here was changed to closing within 300m of the target and the cost function adjusted to minimise distance from the target rather than distance travelled.

This is a fundamental distinction between our approach and the other greedy approaches, which are limited in the number of steps ahead they look by the exponential growth of the number of possiblities to consider. Instead, we can set goal-oriented stopping criteria for the trajectory optimisation, like the one detailed above: Stop when you are within 300m of the objective.

3. OPTIMAL PATHS FOR THE RELAXED DUBINS SHORTEST-PATH PROBLEM

A Dubins path is the shortest curve that connects two points in the two-dimensional Euclidean plane (i.e. x-y plane) with a constraint on the turn and with prescribed initial and terminal tangents to the path, and an assumption that the vehicle traveling the path can only travel forward.

In our case we have no constraints on the direction of the AUV at the end of each step. This generalised version of a Dubins path is called the relaxed Dubins shortest-path problem. The standard Dubins shortest path problem specifies an initial and final configuration (x, y, θ), and results in six possible path types: RSR, LSL, RSL, LSR, RLR and LRL, where R is a right turn at minimum turning radius, L is a left turn at minimum turning radius and S is straight ahead. In contrast, the relaxed Dubins problem only has four possible path types: RS, LS, RL and LR. Intuitively, this is because it doesn't need the final turn to fix up the orientation at the final point.

Bui and Boissonnat[9] show the regions where each path type is optimal. If you canonicalise the problem by translating and rotating until the initial configuration is (0, 0, 0), the partition of the plane shown in Figure 8 arises i.e. LS and RS are the optimal paths for all points except those within the minimum turning radius circles to the left and right of the initial configuration.

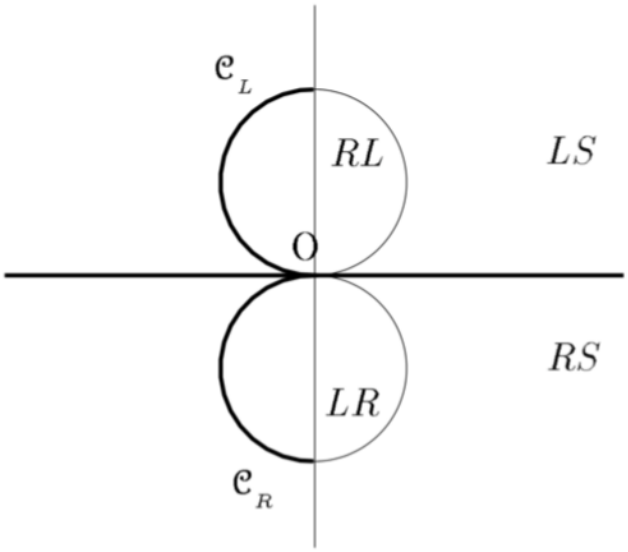


Figure 8: Shows the partition of the plane into Dubins' paths starting from an initial configuration origin moving along the x-axis

4. RESULTS AND CONCLUSIONS

We ran simulations using the length of the Dubins path as the criterion for the optimal trajectory for both the Sequential Monte Carlo Multiple Hypothesis Tracker (SMC-MHT) and Ground-Truth trackers. The results of the comparison with the simulations used in the previous work[6] are shown in Figures 9–12, showing much smoother paths. Figures 9 and 10 shows the tracks in black and the optimal trajectories in red for the straight line and Dubins' path for 30 simulations using the SMC-MHT tracker and Ground truth tracker respectively. Figures 11 and 12 show a histogram of the relative trajectories between the AUV and the estimated position of the recovery ship over the same 30 simulations.

It is clear from both the raw and relative trajectories that using the Dubins model has smoothed the paths generated compared to the straight line algorithm. As an indication of the narrowing of the relative histogram the smallest eigenvalue of the covariance of the data respresented in Figure 11 is 2.65×10^4 for (a) the straight line but only 1.52×10^4 for the relaxed Dubins paths.

REFERENCES

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

[2] Morelande, M. and Ellem, R., "Trajectory optimisation for 3D angle-only tracking (Milestone A)," tech. rep., University of Melbourne (5 2014).

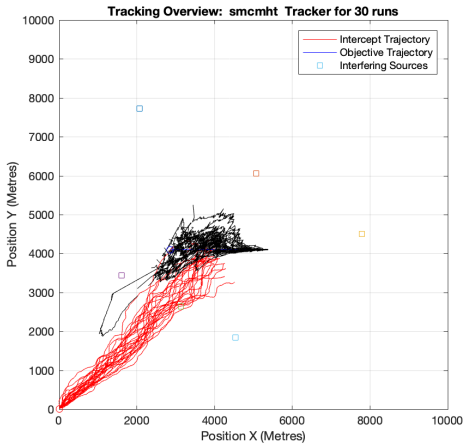
[3] Morelande, M. and Ellem, R., "Trajectory optimisation for 3D angle-only tracking (Milestone B)," tech. rep., University of Melbourne (9 2014).

[4] Karaman, S. and Frazzoli, E., "Incremental sampling-based algorithms for optimal motion planning," *Robotics Science and Systems VI* **104**, 2 (2010).

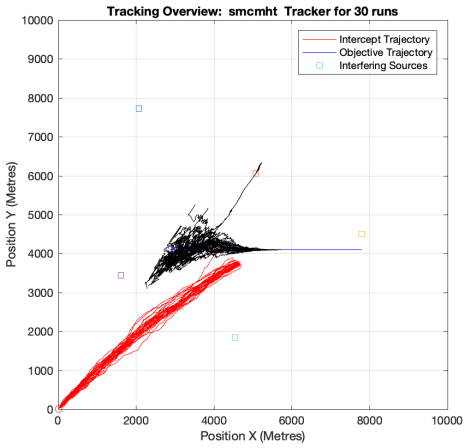
[5] Karaman, S. and Frazzoli, E., "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research* **30**(7), 846–894 (2011).

[6] Williams, S., Wang, X., Angeley, D., Gilliam, C., Moran, B., Jackson, T., and Ellem, R., "Dynamic target driven trajectory planning using RRT*," in *22nd International Conference on Information Fusion*.[10]

[7] Wang, X., Williams, S., Angeley, D., Gilliam, C., Moran, B., Jackson, T., and Ellem, R., "RRT* trajectory scheduling using angles-only measurements for AUV recovery," in *22nd International Conference on Information Fusion*.[10]

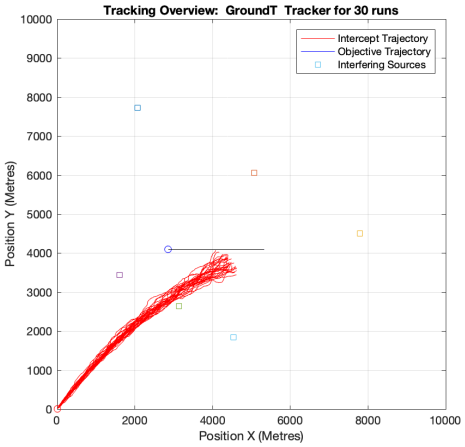


(a) Straight Line

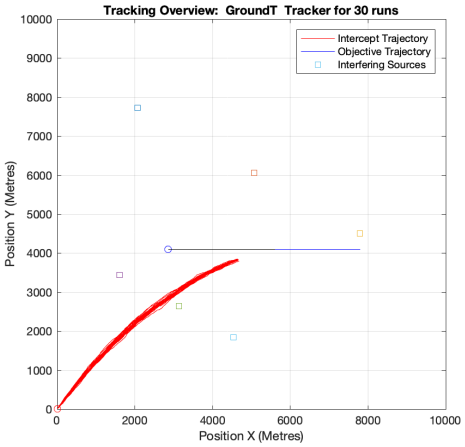


(b) Dubins path

Figure 9: Target track (black) and optimal trajectories (red) for 30 runs using RRT* trajectory planning for the scenario in Figure 1 using an SMC-MHT tracker



(a) Straight Line



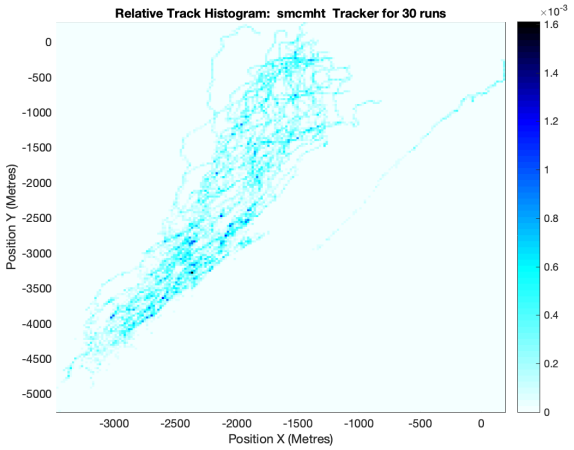
(b) Dubins path

Figure 10: Target track (black) and optimal trajectories (red) for 30 runs using RRT* trajectory planning for the scenario in Figure 1 using the Ground Truth tracker

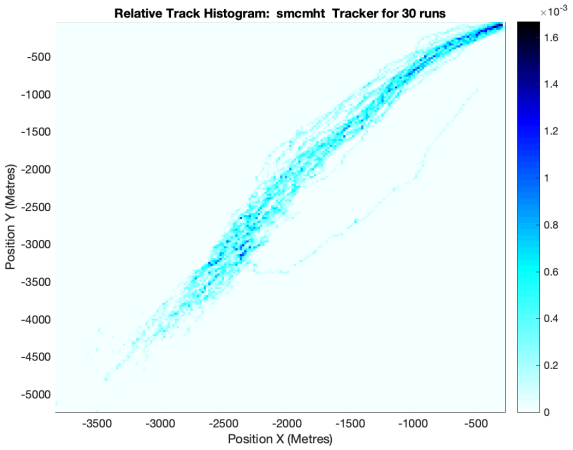
[8] Reid, D. B., "An algorithm for tracking multiple targets.," *IEEE Transactions on Automatic Control* **24**(6), 843–854 (1979).

[9] Boissonnat, J.-D. and Bui, X.-N., "Accessibility region for a car that only moves forwards along optimal paths," Tech. Rep. 2181, INRIA, BP93, 06902 SOPHIA-ANTIPOLIS (1994).

[10] [*22nd International Conference on Information Fusion*] (2019).

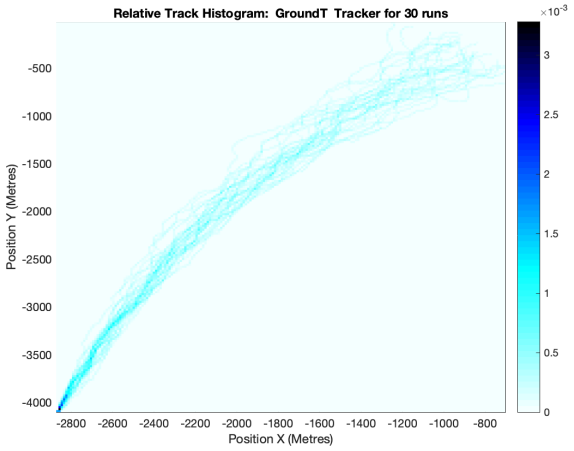


(a) Straight Line

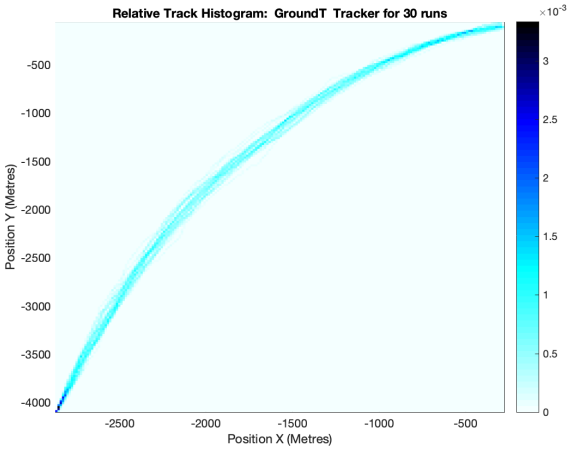


(b) Dubins path

Figure 11: Probabilty of relative observer-target track using RRT* trajectory planning for the scenario in Figure 1 using an SMC-MHT tracker



(a) Straight Line



(b) Dubins path

Figure 12: Probabilty of relative observer-target track using RRT* trajectory planning for the scenario in Figure 1 using the Ground Truth tracker