

Parameterized Trajectories for Target Localization Using Small and Micro Unmanned Aerial Vehicles

Jeffrey B. Corbets
Jack W. Langelaan

The Pennsylvania State University, University Park, PA 16802, USA

Abstract—This paper presents an approach to near-optimal target localization for small and micro uninhabited aerial vehicles using a family of pre-computed parameterized trajectories. These trajectories are pre-computed for a set of nominal target locations uniformly distributed over the sensor field of view and stored offline as a sequence of non-dimensional waypoints. Upon target detection, a trajectory corresponding to the nearest nominal target location is selected and dimensionalized. An onboard waypoint-navigating controller follows the waypoints. Thus, trajectory generation occurs in near-constant time, which allows for fast adaptation as the target state estimate is refined. Parameterization of the trajectories with respect to relative vehicle speed, sensor range, and sensor update rate allows the same table to be used for various combinations of sensor package and vehicle or vehicle operating conditions.

I. INTRODUCTION

Because of dynamic observability the path followed by the vehicle has an enormous effect on the quality of state estimates[1] and optimal trajectory generation for SLAM and the related problem of target tracking has become an active area of research[2], [3], [4].

Computing the optimal trajectory for a realistic vehicle model and realistic sensor models can become computationally prohibitive, and simplified models are generally used. For example, vehicle dynamics have been modeled as a point mass with velocity and acceleration constraints[4] and sensor models have been linearized[5]. Solution methods including dynamic programming[6] and direct collocation[7] have been used to generate the optimal trajectories. These techniques still require fairly powerful computers and depending on the complexity of the model (e.g. field of view constraints also increase complexity) may not be suitable for real-time operation.

This research is focused on real-time target state estimation using computing hardware that could reasonably be installed on a μ AV or airborne munition. To achieve this a family of optimal trajectories is pre-computed for targets located at varying ranges and bearings from the vehicle. When a target is detected the vehicle chooses the appropriate trajectory from the trajectory lookup table and begins to follow it. New trajectories can be selected as the target state estimate is refined, allowing for fast adaptation.

The use of a lookup table rather than on-line trajectory generation assumes that memory is cheaper than computation. Given the ubiquity of very high-capacity flash memory (1+GB USB thumb drives are common), this appears to be a

good assumption. Because the trajectories are pre-computed, realistic vehicle models, sensor models, and constraints can be used in the optimization process. This paper considers a 2D target localization task using a kinematic model for a nonholonomic vehicle. A bearings-only sensor (i.e. a camera) provides measurements to the target and observer vehicle position is assumed to be known precisely.

In our previous work, the trajectory optimization was performed over a sequence of turn rate commands[8]. In this work, the optimization is performed over a sequence of waypoints in a space non-dimensionalized with respect to sensor parameters and relative vehicle speed. This has two main advantages: first, trajectories are not specific to a particular camera or vehicle; second, low-level control can be performed using the waypoint following controller that is typically present on small and micro uninhabited air vehicles.

The remainder of this paper is organized as follows. Section II describes the mission scenario, defines vehicle and sensor models and the non-dimensionalization step, and defines the optimization problem. Section III describes the resulting look up table and how it is implemented in practice. Section IV compares trajectories obtained from the table with those obtained by direct optimization and presents results of Monte Carlo simulations of a target localization task. Finally, Section V presents concluding remarks.

II. PROBLEM FORMULATION

The problem of a μ AV or autonomous submunition performing a surveillance and target tracking task is considered. An on-board vision system (e.g. a monocular camera) obtains bearing measurements to the target. Vehicle position is assumed to be known precisely. A schematic of a sequential target localization task (where the sequence of targets to be visited is determined a priori) is shown in Figure 1.

Vehicle and target positions are denoted \mathbf{x}_v and $\mathbf{x}_1 \dots \mathbf{x}_n$, respectively, in an inertial frame O . The vision system obtains a bearing γ to the target (in the vehicle body frame B). An estimation algorithm (implemented using a Sigma Point Kalman Filter[9], [10]) uses knowledge of vehicle position and the bearing measurements to compute an estimate of target position.

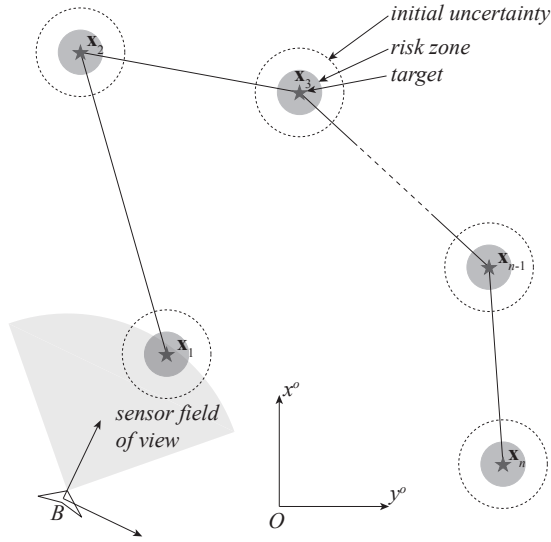


Fig. 1. Schematic of target localization task showing a sequence of targets to be tracked.

A. Sensor and Vehicle Models

The vision system obtains a bearing to the target:

$$\gamma = \arctan\left(\frac{y_t - y_v}{x_t - x_v}\right) - \psi_v + \nu \quad (1)$$

where x_t, y_t represent the location of the stationary target in the 2D plane; x_v, y_v, ψ_v represent the vehicle position and heading; and ν is uncorrelated zero-mean Gaussian random noise covariance Σ_ν . Maximum sensor range is R and the sensor field of view is limited to $-\gamma_{max} \leq \gamma \leq \gamma_{max}$. The sensor frame rate is T_f . The parameters R and T_f will be used to non-dimensionalize the trajectories.

It is assumed that an on-board guidance controller is able to follow a splined path constructed from a series of waypoints. It is assumed that the path can be well approximated by a series of ten waypoints. Following the splined path, the controller provides velocity and turn rate commands, leading to a kinematic vehicle model. Velocity is assumed to be constant:

$$\dot{x}_v = v \cos \psi_v \quad (2)$$

$$\dot{y}_v = v \sin \psi_v \quad (3)$$

$$\dot{\psi}_v = u \quad (4)$$

where u is a commanded turn rate from the waypoint-following controller.

To non-dimensionalize the vehicle kinematics with respect to sensor parameters distances are scaled by sensor range R and time is scaled by the sensor frame sample time T_f :

$$\dot{\tilde{x}}_v = \frac{T_f}{R} v \cos \psi_v \quad (5)$$

$$\dot{\tilde{y}}_v = \frac{T_f}{R} v \sin \psi_v \quad (6)$$

$$\dot{\tilde{\psi}}_v = T_f u \quad (7)$$

A second order approximation is used to generate a discrete time model for vehicle kinematics with sample time T_s . Note that the integration time is also scaled by the sensor frame sample time (i.e. $\tilde{\Delta}t = T_s/T_f$):

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + \frac{T_s}{T_f} \begin{bmatrix} \frac{T_f}{R} v \cos \psi \\ \frac{T_f}{R} v \sin \psi \\ T_f u \end{bmatrix} + \frac{1}{2} \frac{T_s^2}{T_f^2} \begin{bmatrix} -\frac{T_f^2}{R} v u \sin \psi \\ \frac{T_f^2}{R} v u \cos \psi \\ 0 \end{bmatrix} \quad (8)$$

The non-dimensionalized discrete time vehicle kinematics are therefore

$$\tilde{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k + T_s \begin{bmatrix} \frac{v}{R} \cos \psi \\ \frac{v}{R} \sin \psi \\ u \end{bmatrix} + \frac{T_s^2}{2} \begin{bmatrix} -\frac{v}{R} u \sin \psi \\ \frac{v}{R} u \cos \psi \\ 0 \end{bmatrix} \quad (9)$$

Here we shall assume that an on-board controller is capable of following waypoint commands, so that

$$\tilde{\mathbf{x}}_k = f(\tilde{\mathbf{X}}, T_s \frac{v}{R}, k) \quad (10)$$

where $\tilde{\mathbf{X}}$ is a sequence of waypoints expressed in the non-dimensionalized space. The function f is dependent on the particular implementation of waypoint controller, here we assume f is a spline interpolation function.

B. Target State Estimation

The target is assumed to be stationary, hence $\mathbf{x}_{t,k+1} = \mathbf{x}_{t,k}$. The bearing model given in Equation 1 results in a non-linear estimation problem, and the algorithm for a Sigma Point Kalman Filter (i.e. an Unscented Kalman Filter) given in van der Merwe and Wan[9] is used to compute the target state estimate.

C. Trajectory Generation

Typically, small and micro UAVs are equipped with a waypoint following controller. Here we will compute a sequence of waypoints in polar coordinates relative to the observer vehicle which will optimize the information gained about a particular target.

Using the vehicle and sensor models a set of optimal trajectories can be generated for targets lying within the sector defined by the field of view. The field of view is discretized into cells and an optimal trajectory will be generated for a target located at each cell corner.

Let $\tilde{\mathbf{X}}_{mn} = [\tilde{\mathbf{x}}_{mn,1}, \dots, \tilde{\mathbf{x}}_{mn,N}]$ be a sequence of N waypoints in non-dimensional space associated with a trajectory designed to localize a target in cell (m, n) . The trajectory table consists of optimal trajectories

$$\tilde{\mathbf{X}}_{mn}^* = \arg \min J(\tilde{\mathbf{X}}_{mn}) \quad (11)$$

where J is the cost function to be minimized.

The cost function includes a term related to the uncertainty in the target state estimate with a dynamic weight to assist the optimizer in keeping the target within the field of view of the observer vehicle:

$$J = (w_{info} + w_{fov}) J_{info} \quad (12)$$

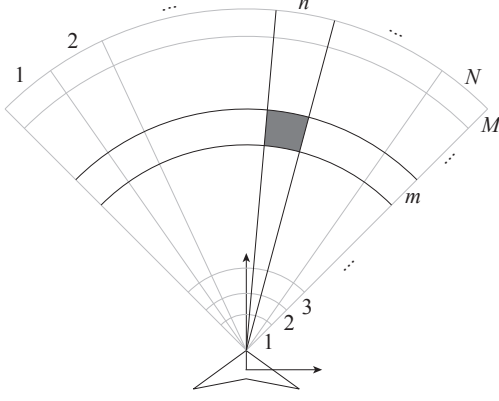


Fig. 2. Discretization of the sensor field of view.

1) *Information Cost*: The information cost is obtained by computing a prediction of the information that can be gained about the target by following a particular trajectory. This is computed using the Fisher Information Matrix (FIM).

Consider a discrete time system with trivial dynamics and non-linear measurement model:

$$\mathbf{x}_{k+1} = \mathbf{x}_k \quad (13)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (14)$$

where \mathbf{v}_k is uncorrelated zero-mean Gaussian random noise.

The FIM for the estimation problem associated with this system can be computed recursively

$$\mathbf{Y}_k = \mathbf{Y}_{k-1} + \mathbf{H}_k^T \Sigma_v^{-1} \mathbf{H}_k \quad (15)$$

where \mathbf{H}_k is the Jacobian of the measurement model evaluated at time k , i.e.

$$\mathbf{H}_k = \frac{\delta}{\delta \mathbf{x}} h(\mathbf{x}_k) \quad (16)$$

For the vision model given by Equation 1 the Jacobian of the sensor model with respect to the estimate of the target is

$$\mathbf{H}_k = \begin{bmatrix} -\frac{\sin \gamma_k}{r_k} & \frac{\cos \gamma_k}{r_k} \end{bmatrix} \quad (17)$$

where $r_k = \sqrt{(x_t - x_{v,k})^2 + (y_t - y_{v,k})^2}$. Non-dimensionalizing with respect to the sensor range gives

$$\tilde{\mathbf{H}}_k = \begin{bmatrix} -\frac{R \sin \gamma_k}{r_k} & \frac{R \cos \gamma_k}{r_k} \end{bmatrix} \quad (18)$$

The non-dimensionalized information gained about the target from a single measurement can now be expressed as

$$\tilde{\mathbf{Y}}_k = \tilde{\mathbf{Y}}_{k-1} + \tilde{\mathbf{H}}_k^T \Sigma_\nu^{-1} \tilde{\mathbf{H}}_k \quad (19)$$

For a single bearing measurement to a single target, $\Sigma_\nu = \sigma_\nu^2$. Expanding gives

$$\tilde{\mathbf{Y}}_k = \tilde{\mathbf{Y}}_{k-1} + \frac{R^2}{r_k^2 \sigma_\nu^2} \begin{bmatrix} \sin^2 \gamma_k & -\sin \gamma_k \cos \gamma_k \\ -\sin \gamma_k \cos \gamma_k & \cos^2 \gamma_k \end{bmatrix} \quad (20)$$

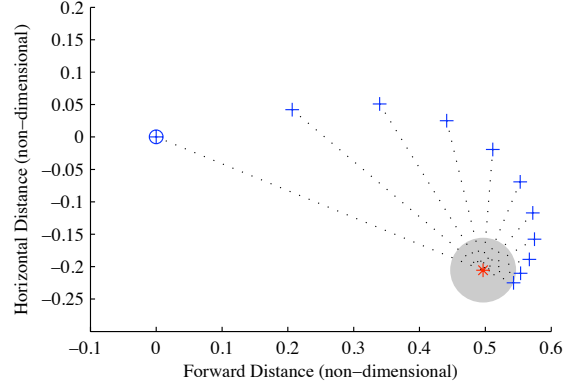


Fig. 3. Waypoints are defined by a distance r from the target location at fixed angles.

Writing Equation 20 as $\tilde{\mathbf{Y}}_k = \tilde{\mathbf{Y}}_{k-1} + \Delta \tilde{\mathbf{Y}}_k$, the information gained about a target over a trajectory can be expressed as

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{Y}}_0 + \sum_{k=1}^K \Delta \tilde{\mathbf{Y}}_k \quad (21)$$

Note that the target is assumed to be stationary. The information cost is

$$J_{info} = \log \det \tilde{\mathbf{Y}}^{-1} \quad (22)$$

$$J_{info} = -\log \det \tilde{\mathbf{Y}} \quad (23)$$

2) *Field of View Weight*: To keep the target in the field of view, a weight is computed based on the bearing to the target, γ_k :

$$w_{fov} = \left(\frac{\gamma_k}{\gamma_{max}} \right)^4 \quad (24)$$

3) *Parameterization of Trajectories*: The trajectories exist in non-dimensional space as a sequence of ten waypoints, $\tilde{\mathbf{X}}_{mn} = [\tilde{x}_{mn,1}, \dots, \tilde{x}_{mn,N}]$. Each waypoint, $\tilde{x}_{mn,N}$, consists of an angle, θ , and a distance, r , to the waypoint relative to the nominal target location. Thus, relative to the target location, each waypoint exists at Cartesian coordinates

$$x_k = r_k \cos \theta_k \quad (25)$$

$$y_k = r_k \sin \theta_k \quad (26)$$

where

$$\theta_k = [0 \dots \pi] \quad (27)$$

$$r_k = r(\theta) \quad (28)$$

For this work, each waypoint exists at a fixed angle from the target location and the distance of the waypoint from the target is allowed to vary, subject to constraints. This is shown in Figure 3.

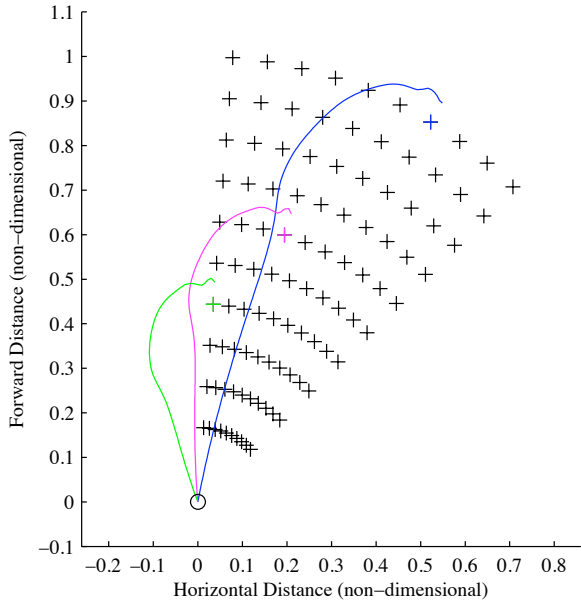


Fig. 4. Sample trajectory table target locations and sample trajectories.

4) *Optimization:* The trajectory generation problem for target (m, n) can now be summarized:

$$\text{minimize} \quad J(\tilde{\mathbf{X}}_{mn}) \quad (29)$$

$$\text{subject to} \quad \tilde{\mathbf{x}}_k = f(\tilde{\mathbf{X}}_{mn}, T_s \frac{v}{R}, k) \quad (30)$$

$$\frac{b}{R} \leq \frac{r_k}{R} \leq 2 \quad (31)$$

$$\frac{R\dot{\psi}_{min}}{v} \leq \tilde{\kappa}_k \leq \frac{R\dot{\psi}_{max}}{v} \quad (32)$$

where the cost J is given by Equations 12, 23 and 24; vehicle path is computed using the interpolating function f . The optimization is bounded by not allowing an optimized waypoint to exist within the safety zone of radius b centered at the target location. The optimization is further bounded by not allowing a waypoint to exist at more than twice the distance between the initial position of the observer vehicle and the target. Finally, the optimization is constrained by not allowing the curvature of the path, $\tilde{\kappa}_k$, non-dimensionalized using sensor range, to exceed a curvature permitted by the vehicle turn rate limits. The resulting vector optimization problem, with w_{info} equal to unity, is solved using a Newton Minimization method[11].

III. THE TRAJECTORY TABLE

To generate a target localization table, the sensor field of view is uniformly discretized in the radial and angular directions: i.e. a 10×10 polar grid is defined over the sensor field of view and a nominal target location is defined at each grid point. A schematic of a target localization table is shown in Figure 4. The figure shows nominal target locations and three sample trajectories.

Each trajectory consists of a sequence of ten waypoints in non-dimensional space. To compute a path in physical space

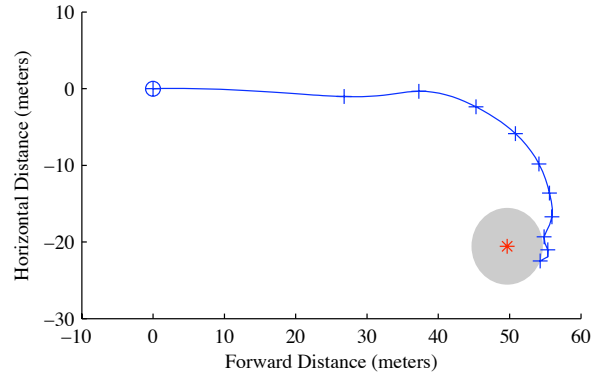


Fig. 5. Ten waypoints are interpolated and dimensionalized to form a complete path.

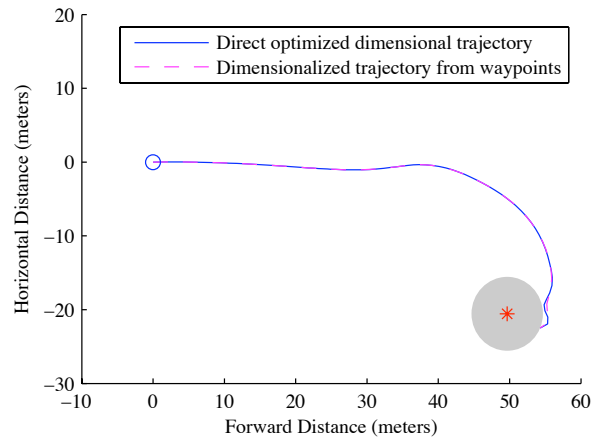


Fig. 6. Direct optimization in dimensional space and using waypoints from a lookup table yield essentially the same path.

the waypoints are first dimensionalized by multiplying by sensor range

$$\mathbf{X}_{mn} = R\tilde{\mathbf{X}}_{mn} = \begin{bmatrix} \mathbf{x}_{mn,1} & \mathbf{x}_{mn,2} & \dots & \mathbf{x}_{mn,10} \end{bmatrix} \quad (33)$$

Finally, a cubic spline is used to compute the complete path, as shown in Figure 5.

Figure 6 shows that the trajectory created by dimensionalizing a set of ten waypoints stored in the lookup table and then interpolating generates essentially the same trajectory as a direct optimization in dimensional space. For practical purposes, the trajectories are the same because the small variation in paths near the target occur when the target is outside the field of view of the observer vehicle's sensor package.

Only half of the sensor field of view is covered by the sample trajectory table shown in Figure 4. Early results from the optimization routine showed good reflection symmetry about the longitudinal axis of the vehicle. The size of the lookup table can be made 50% smaller by exploiting this reflection symmetry. Targets in the left half plane of the observer vehicle are localized with a reflected trajectory from

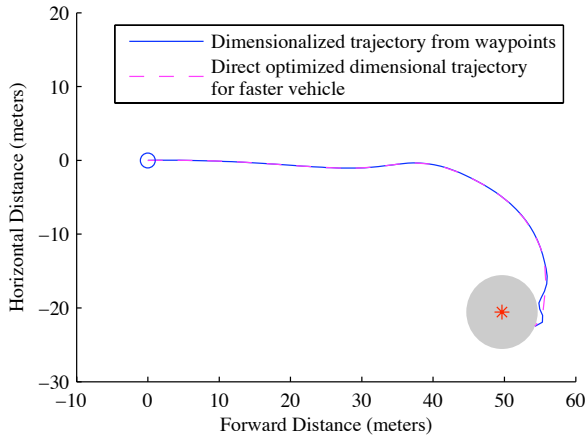


Fig. 7. Comparison of a directly optimized path versus the trajectory table for different “characteristic numbers”.

the lookup table.

A. Observer Vehicle and Sensor Variations

Because the trajectory is stored as a series of waypoints, a trajectory following controller allows vehicles of different speeds to use the single trajectory stored in the lookup table to best localize a target. Since the table was generated by non-dimensionalizing the target localization problem using sensor range R , sensor update period T_f and relative vehicle speed \hat{v} , intuition suggests (and simulations show) that trajectories stored in the lookup table are optimal for any vehicle and sensor package that has the same “characteristic number” as the generated trajectory table. This “characteristic number” relates the sensor range and update rate, R and T_f to the relative vehicle speed, \hat{v} . It is important to note that the relative vehicle speed, \hat{v} , is affected by both the observer vehicle speed and speed of the target, if moving. Additionally, other outside factors, such as wind, can also affect the relative vehicle speed.

$$CN = \frac{(R/T_f)}{\hat{v}} \quad (34)$$

Intuitively, CN is the maximum number of measurements of the target that can be obtained while the path is flown.

Simulations have further shown that observer setups having different CN from that of the table (e.g. due to a difference in speed) can still localize a given target well using the trajectory from the lookup table. Because the trajectories are stored as non-dimensional waypoints, an observer vehicle setup with a sensor of any range can dimensionalize the trajectories to a valid real space path. Figure 7 shows a path that was optimized within dimensional space for a vehicle and sensor package with “characteristic number” CN twice as large as the CN of the trajectory table compared with the path from the trajectory table. Simulation results have shown that the difference in cost is less than 0.5% when a path (i.e. sequence of waypoints) obtained from the trajectory table is flown.

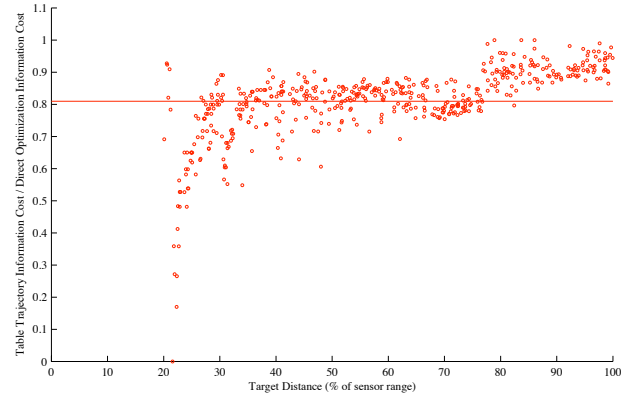


Fig. 8. Comparison of the cost for trajectories from the lookup table versus a direct optimized trajectory for 500 random target locations.

B. Adaptation

As the vehicle follows a trajectory selection from the table, target state estimation occurs in real time. By selecting a new trajectory from the table when the estimated target state has changed, the target localization trajectory can be made adaptive to changes. Currently, a particular trajectory is followed for some control horizon T_c and a new trajectory is selected at the end of the control horizon.

IV. SIMULATION RESULTS

A series of simulations was conducted to evaluate the performance of the trajectory table versus a direct optimization. For the simulations, a target was placed at a random location within the field of view of the sensor. The trajectory corresponding to the nearest nominal target location was chosen and the observer aircraft flew and took measurements along the entire trajectory. This was then compared to a trajectory optimized for the actual (random) target location. The ratio of cost from the lookup table trajectory to the cost from a direct optimized trajectory versus target distance is shown in Figure 8. On average, a trajectory from the lookup table results in 81.0% the cost as the direct (true) optimized trajectory for a random target placed in the sensor field of view.

When comparing information gain alone, which is directly representative of the target localization performance of the trajectory, the table lookup method performs extremely well. For the 500 simulations, the lookup table method provides, on average, 90.0% the information gain as the true optimized path. This is because the field of view weight in the optimized cost function, while necessary for the optimizer to find a solution, does not affect the information gain in real-life. As can be seen in Figure 9, in some cases, the table lookup method performs significantly (greater than 20%) better than the directly optimized path. In these cases, small differences in the trajectory result in the observer vehicle obtaining a small increase in number of measurement locations where the target is within the field of view of the sensor.

The table trajectory can also be retrieved from memory and dimensionalized online much faster than an optimization

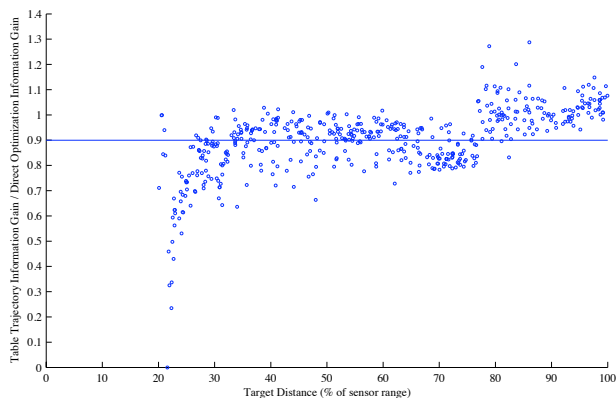


Fig. 9. Comparison of the information gain alone for trajectories from the lookup table versus a direct optimized trajectory for 500 random target locations.

can take place. Times for generating a sequence of ten waypoints in dimensional space for a waypoint-following controller are given in Table I and shows that, on average, the table lookup occurs more than 130 times faster than the online optimization. It should be noted that the median optimization time of 1.28 seconds represents flight over 10-15% of the sensor range for a nominal μAV . Thus, trajectories computed online are likely to be obsolete before they can be flown.

TABLE I

COMPARISON OF CPU TIMES FOR GENERATING A DIMENSIONAL TRAJECTORY FROM THE LOOKUP TABLE VERSUS ONLINE OPTIMIZATION USING A 2.6 GHZ AMD OPTERON PROCESSOR.

	Lookup Table	Online Optimization
Minimum	0.0084 s	0.4822 s
Maximum	0.0403 s	2.5074 s
Median	0.0084 s	1.2779 s
Mean	0.0089 s	1.2364 s

The trajectory table can also be used in a sequential target localization task. The order of targets to be visited and an initial estimate of position is determined by a human operator or higher level planner and sent to the vehicle. The vehicle chooses an appropriate trajectory from the table for the first target and flies the path. When a terminal condition for the target is reached (e.g. the covariance of the target state estimate has reached a certain value, or the safety zone for the target has been reached), the next target is selected from the list and the process repeats.

If a target is outside sensor range, the vehicle is commanded to turn towards the initial assumed position of the target and fly straight towards that point until the target enters the field of view. A trajectory can then be selected from the table. If the target does not enter the field of view, the observer vehicle will fly a search path of expanding circles until a target is located or the search area is cleared.

V. CONCLUSION

This paper has presented a method of target localization based on selecting optimal parameterized trajectories from a lookup table. A trajectory is represented as a sequence of waypoints non-dimensionalized with respect to sensor range, and way-point positions are computed off-line for a particular choice of vehicle speed, sensor update period and sensor range. A characteristic number CN is defined and provides an indication of the number of measurements of target state that can be obtained.

Simulation results show that the path obtained from the lookup table is identical (within numerical accuracy) to that obtained by direct optimization when the characteristic numbers are the same. When the characteristic numbers are different (e.g. due to a difference in speed) the paths differ slightly, but the information gained about the target is nearly identical.

This parameterized approach to trajectory generation for optimal target localization significantly reduces the real-time computational load on a small or micro UAV's processor, freeing capacity for other tasks such as state estimation, navigation, or communication. This method of generating near-optimal trajectories could also be used to provide better estimates of "cost-to-go" in dynamic programming problems and other methods of online optimization in observer vehicles where enough computation power is available.

REFERENCES

- [1] S. Huang, N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang, "Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity," in *IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers. Piscataway, New Jersey: IEEE, April 2005.
- [2] E. W. Frew, "Observer trajectory generation for target-motion estimation using monocular vision," Ph.D. dissertation, Stanford University, Stanford, California, August 2003.
- [3] B. Grocholsky, A. Makarenko, and H. Durrant-Whyte, "Information theoretic coordinated control of multiple sensor platforms," in *IEEE International Conference on Robotics and Automation*, vol. 1. Piscataway, New Jersey: IEEE, September 2003, pp. 1521-1526.
- [4] J. Ousingsawat and M. E. Campbell, "Optimal cooperative reconnaissance using multiple vehicles," *Journal of Guidance, Control and Dynamics*, vol. 30, no. 1, pp. 122-132, January 2007.
- [5] A. J. Sinclair, R. J. Prazenica, and D. E. Jeffcoat, "Simultaneous localization and planning for cooperative air munitions," in *International Conference on Cooperative Control*, 2007.
- [6] J. Nygards, P. Skoglar, J. Karlholm, R. Bjorstrom, and M. Ulvklo, "Towards concurrent sensor and path planning," FOI, Tech. Rep., May 2005.
- [7] B. R. Geiger, J. F. Horn, A. Delullo, L. N. Long, and A. F. Neissner, "Optimal path planning of uavs using direct collocation with nonlinear programming," in *AIAA Guidance, Navigation and Control Conference*, Keystone, Colorado, 2006.
- [8] J. B. Corbets and J. W. Langelaan, "Parameterized optimal trajectory generation for target localization," in *AIAA Guidance, Navigation and Control Conference*, Hilton Head, South Carolina, August 2007.
- [9] R. van der Merwe and E. Wan, "The square-root unscented Kalman filter for state and parameter-estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. Salt Lake City, UT: IEEE, 2001.
- [10] R. van der Merwe, E. Wan, and S. Julier, "Sigma point Kalman filters for nonlinear estimation and sensor fusion: Applications to integrated navigation," in *AIAA Guidance, Navigation and Controls Conference*. Providence, RI USA: AIAA, August 2004.
- [11] E. K. P. Chong and S. H. Zak, *Introduction to Optimization*, 3rd ed. John Wiley & Sons, 2008.