

Parameterized Optimal Trajectory Generation for Target Localization

Jeffrey B. Corbets*

Jack W. Langelaan†

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

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 off-line. Upon target detection the vehicle chooses the trajectory corresponding to the closest nominal target location. Adaptation is enabled with the ability to select new trajectories as the target state estimate is updated. Simulation results show the validity of this approach for both single target and sequential target localization missions. Further, results show that very coarse trajectory tables give the same or better target localization performance as finely discretized tables.

I. Introduction

Unmanned aerial vehicles (UAVs) have played an integral role in the air defenses of many nations since their earliest inception, as drones for air target practice. With the development of the Global Positioning System (GPS), fast digital computers, and digital vision systems, the capabilities of UAVs have increased tremendously. Predator and Global Hawk UAVs are used almost continuously for surveillance purposes, and the air forces of many countries have announced orders and design contracts for many new UAVs. The role of the UAV within an air force is to perform jobs of the type of one of the three Ds - jobs that are either too dangerous, too dirty, or too dull for onboard human pilots. This paper will focus on trajectory algorithms for jobs of the latter two categories, dirty and dull. Dirty jobs are those in which the environment of the feature of interest is contaminated in some way. Dull jobs are those that are repetitive or monotonous, such as station keeping. For these two types of jobs, the UAV has the potential to be automated, with the flight of the vehicle from takeoff to landing conducted without intervention from a pilot in a ground station.

Recent interest in very small UAVs or micro air vehicles (μ AVs) has led to research in target tracking and state estimation using very limited sensor suites¹ (for example a low-cost vision system and inertial measurement unit, or IMU). Fusing bearing data from the vision system with accelerations and angular rates from the IMU enables estimation of both vehicle state and target state: this is known as Simultaneous Localization and Mapping (SLAM). SLAM has been an extremely active area of research in the mobile robotics community for the past two decades²⁻⁴ with applications to ground,⁵ air,⁶ and sea⁷-based robots. In most cases both bearings and ranges to targets are available, the bearings-only case considered here adds challenges stemming from the reduced observability of states. Bearings-only SLAM is characterized by *dynamic observability*, meaning that observability evolves over time as measurements are collected from geometrically dispersed locations.^a

*Research Assistant, Department of Aerospace Engineering, Student Member AIAA

†Assistant Professor, Department of Aerospace Engineering, Senior Member AIAA.

Copyright © 2007 by Jeffrey B. Corbets and Jack W. Langelaan. Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

^aStrictly speaking SLAM estimation algorithms are all unobservable: they attempt to estimate the global locations of the vehicle and landmarks simultaneously using only relative measurements and vehicle odometry. However, strict observability can be regained with the availability of sporadic GPS updates. Continuous availability of GPS simplifies the SLAM problem significantly, resulting in a target tracking or mapping problem.

Because of dynamic observability the path followed by the vehicle has an enormous effect on the quality of state estimates⁸ and optimal trajectory generation for SLAM and the related problem of target tracking has become an active area of research.^{9–11}

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 constraints¹¹ and sensor models have been linearized.¹² Solution methods including dynamic programming¹³ and direct collocation¹⁴ 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 (1GB 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 vehicle position is assumed to be known precisely.

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 optimal trajectory table and discusses a method for compressing the table. Section IV describes a sample mission consisting of sequential target localization and finally Section V presents conclusions and areas for future work.

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.

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^{15,16}) uses knowledge of vehicle position and the bearing measurements to compute an estimate of target position.

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 low-level controller is able to follow velocity and heading 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 command input.

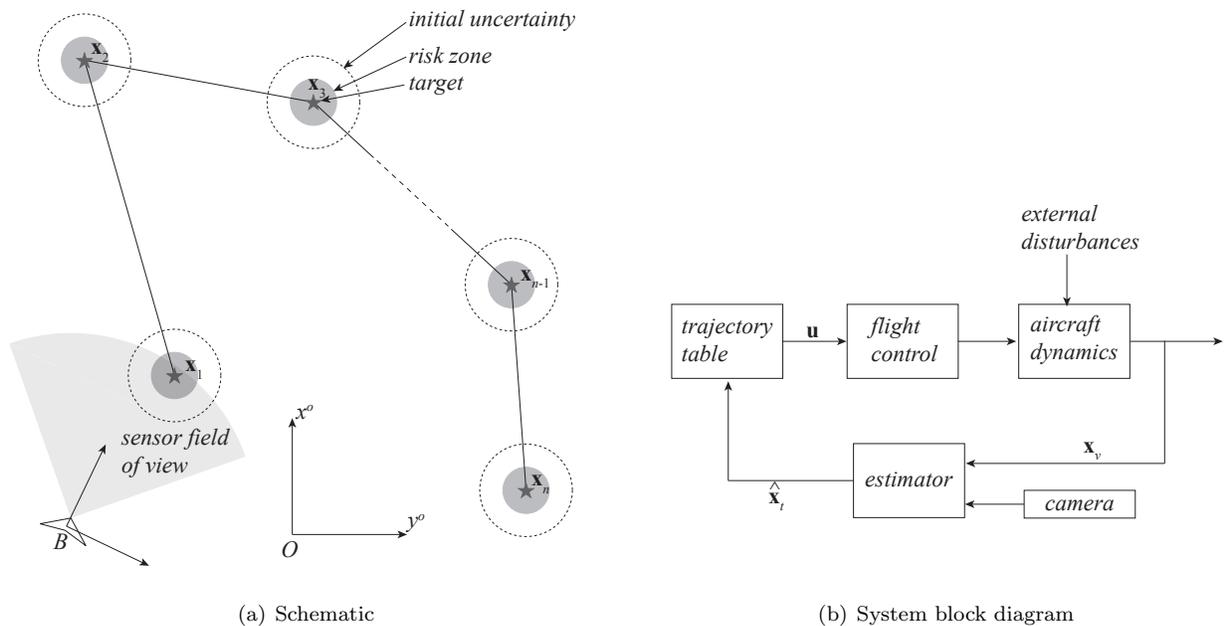


Figure 1. Left: Schematic of target localization task showing a sequence of targets to be tracked; Right: block diagram of system.

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 vehicle kinematics are therefore

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

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 Wan¹⁵ is used to compute the target state estimate.

C. Trajectory Generation

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 $\mathbf{u}_{mn} = [u_0, \dots, u_K]$ be a sequence of heading rate commands associated with a trajectory designed to localize a target in cell (m, n) . The trajectory table consists of optimal trajectories

$$\mathbf{u}_{mn}^* = \arg \min J(\mathbf{u}_{mn}) \quad (10)$$

where J is the cost function to be minimized.

The cost function includes a term related to the uncertainty in the target state estimate, a ‘‘safety’’ term which defines a barrier function around the target and a term to keep the target within the field of view:

$$J = w_{info} J_{info} + w_{safe} J_{safe} + w_{fov} J_{fov} \quad (11)$$

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:

$$\mathbf{x}_{k+1} = \mathbf{x}_k \quad (12)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (13)$$

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

$$\mathbf{Y}_k = \mathbf{Y}_{k-1} + \mathbf{H}_k^T \Sigma_v^{-1} \mathbf{H}_k \quad (14)$$

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

$$\mathbf{H}_k = \frac{\delta}{\delta \mathbf{x}} h(\mathbf{x}_k) \quad (15)$$

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

$$\mathbf{H}_k = \begin{bmatrix} -\frac{\sin \gamma_k}{r_k} & \frac{\cos \gamma_k}{r_k} \end{bmatrix} \quad (16)$$

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

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

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

$$\tilde{\mathbf{Y}}_k = \tilde{\mathbf{Y}}_{k-1} + \tilde{\mathbf{H}}_k^T \Sigma_\nu^{-1} \tilde{\mathbf{H}}_k \quad (18)$$

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

$$\tilde{\mathbf{Y}}_k = \tilde{\mathbf{Y}}_{k-1} + \frac{R^2}{r_k^2 \sigma_\nu^2} \begin{bmatrix} \sin^2 \gamma_k & -\sin \gamma_k \cos \gamma_k \\ -\sin \gamma_k \cos \gamma_k & \cos^2 \gamma_k \end{bmatrix} \quad (19)$$

Writing Equation 19 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

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{Y}}_0 + \sum_{k=1}^K \Delta \tilde{\mathbf{Y}}_k \quad (20)$$

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

$$J_{info} = \log \det \tilde{\mathbf{Y}}^{-1} \quad (21)$$

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

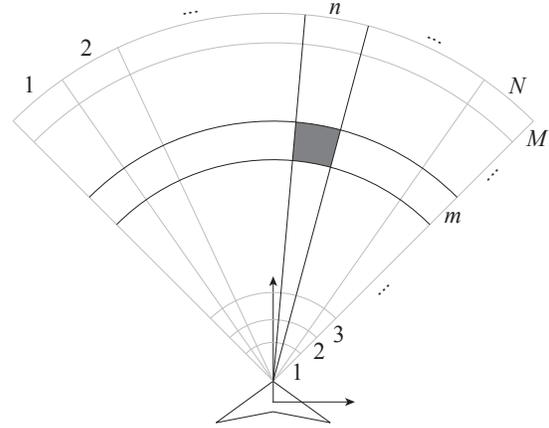


Figure 2. Discretization of the sensor field of view.

2. Safety Cost

A safety zone is included around the target to ensure that sufficient stand-off distance is maintained to prevent collision with the target or detection of the vehicle. A simple high-order polynomial function is used to ensure high cost close to the target and low cost far away.

$$J_{safe} = \sum_{k=1}^K \left(\frac{r_{safe}}{r_k} \right)^8 \quad (23)$$

For cases where the goal consists of impact with the target (e.g. for an autonomous munition) the safety zone may represent the distance at which a terminal guidance algorithm assumes control of the vehicle.

3. Field of View Cost

To keep the target in the field of view a cost is computed based on the bearing to the target:

$$J_{fov} = \sum_{k=1}^K \gamma_k^2 \quad (24)$$

4. Optimization

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

$$\text{minimize} \quad J(\mathbf{u}_{mn}) \quad (25)$$

$$\text{subject to} \quad \tilde{\mathbf{x}}_{k+1} = f(\tilde{\mathbf{x}}_k, u_{k,mn}) \quad (26)$$

$$u_{min} \leq u_{k,mn} \leq u_{max} \quad (27)$$

where the cost J is given by Equations 11, 22, 23 and 24; vehicle kinematics are given by Equation 9; and inputs u_k are limited by turn rate constraints. In principle any optimization method could be used to generate a solution. In this work the trajectory is discretized into K steps with constant turn rate in each step. The resulting vector optimization problem is solved using MATLAB's `fmincon` function. All weights in the cost function (i.e. w_{info} , w_{safe} , and w_{fov}) were set to 1.

III. The Trajectory Table

To generate a target localization table the sensor field of view was uniformly discretized in the radial and angular directions: i.e. a 10×10 polar grid was defined over the sensor field of view and a nominal target location was defined at each grid point. A schematic of a target localization table is shown in Figure 3. The figure shows nominal target locations and three sample trajectories.

Each trajectory consists of a sequence of turn rate commands

$$\mathbf{u}_{mn} = \begin{bmatrix} u_{mn,1} & u_{mn,2} & \dots & u_{mn,K} \end{bmatrix} \quad (28)$$

where K is a planning horizon that extends from $t = t_0$ until the time the risk zone is encountered.

Only half of the sensor field of view is covered by the sample trajectory table shown in Figure 3. 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 the lookup table.

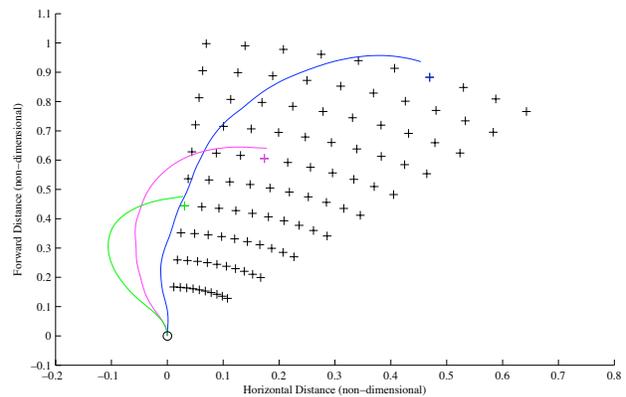


Figure 3. Sample trajectory table target locations and sample trajectories.

A. Target Localization

Target localization using the lookup table follows three steps:

1. An initial target location is passed to the table. If it is within the field of view, the trajectory \mathbf{u}_{mn} associated with the closed cell centroid is selected. If the initial target location is outside the field of view the vehicle is commanded to turn towards the target and fly until the target is seen.
2. The trajectory \mathbf{u}_{mn} is followed in open loop over a control horizon T_c . A target state estimate is computed using a Sigma Point Kalman Filter.^{15,16} The control horizon is dependent on the initial distance between the vehicle and the target at the time of the first camera measurement. When the control horizon is reached, a new trajectory is selected based on the current estimate of the target position.
3. The task is complete when the vehicle reaches the risk zone or the vehicle passes the target. The next target in the sequence is selected, and the process repeats for all given targets.

A sequence of images showing localization of a single target is given in Figure 4. The target is initially within the field of view of the sensor, so a trajectory can be selected immediately. The vehicle follows the trajectory while estimating target state for a control horizon $T_c = 2$ seconds, hence 4.1 seconds into the flight, shown in subfigure (b), the vehicle is following its third trajectory. The vehicle continues to fly, selecting new trajectories as the control horizon is reached. The target localization task is concluded as the vehicle passes the target. Recall that sensor noise is assumed to be Gaussian, here $\sigma_\nu = 0.0175$ rad (i.e. 1°) was used.

B. Compression by Decimation

Because the lookup table is designed to be used on vehicles that are very small and might even be destroyed when the mission is complete (e.g. autonomous munitions), minimizing the amount of memory required to store the table is desired. In addition to compressing the table through reflective symmetry, density of nominal target locations in the trajectory table has a large effect on the memory required to store the table.

A simple study showing the effects of the number of trajectories stored in the lookup table on the localization error was performed. Two lookup tables, one with 100 trajectories and one with 400 trajectories, were created directly. A third lookup table, with 25 trajectories, was created by post-processing the 100 trajectory lookup table: every second row and every second column was deleted from the table.^b A Monte Carlo simulation was then performed for each of the three lookup tables. Random target locations within the sensor field of view were chosen and the vehicle flew the target localization algorithm as described previously. Target state estimation was performed on-line

The results of the Monte Carlo simulation, with 500 runs for each lookup table, are given in Table 1. The table shows values for the determinant, trace, and maximum eigenvalue of the covariance Σ of the *true target localization error* over the 500 run Monte Carlo simulation. The results show the 5×5 trajectory lookup table actually performed the best job localizing the random single targets, with the 10×10 and 20×20 trajectory tables performing slightly worse. A scatter plot of the target localization error for all three Monte Carlo simulations is shown in Figure 5. Finally, a plot of the weighted error for each simulation run for each trajectory lookup table is given in Figure 6. The weighted (i.e. normalized) error e_i for each run is computed from

$$e_i = \sqrt{\frac{1}{2}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^T \mathbf{P}^{-1}(\mathbf{x}_i - \hat{\mathbf{x}}_i)} \quad (29)$$

where \mathbf{x}_i is the true target position for run i , $\hat{\mathbf{x}}_i$ is the estimated target position at the end of the run and \mathbf{P} is the estimated covariance. The factor $1/2$ accounts for the dimension of the estimation problem (2D target position estimation), and for a consistent estimator the average value of e_i is unity. This is the case for all three trajectory lookup tables.

These results suggest that (in this implementation) there is very little difference between the optimal target localization trajectory and a trajectory that is sub-optimal but still “pretty good”. In the 5×5

^bThis is compression by decimation, and is an easy but very crude method of data compression. More sophisticated approaches which compute average trajectories over neighboring nominal target positions may allow greater compression with less loss of accuracy.

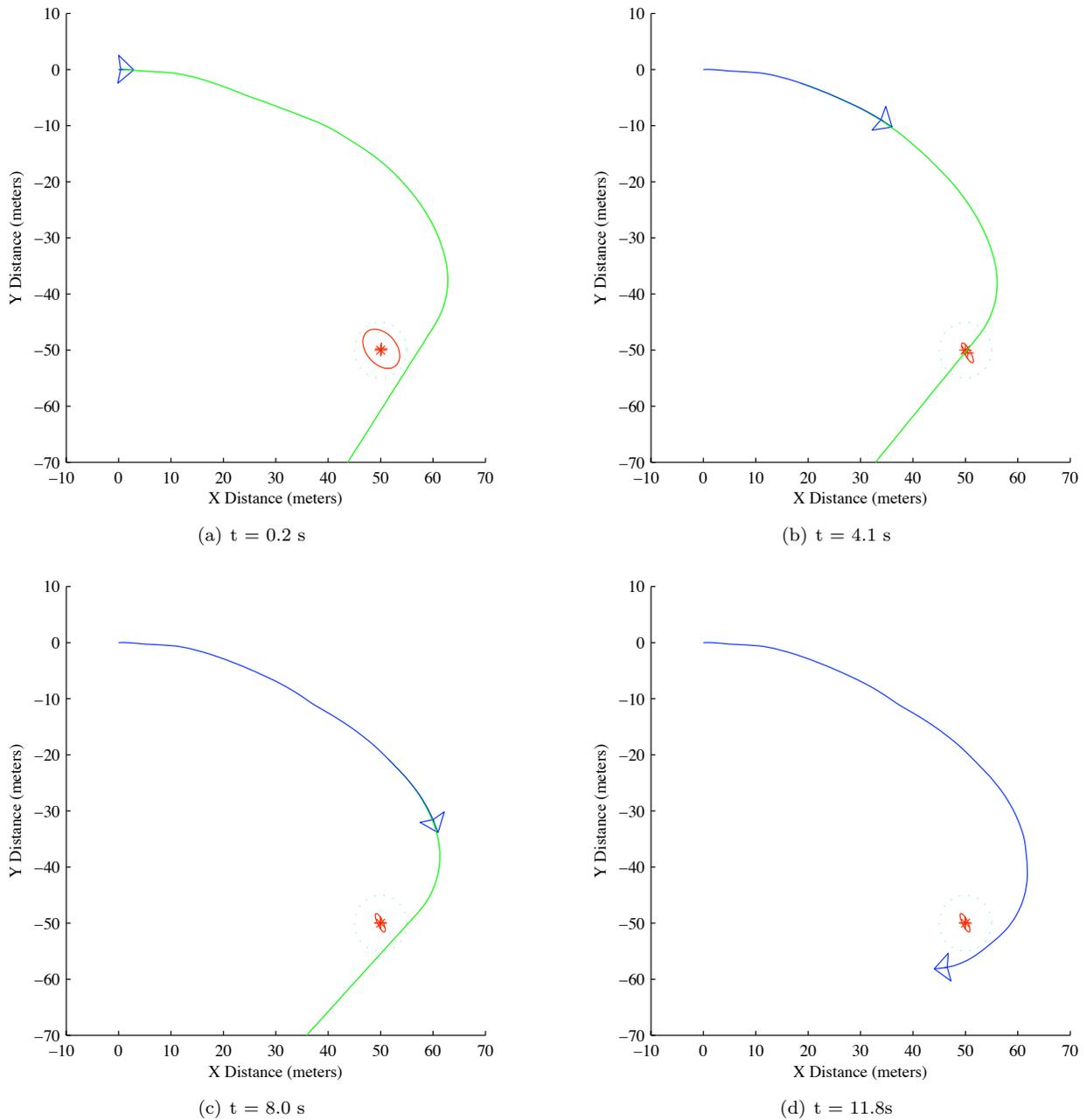


Figure 4. Snapshots of a single target localization run using the trajectory table. True target location shown with *, estimated target location shown with + and associated error ellipse. The green line shows the current path selected from the trajectory table, blue line shows path flow.

Table 1. Effect of trajectory table compression on target localization accuracy.

table size	$\det \Sigma$	$\text{Tr} \Sigma$	$\max(\text{eig} \Sigma)$
5×5	0.0186	0.4330	0.3846
10×10	0.0254	0.4818	0.4217
20×20	0.0281	0.4815	0.4136

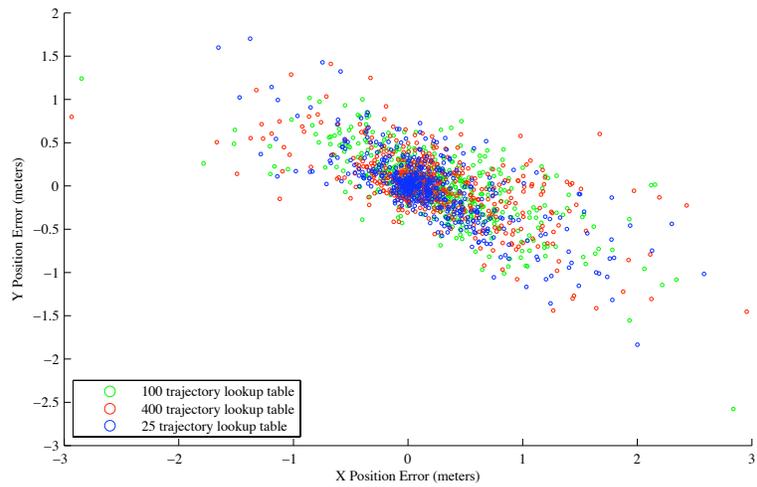
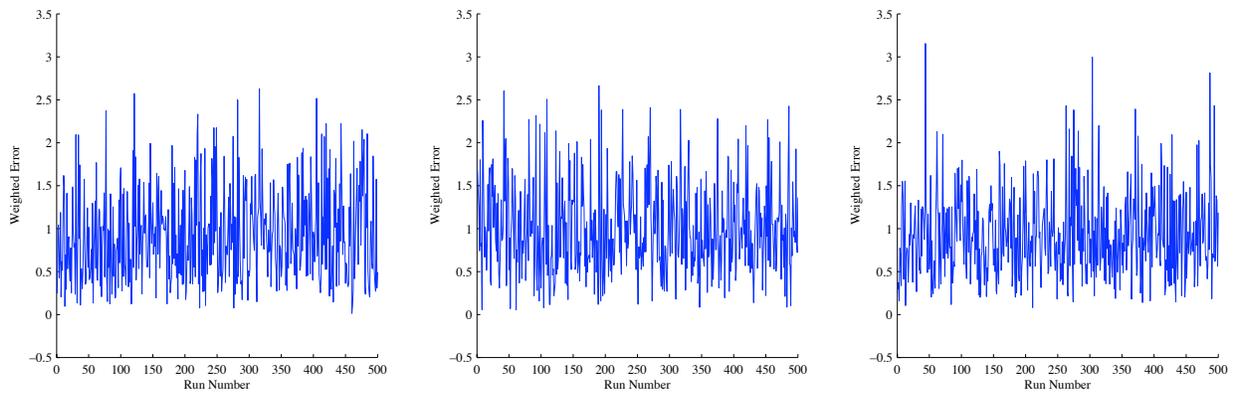


Figure 5. Scatter plot of target localization error for 500 run Monte Carlo simulation.



(a) 5×5 lookup table

(b) 10×10 lookup table

(c) 20×20 lookup table

Figure 6. Weighted (i.e. normalized) error for 500 simulations with different size lookup tables.

trajectory table the actual target position will generally be farther away from the nominal target position used to generate the trajectory than for either the 10×10 or 20×20 tables. It can thus be inferred that the cost function used in this paper is very “flat” in the vicinity of the optimal value.

C. Observer Vehicle Variations

In the current form, the lookup table returns a set of turn rate inputs and is specific to an observer vehicle. Thus, vehicle speed is a dependent parameter. One way to account for variations in vehicle speed would be to add a new dimension to the lookup table that contains different trajectories based on the observer vehicle speed. Another possible solution would be to post-process the lookup table’s set of control inputs into a trajectory path or a sequence of way-points in non-dimensional space. A lower level controller could then command the observer vehicle to follow the trajectory or way-point sequence in real space. The problem of variations in the observer vehicles is an area for future work.

IV. Sequential Target Localization

A mission for a small or micro UAV may consist of localizing and tracking a sequence of targets provided by a human operator. Here it is assumed that initial uncertain estimates of target locations are available and that the ‘visit sequence’ is determined by the human operator.

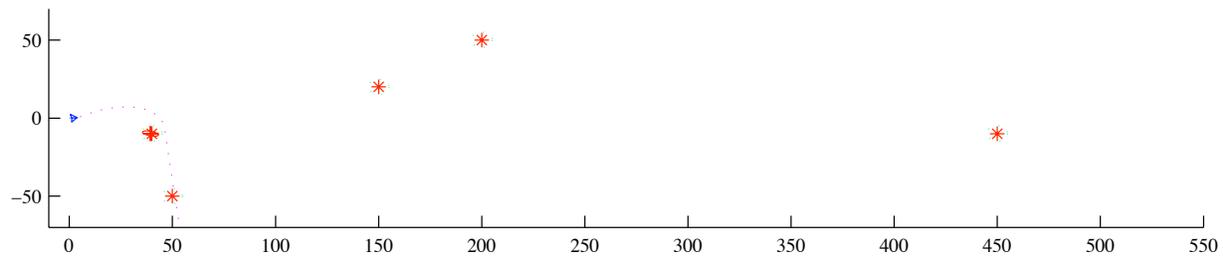
Results of a simulation of this mission are shown in Figure 7. The first subfigure shows the layout of the targets and position of the vehicle immediately after takeoff. The observer vehicle has picked an optimal trajectory out of the lookup table for the first target, shown by the magenta dotted line. The vehicle continues to fly this path for the control horizon before choosing a new trajectory. Subfigure (b) shows the S-shaped trajectory flown between the first and second targets, along with a long optimal trajectory flown to localize the third target. Because the fourth target is not in view, the control system chooses to fly a hard right turn to acquire the target with the sensor system. Once acquired, an optimal trajectory is selected from the lookup table and shown by the magenta dotted line. The third subfigure shows the vehicle after passing the fourth target. Because the fifth target is well outside the sensor range of the vehicle, the control system flies a trajectory to minimize the distance between the observer vehicle and suspected location of the target as quickly as possible. This trajectory is not obtained directly from the trajectory look up table, hence no dotted line is shown. Finally, in subfigure (d), the complete path flown by the observer vehicle is shown. The long straight line between the fourth and fifth targets indicates the time during which the fifth target was outside of the sensor range of the observer. When the sensor does acquire the target, a curved optimal path is flown to localize the fifth target. During the flight, portions of 18 trajectories in the lookup table were flown.

V. Conclusion

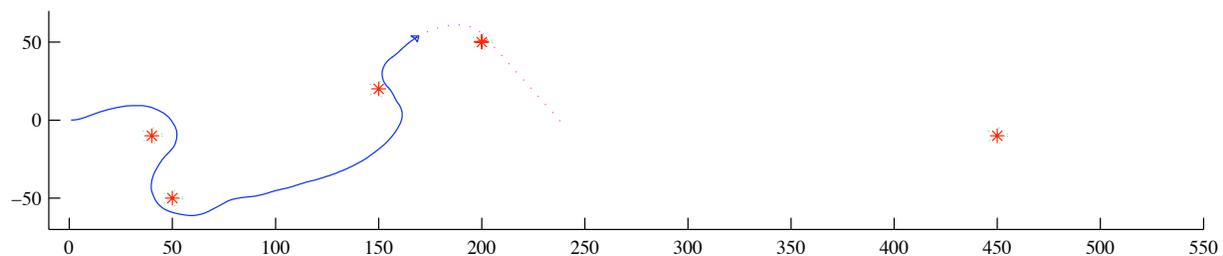
This research has presented an approach bearings-only to target localization by a small or micro uninhabited aerial vehicle which uses a table of pre-computed nominal trajectories. A family of nominal target locations is defined and parameterized into a non-dimensional space with respect to the sensor field of view and sensor frame rate. An optimal trajectory is computed for each nominal target location and stored in a look up table. When the UAV detects a target, its location is transformed into the non-dimensional sensor space and the trajectory corresponding to the nearest nominal target position is selected. The vehicle follows this trajectory over a control horizon and then selects a new trajectory from the table based on the latest estimate of target position.

This research was motivated by the limited computational resources available on small and micro UAVs. The time required to select a trajectory is determined by memory access time rather than computational power, hence more computation can be devoted to tasks such as target state estimation, vehicle control, or higher level tasks.

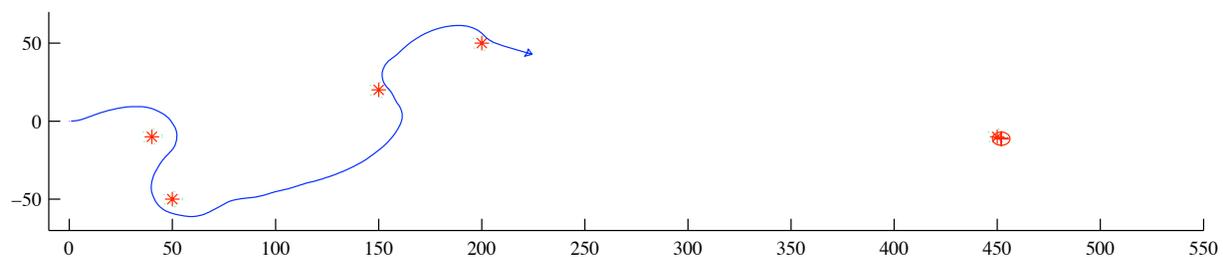
Monte Carlo simulations of target localization tasks were show that this approach can localize targets consistently and within the estimated uncertainty. Results also show that a trajectory lookup table consisting of 25 nominal target locations (distributed uniformly on a grid over the sensor field of view) performs as well or better than lookup tables with 100 and 400 nominal target locations. This suggests that the cost function used in this research is very “flat” in the vicinity of the optimal value and allows significant reduction the



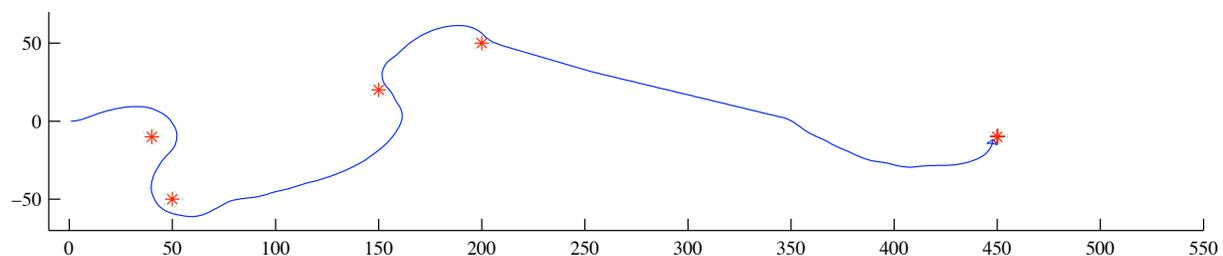
(a) $t = 0.2$ s



(b) $t = 34.7$ s



(c) $t = 41.5$ s



(d) $t = 70.1$ s

Figure 7. Sequential target localization. Blue ‘wings’ show vehicle position, solid blue line shows path flown, dotted line shows path selected from trajectory table. True target position is shown with red *, estimated target position (for the target currently being localized) is shown with + and error ellipse.

memory required to store the data required for trajectory table based plans.

Finally a results of a simulation of a sequential target localization task are presented. The trajectory table based approach is able to localize the targets sequentially by selecting appropriate trajectories from the table.

References

- ¹Langelaan, J. and Rock, S., "Navigation of Small UAVs Operating in Forests," *AIAA Guidance, Navigation and Controls Conference*, American Institute of Aeronautics and Astronautics, Providence, RI USA, August 2004.
- ²Smith, R. C. and Cheeseman, P., "On the respresentation and estimation of spatial uncertainty," *International Journal of Robotic Research*, Vol. 4, No. 5, 1986.
- ³Thrun, S., Burgard, W., and Fox, D., "A Real-Time Algorithm for Mobile Robot Mapping with Applications to Multi-Robot and 3D Mapping," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, San Francisco, CA, 2000.
- ⁴Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B., "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges," *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- ⁵Liu, Y. and Thrun, S., "Results for Outdoor-SLAM using Sparse Extended Information Filters," *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, Taipei, Taiwan, 2003.
- ⁶Kim, J. and Sukkarieh, S., "Autonomous Airborne Navigation in Unknown Environments," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 40, No. 3, July 2004, pp. 1031–1045.
- ⁷Williams, S. B., Dissanayake, G., and Durrant-Whyte, H., "Field Deployment of the Simultaneous Localisation and Mapping Algorithm," *15th IFAC World Congress on Automatic Control*, Barcelona, Spain, June 2002.
- ⁸Huang, S., Kwok, N. M., Dissanayake, G., Ha, Q. P., and Fang, G., "Multi-Step Look-Ahead Trajectory Planning in SLAM: Possibility and Necessity," *IEEE International Conference on Robotics and Automation*, Institute of Electrical and Electronics Engineers, IEEE, Piscataway, New Jersey, April 2005.
- ⁹Frew, E. W., *Observer Trajectory Generation for Target-Motion Estimation Using Monocular Vision*, Ph.D. thesis, Stanford University, Stanford, California, August 2003.
- ¹⁰Grocholsky, B., Makarenko, A., and Durrant-Whyte, H., "Information Theoretic Coordinated Control of Multiple Sensor Platforms," *IEEE International Conference on Robotics and Automation*, Vol. 1, IEEE, Piscataway, New Jersey, September 2003, pp. 1521–1526.
- ¹¹Ousingsawat, J. and Campbell, M. E., "Optimal Cooperative Reconnaissance Using Multiple Vehicles," *Journal of Guidance, Control and Dynamics*, Vol. 30, No. 1, January 2007, pp. 122–132.
- ¹²Sinclair, A. J., Prazenica, R. J., and Jeffcoat, D. E., "Simultaneous Localization and Planning for Cooperative Air Munitions," *International Conference on Cooperative Control*, 2007.
- ¹³Nygards, J., Skoglar, P., Karlholm, J., Bjorstrom, R., and Ulvklø, M., "Towards Concurrent Sensor and Path Planning," Tech. rep., FOI, May 2005.
- ¹⁴Geiger, B. R., Horn, J. F., Delullo, A., Long, L. N., and Neissner, A. F., "Optimal Path Planning of UAVs Using Direct Collocation with Nonlinear Programming," *AIAA Guidance, Navigation and Control Conference*, Keystone, Colorado, 2006.
- ¹⁵van der Merwe, R. and Wan, E., "The Square-Root Unscented Kalman Filter for State and Parameter-Estimation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, Salt Lake City, UT, 2001.
- ¹⁶van der Merwe, R., Wan, E., and Julier, S., "Sigma Point Kalman Filters for Nonlinear Estimation and Sensor Fusion: Applications to Integrated Navigation," *AIAA Guidance, Navigation and Controls Conference*, American Institute of Aeronautics and Astronautics, Providence, RI USA, August 2004.