INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust Nonlinear Control 2004; **14**:741–766 (DOI: 10.1002/rnc.933)

On-line estimation and path planning for multiple vehicles in an uncertain environment

Jarurat Ousingsawat*,† and Mark E. Campbell

Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853-7501, U.S.A.

SUMMARY

A unified approach to cooperative target tracking and path planning for multiple vehicles is presented. All vehicles, friendly and adversarial, are assumed to be aircraft. Unlike the typical target tracking problem that uses the linear state and nonlinear output dynamics, a set of aircraft nonlinear dynamics is used in this work. Target state information is estimated in order to integrate into a path planning framework. The objective is to fly from a start point to a goal in a highly dynamic, uncertain environment with multiple friendly and adversarial vehicles, without collision. The estimation architecture proposed is consistent with most path planning methods. Here, the path planning approach is based on evolutionary computation technique which is then combined with a nonlinear extended set membership filter in order to demonstrate a unified approach. A cooperative estimation approach among friendly vehicles is shown to improve speed and routing of the path. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: target tracking; path planning; nonlinear estimation; cooperative estimation; ellipsoids

1. INTRODUCTION

Multiple vehicle command and control is the subject of great interest in recent years. Tasks, such as battlefield surveillance, mapping and offensive maneuvers, can be completed more quickly and efficiently using multiple vehicles, as compared with a single vehicle. Other benefits of multiple vehicle systems include reduced cost by using smaller, cheaper vehicles and increased redundancy as compared to one large vehicle. Path planning, or maneuvering from one point to another, is a basic building block of these systems; solutions to the path planning problem become increasingly challenging in an uncertain environment. The dominant characteristic of these future systems is the lack of the information about the environment, and lack of information shared between vehicles, which can cause missed objectives, wasted resources and potentially collisions. Therefore, it is necessary to be able to estimate or predict locations,

*Correspondence to: Ms. J. Ousingsawat, Department of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, U.S.A.
†E-mail: jo65@cornell.edu

Contract/grant sponsor: DARPA ITO Grant; contract/grant number: F33615-99-C-3612

Received 31 March 2003
Revised 16 October 2003
Accepted 18 December 2003

capabilities or even intentions of adversaries in the focus environment; combine information from disparate sources; and deliver the appropriate information to blocks such as path planners.

Two primary research areas are addressed in this work. The first is target tracking, defined as the ability to estimate and predict locations of adversaries. There is a large body of research in target tracking. The extended Kalman filter (EKF) is widely used in tracking, such as in Reference [1]. In Reference [2], the interactive multiple model (IMM) filter is developed to track maneuvering targets. In Reference [3], the multiple hypothesis tracking (MHT) is applied to a target formation. A particle filter approach is used to track multiple objects in Reference [4]. The comparison between different nonlinear filters: EKF, unscented Kalman filter (UKF) and particle filter is made in Reference [5]. While these methods have worked well in a variety of applications, there are several limitations for future multiple vehicle system such as: (1) most approaches linearize the dynamics and mask higher order terms using tuned noise parameters; (2) they are stochastic, which does not lend well to current constrained optimal control approaches, such as in Reference [6]; (3) a complete cooperative framework between dynamic vehicles had yet to be developed.

The second research area is path planning, which is currently receiving much attention from the community. Many solution approaches exist in current research of single and multiple vehicle path planning. In Reference [7], Capozzi groups these into the following areas, with an example given for each: Reactive Behaviours [8], Calculus of Variations [9], Graph Search [10], Biologically inspired [11], and Probabilistic Roadmaps [12]. Each of these methodologies have their advantages and disadvantages, but they all require a good tracking model in order to develop a good plan.

The objective of this paper is to develop an estimation architecture for multiple aerial vehicles that enables path planning in an uncertain, realistic environment, as illustrated in Figure 1. The approach here is to use bounded estimation methods, where the linearization error is directly incorporated into the estimation architecture, thus addressing the fundamental limitation in the estimation methods described above. A second objective is to develop a cooperative estimation approach which can be integrated into a wide variety of planners. Specifically, adversarial vehicle positions and uncertainties are tracked using radar and a nonlinear bounded set membership filter, while cooperative vehicle positions are tracked using

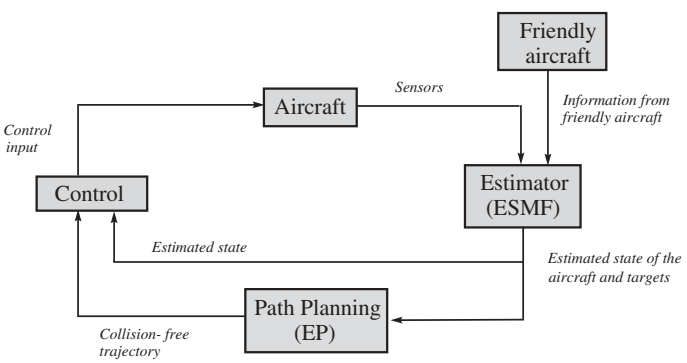

Figure 1. Diagram of the full architecture for cooperative estimation and planning.

GPS like sensors and cross-link communication. This enables cooperative estimation among friendly vehicles by defining an intersecting uncertainty ellipsoid that combines tracking information from more than one vehicle. Uncertainties addressed specifically in this architecture include model errors, model nonlinearities, sensor noise, jamming and radar/communication black-outs.

The estimation architecture developed here is generic to the path planning problem, implying that it can be integrated with many different types of planners. For this work, the estimation approach is integrated with evolutionary computation based path planner in order to demonstrate cooperative planning in a realistic uncertain dynamic environment. The case of dynamic obstacle estimation and path planning is specifically addressed. Because the algorithm must plan the trajectory in advance, state information of obstacles in the near future must be known (especially for moving obstacles). Here, the position and velocity error ellipsoids are used in the path planner as dynamic obstacles. A set membership filter is used in order to bound the uncertainty in the position of each vehicle, which in turn prevents collisions. A diagram of the full architecture is shown in Figure 1.

The paper is presented as follows. First, the nonlinear aircraft model and estimation architecture used in target tracking is introduced. Then, a bounded estimator for nonlinear dynamics, termed the extended set membership filter (ESMF) is presented. A cooperative estimation approach between friendly, dynamic vehicles is introduced. Finally, integration between path planning and estimation is discussed, followed by several simulation examples using a nonlinear F-16 like simulation.

2. AIRCRAFT MODEL AND SENSORS

Typical radar tracking algorithms assume the noise drives a linear point mass model and nonlinear output dynamics (radar), as shown in References [1, 13]. The use of a nonlinear radar measurement results in the necessity of using a nonlinear filter. In this work, a more complex nonlinear aircraft model is used in order to (1) reduce conservatism and (2) develop additional information for planning. Because a nonlinear filter is required due to the output dynamics, the use of a nonlinear aircraft model does not significantly increase the level of the complexity in estimation. The primary question is whether the sensor suite can observe the state vector.

The model consists of two parts. The first part is the state vector related to the aircraft itself, which is referred as $\mathrm{AC}_{\mathrm{F},i}$ for ith friendly (F) aircraft. The second part is the dynamics of the target or adversarial (A) aircraft, which is referred as $\mathrm{AC}_{\mathrm{A},j}$ for jth adversarial aircraft. Although it is assumed that the target is also an aircraft, estimation of the full state of each target aircraft is not necessary because only a subset of the states are required for use in a path planner. Therefore, a subset of the full nonlinear aircraft model is used for target tracking.

The two-dimensional system ($x - h$ plane) is used in order to simplify and gain insight into the problem. The state vector of ith friendly aircraft is given as

$$\mathbf{x}_{\mathrm{F},i} = [x_{\mathrm{F},i}\ h_{\mathrm{F},i}\ \theta_{\mathrm{F},i}\ V_{\mathrm{F},i}\ \alpha_{\mathrm{F},i}\ q_{\mathrm{F},i}]^{\mathrm{T}} \tag{1}$$

where x is the position of an aircraft in the horizontal direction, h the altitude, θ the pitch angle, V the speed, α the angle of attack, and q the pitch rate. The dynamics of aircraft with process

noise (**w**) is as following (details can be found in Reference [21])

$$\left.\begin{aligned}
\dot{x}_{\mathrm{F},i} &= V_{\mathrm{F},i}\cos(\theta_{\mathrm{F},i}-\alpha_{\mathrm{F},i}) + w_{x_{\mathrm{F},i}}\\
\dot{h}_{\mathrm{F},i} &= V_{\mathrm{F},i}\sin(\theta_{\mathrm{F},i}-\alpha_{\mathrm{F},i}) + w_{h_{\mathrm{F},i}}\\
\dot{\theta}_{\mathrm{F},i} &= q_{\mathrm{F},i} + w_{\theta_{\mathrm{F},i}}\\
\dot{V}_{\mathrm{F},i} &= -m_{\mathrm{F},i}g\sin(\theta_{\mathrm{F},i}-\alpha_{\mathrm{F},i}) + \frac{(T+X)_{\mathrm{F},i}\cos\alpha_{\mathrm{F},i}+Z_{\mathrm{F},i}\sin\alpha_{\mathrm{F},i}}{m_{\mathrm{F},i}} + w_{V_{\mathrm{F},i}}\\
\dot{\alpha}_{\mathrm{F},i} &= q_{\mathrm{F},i} + \frac{g}{V_{\mathrm{F},i}}\cos(\theta_{\mathrm{F},i}-\alpha_{\mathrm{F},i}) - \frac{(T+X)_{\mathrm{F},i}\sin\alpha_{\mathrm{F},i}+Z_{\mathrm{F},i}\cos\alpha_{\mathrm{F},i}}{m_{\mathrm{F},i}V_{\mathrm{F},i}} + w_{\alpha_{\mathrm{F},i}}\\
\dot{q}_{\mathrm{F},i} &= \frac{1}{J_{\mathrm{F},i}}[(\bar{Q}S\bar{c})_{\mathrm{F},i}(C_m + \frac{2\bar{c}}{V}C_{mq}q)_{\mathrm{F},i} + Z_{\mathrm{F},i}(x_{\mathrm{ac}}-x_{\mathrm{cg}})_{\mathrm{F},i}] + w_{q_{\mathrm{F},i}}
\end{aligned}\right\} \tag{2}$$

where the aircraft parameters are as following: m is the mass, $\bar{Q}$ the dynamic pressure, S the wing span, $\bar{c}$ the chord length, J the moment of inertia, x_{ac} the aerodynamic centre, x_{cg} the centre of gravity, and T the thrust. In addition, X and Z are forces in x, h direction respectively; and they are functions of angle of attack, elevator deflection, dynamic pressure and wingspan. The variables C_m and C_{mq} are stability derivatives, which are functions of angle of attack and elevator deflection.

A cartesian co-ordinate system for tracking is used here in order to make integration with path planning easier. The pitching angle ($\theta_{\mathrm{A},j}$) and angle of attack ($\alpha_{\mathrm{A},j}$) are not particularly useful information in a path planner individually; however the difference, which is the flight path angle ($\gamma_{\mathrm{A},j} = \theta_{\mathrm{A},j} - \alpha_{\mathrm{A},j}$), gives information on the direction to which the target is heading. The velocity of the target is also important in a path planner. Following the dynamics of an aircraft (Equation (2)), the adversary dynamics are described as

$$\left.\begin{aligned}
\dot{x}_{\mathrm{A},j} &= V_{\mathrm{A},j}\cos(\gamma_{\mathrm{A},j}) + w_{x_{\mathrm{A},j}}\\
\dot{h}_{\mathrm{A},j} &= V_{\mathrm{A},j}\sin(\gamma_{\mathrm{A},j}) + w_{h_{\mathrm{A},j}}\\
\dot{\gamma}_{\mathrm{A},j} &= \frac{g}{V_{\mathrm{A},j}}\cos(\gamma_{\mathrm{A},j}) + \frac{1}{V_{\mathrm{A},j}}\underbrace{\left(\frac{-(T+X)_{\mathrm{A},j}\sin(\alpha_{\mathrm{A},j})+Z_{\mathrm{A},j}\cos(\alpha_{\mathrm{A},j})}{m_{\mathrm{A},j}}\right)}_{F1_j} + w_{\gamma_{\mathrm{A},j}}\\
\dot{V}_{\mathrm{A},j} &= -m_{\mathrm{A},j}g\sin(\gamma_{\mathrm{A},j}) + \underbrace{\frac{(T+X)_{\mathrm{A},j}\cos(\alpha_{\mathrm{A},j})+Z_{\mathrm{A},j}\sin(\alpha_{\mathrm{A},j})}{m_{\mathrm{A},j}}}_{F2_j} + w_{V_{\mathrm{A},j}}
\end{aligned}\right\} \tag{3}$$

The quantities $F1_j$ and $F2_j$, which represent the normalized force in the inertial co-ordinate system, are unknown. Here, the inertial forces are grouped together and estimated using parameter identification within the estimation framework. The parameters, $F1_j$ and $F2_j$, can be added to the estimator by modelling them as a random walk, or

$$\dot{F1}_j = w_{F1_j} \quad \text{and} \quad \dot{F2}_j = w_{F2_j}$$

where w_{F1_j} and w_{F2_j} are random noise with a range that is appropriate for the magnitude of the forces $F1_j$ and $F2_j$, yet also enable enough sensitivity in order to track the forces. The state

vector of jth adversarial aircraft is then given as

$$\mathbf{x}_{\mathrm{A},j} = [x_{\mathrm{A},j}\ h_{\mathrm{A},j}\ \gamma_{\mathrm{A},j}\ V_{\mathrm{A},j}\ F1_j\ F2_j]^{\mathrm{T}} \tag{4}$$

Therefore, the complete model that is implemented for estimation of a single friendly and a single adversarial vehicle can be described as

$$\mathbf{x} = [\mathbf{x}_{\mathrm{F},i}\ \mathbf{x}_{\mathrm{A},j}]^{\mathrm{T}} \tag{5}$$

where $\mathbf{x}_{\mathrm{F},i}$ and $\mathbf{x}_{\mathrm{A},j}$ are given by Equations (1) and (4), respectively.

For case of multiple vehicles, the states of additional adversaries can be added in a similar manner. This structure allows vehicles to enter and exit environment without having to modify the full system architecture. Note that full state information of other friendly aircraft, $\mathbf{x}_{\mathrm{F},k\neq i}$, can be provided through communication link between friendly aircraft.

2.1. Nominal sensor suite

It is assumed that all states of the nominal aircraft are available through noisy sensors. The target is detected using a radar measurement that provides data in polar co-ordinates: range (R_{ij}) and azimuth angle ($\theta_{R,ij}$) with measurement noise, $v_{R,i}$ and $v_{\theta_R,i}$, respectively. The radar measurement is illustrated in Figure 2 and can be written as

$$R_{ij}(k) = \sqrt{(x_{\mathrm{A},j}(k) - x_{\mathrm{F},i}(k))^2 + (h_{\mathrm{A},j}(k) - h_{\mathrm{F},i}(k))^2} + v_{R,i}$$

$$\theta_{R,ij}(k) = \tan^{-1}\frac{h_{\mathrm{A},j}(k) - h_{\mathrm{F},i}(k)}{x_{\mathrm{A},j}(k) - x_{\mathrm{F},i}(k)} + v_{\theta_R,i}$$

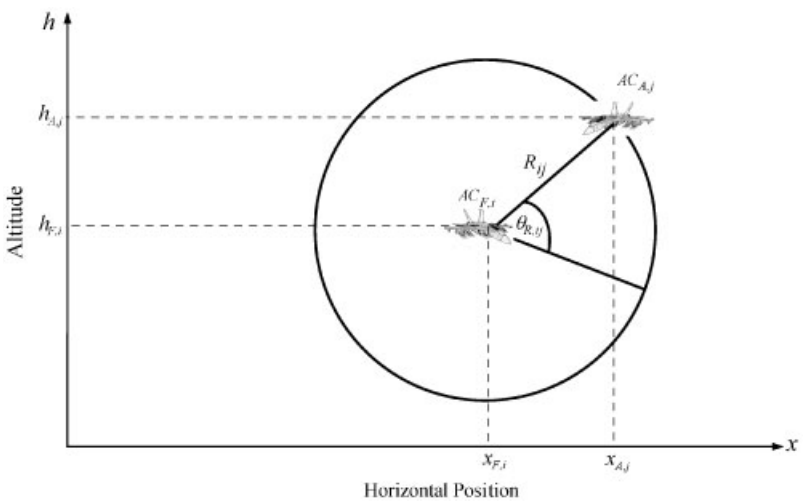

Figure 2. Description of the parameters of the radar.

where sensor noise are bounded by an ellipsoid:

$$\{v_{R,i} \;\; v_{\theta_R,i}\} \begin{bmatrix} \sigma^2_{R,i} & 0 \\ 0 & \sigma^2_{\theta_R,i} \end{bmatrix}^{-1} \begin{Bmatrix} v_{R,i} \\ v_{\theta_R,i} \end{Bmatrix} \leqslant 1$$

Although the radar measurement provides enough data to indicate the current position of the target, it is not enough to observe all of the other states in Equation (5). The heading angle and velocity play a major role in predicting the near future location of the target, which is valuable information in path planning. For this reason, bounded measurements are added to make the system observable.

2.2. *Bounded heading angle*

Because the measurement equations are nonlinear, the propagation of uncertainty bounds is not simple. In Reference [13], a linearization is made to compute the variance of azimuth and elevation angles in the cartesian filtering. The position of $\mathrm{AC}_{\mathrm{A},j}$ at times $k-1$ and k are known to be bounded using ellipsoids, as described in Figure 3. In order to approximate the heading angle, the uncertainty ellipsoid at time $k-1$ is obtained from the estimator at the previous time step, while the uncertainty ellipsoid at time k is determined from the radar measurement at the current time. Here, consecutive measurements of the target location provide an approximation of the direction to which the target is heading, and are utilized as a bounded measurement of the heading angle ($\gamma_{\mathrm{A},j}$). Consider Figure 3, which shows a graphic description of the approximate heading angle ($\tilde{\gamma}_{\mathrm{A},j}$). The details on how to compute $\mathscr{E}\left(\begin{Bmatrix} \tilde{x}_{\mathrm{A},j}(k) \\ \tilde{h}_{\mathrm{A},j}(k) \end{Bmatrix}, \tilde{P}(k-1)\right)$ can be found in the appendix.

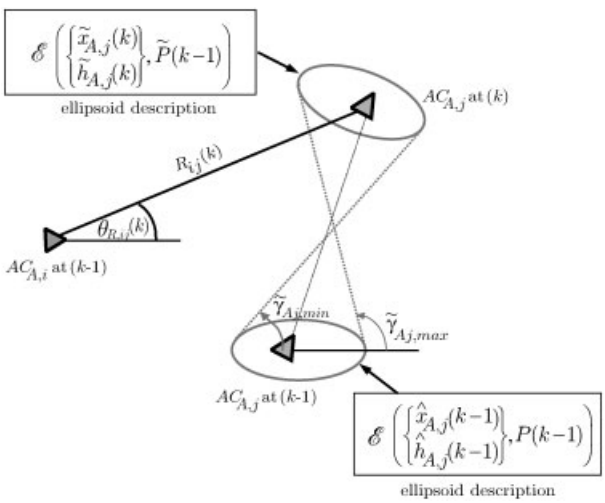

Figure 3. Geometric description of approximate heading angle.

The heading $\tilde{\gamma}_{\mathrm{A},j}$ is then bounded between $\tilde{\gamma}_{\mathrm{A}\,j,\min}$ and $\tilde{\gamma}_{\mathrm{A}\,j,\max}$, which depend on orientations and sizes of uncertainty ellipsoids. This is written as

$$\tilde{\gamma}_{\mathrm{A}\,j,\max}(k) = \tan^{-1}\left(\frac{\tilde{h}_{\mathrm{A},j}(k) - b_k \sin(\beta_k) - \hat{h}_{\mathrm{A},j}(k-1) - b_{k-1}\sin(\beta_k)}{\tilde{x}_{\mathrm{A},j}(k) - b_k \cos(\beta_k) - \hat{x}_{\mathrm{A},j}(k-1) - b_{k-1}\cos(\beta_k)}\right)$$

$$\tilde{\gamma}_{\mathrm{A}\,j,\min}(k) = \tan^{-1}\left(\frac{\tilde{h}_{\mathrm{A},j}(k) + b_k \sin(\beta_k) - \hat{h}_{\mathrm{A},j}(k-1) + b_{k-1}\sin(\beta_k)}{\tilde{x}_{\mathrm{A},j}(k) + b_k \cos(\beta_k) - \hat{x}_{\mathrm{A},j}(k-1) + b_{k-1}\cos(\beta_k)}\right)$$

where $\tilde{x}_{\mathrm{A},j}(k) = R_{ij}(k)\cos(\theta_{R,ij}(k) + \theta_{\mathrm{F},i}(k)) + \hat{x}_{\mathrm{F},i}(k)$, $\tilde{h}_{\mathrm{A},j}(k) = R_{ij}(k)\sin(\theta_{R,ij}(k) + \theta_{\mathrm{F},i}(k)) + \hat{h}_{\mathrm{F},i}(k)$, b_k is the square root of the maximum eigenvalue of $\tilde{P}(k)$, β_k is the orientation of the uncertainty ellipsoid at time k, $\hat{x}_{\mathrm{A},j}(k-1)$ is estimated $x_{\mathrm{A},j}$ from the previous step, $\hat{h}_{\mathrm{A},j}(k-1)$ is estimated $h_{\mathrm{A},j}$ from the previous step, b_{k-1} is the square root of the maximum eigenvalue of $P(k-1)$, β_{k-1} is the orientation of the uncertainty ellipsoid at time $k-1$.

The bounded heading angle is then computed as

$$\tilde{\gamma}_{\mathrm{A},j}(k) = \frac{\tilde{\gamma}_{\mathrm{A}\,j,\max}(k) + \tilde{\gamma}_{\mathrm{A}\,j,\min}(k)}{2}$$

$$\sigma_{\tilde{\gamma}_{\mathrm{A}\,j}}(k) = \frac{\tilde{\gamma}_{\mathrm{A}\,j,\max}(k) - \tilde{\gamma}_{\mathrm{A}\,j,\min}(k)}{2}$$

and used as a measurement in the estimation architecture.

2.3. Bounded velocity

The velocity is approximated in an approach very similar to the heading angle. Using the target location at consecutive times. The maximum velocity is determined from the longest distance between two ellipsoids in Figure 3, and minimum velocity is found using shortest distance. The longest and shortest distance can be computed as

$$\Delta x_{\max}(k) = \Delta x(k) + \sqrt{P_{xx}(k-1)} + \sqrt{\tilde{P}_{xx}(k-1)}$$

$$\Delta h_{\max}(k) = \Delta h(k) + \sqrt{P_{hh}(k-1)} + \sqrt{\tilde{P}_{hh}(k-1)}$$

$$\Delta x_{\min}(k) = \begin{cases} 0 & \text{if } \Delta x_{\max}(k) < 2\left(\sqrt{\tilde{P}_{xx}(k-1)} + \sqrt{P_{xx}(k)}\right) \\ \Delta x(k) - \sqrt{P_{xx}(k-1)} - \sqrt{\tilde{P}_{xx}(k-1)} & \text{otherwise} \end{cases}$$

$$\Delta h_{\min}(k) = \begin{cases} 0 & \text{if } \Delta h_{\max}(k) < 2\left(\sqrt{\tilde{P}_{hh}(k-1)} + \sqrt{P_{hh}(k)}\right) \\ \Delta h(k) - \sqrt{P_{hh}(k-1)} - \sqrt{\tilde{P}_{hh}(k-1)} & \text{otherwise} \end{cases}$$

where $\Delta x(k) = \tilde{x}_{\mathrm{A},j}(k) - \hat{x}_{\mathrm{A},j}(k-1)$, $\Delta h(k) = \tilde{h}_{\mathrm{A},j}(k) - \hat{h}_{\mathrm{A},j}(k-1)$, and P_{xx}, P_{hh} are the (1,1) and (2,2) entries of P.

It follows that the maximum and minimum velocity are:

$$\tilde{V}_{\mathrm{A}\,j,\max}(k) = \frac{\sqrt{\Delta x_{\max}(k)^2 + \Delta h_{\max}(k)^2}}{\Delta t}$$

$$\tilde{V}_{\mathrm{A}\,j,\min}(k) = \frac{\sqrt{\Delta x_{\min}(k)^2 + \Delta h_{\min}(k)^2}}{\Delta t}$$

The bounded velocity can then be easily computed as

$$\tilde{V}_{\mathrm{A},\,j}(k) = \frac{\tilde{V}_{\mathrm{A}\,j,\max}(k) + \tilde{V}_{\mathrm{A}\,j,\min}(k)}{2}$$

$$\sigma_{\tilde{V}_{\mathrm{A}\,j}}(k) = \frac{\tilde{V}_{\mathrm{A}\,j,\max}(k) - \tilde{V}_{\mathrm{A}\,j,\min}(k)}{2}$$

3. EXTENDED SET MEMBERSHIP FILTER (ESMF)

Because the dynamics and measurement are nonlinear, the estimation architecture requires the nonlinear filter. Instead of EKF, as most applications in this area use, the ESMF developed in Reference [14] is used in this work. The advantage of the set membership method is that it bounds the estimates, including the linearization error, assuming bounded noise. The disadvantage is that the estimates can be conservative. It is assumed here that path planning and collision avoidance [15] are safety critical applications, and thus the use of a conservative, bounded estimation approach is a valid one.

The ESMF is a nonlinear recursive estimation approach analogous to the EKF which uses bounded sets rather than probabilistic distributions. The ESMF filter linearizes the model about each time step, as in the EKF. The difference is that the remainder of the linearization is bounded and added as an uncertainty. This guarantees that the estimate falls within the uncertain sets.

The ESMF delivers an 'ellipsoid set', $\mathscr{E}(\hat{\mathbf{x}}, P)$ given as

$$\mathscr{E}(\hat{\mathbf{x}}, P) = \{\mathbf{x} \mid [\mathbf{x} - \hat{\mathbf{x}}]^{\mathrm{T}} P^{-1} [\mathbf{x} - \hat{\mathbf{x}}] \leqslant 1\}$$

in which the state lies within an ellipsoid at each time step. The ellipsoid set, $\mathscr{E}(\hat{\mathbf{x}}, P)$, is analogous to the mean and covariance of the EKF. The differences are that in the ESMF, the state (a) is guaranteed to lie within the ellipsoid, assuming the noise are bounded, and (b) has uniform probability to lie anywhere within the set (i.e. not a higher probability at the centre as in the EKF). The bounded ellipsoid matches well with path planning approaches, as most of these methods consider hard circles and ellipsoids as obstacles.

Consider a discrete nonlinear system written as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k \tag{6}$$

$$\mathbf{y}_{k+1} = h(\mathbf{x}_{k+1}) + \mathbf{v}_{k+1} \tag{7}$$

where $\mathbf{w}_k$ and $\mathbf{v}_{k+1}$ are process and sensor noise. They are bounded by ellipsoids: $[\mathbf{w}_k]^{\mathrm{T}} Q_k^{-1}[\mathbf{w}_k] \leqslant 1$ and $[\mathbf{v}_{k+1}]^{\mathrm{T}} R_{k+1}^{-1}[\mathbf{v}_{k+1}] \leqslant 1$. Similarly, the initial state is also bounded: $[\mathbf{x}_0 - \hat{\mathbf{x}}_0]^{\mathrm{T}} \Sigma_{0,0}^{-1}[\mathbf{x}_0 - \hat{\mathbf{x}}_0] \leqslant 1$. The matrices Q_k, R_{k+1}, and $\Sigma_{0,0}$ are symmetric and positive definite.

The state at the next time step $k+1$ can be written as a Taylor series expansion about the centre of the ellipsoid at time k, denoted $\hat{x}_k$

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k)|_{\mathbf{x}_k=\hat{\mathbf{x}}_k} + \left.\frac{\partial f(\mathbf{x}_k, t_k)}{\partial \mathbf{x}}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k} (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \underbrace{\text{H.O.T.} + \mathbf{w}_k}_{\hat{\mathbf{w}}_k} \tag{8}$$

In the traditional EKF, the higher order terms (H.O.T.) are ignored. In order to bound the state uncertainty, the H.O.T. term is combined with the process noise term such that Equation (8) can be written as

$$\mathbf{x}_{k+1} - f(\mathbf{x}_k)|_{\mathbf{x}_k=\hat{\mathbf{x}}_k} = \left.\frac{\partial f(\mathbf{x}_k, t_k)}{\partial \mathbf{x}}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_k} (\mathbf{x}_k - \hat{\mathbf{x}}_k) + \hat{\mathbf{w}}_k$$

where a new ellipsoid that bounds the noise term is defined, $\hat{w}_k$. Similarly for the output equation,

$$\mathbf{y}_{k+1} - h(\mathbf{x}_{k+1})|_{\mathbf{x}_{k+1}=\hat{\mathbf{x}}_{k+1}} = \left.\frac{\partial h(\mathbf{x}_{k+1})}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k+1}=\hat{\mathbf{x}}_{k+1}} (\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}) + \hat{\mathbf{v}}_{k+1}$$

The ESMF requires f and h to be C^2.

The H.O.T. can be bounded several ways. One approach is to choose the process and sensor noise ellipsoids large enough to assure that they bound both the original noise and the H.O.T. This is a common in other applications of the EKF, albeit ad hoc. A more formal approach is based on Taylor's theorem where the remainder of a Taylor expansion can be bounded. The approach here is to bound the H.O.T. using interval mathematics at each time step, as described in Reference [14]. The interval bound can then be bounded using an ellipsoid, and combined with the original process noise bound. The new noise bounds are then written as $\mathscr{E}(0, \hat{Q}_k)$, $\mathscr{E}(0, \hat{R}_{k+1})$ or

$$[\hat{\mathbf{w}}_k]^{\mathrm{T}} \hat{Q}_k^{-1}[\hat{\mathbf{w}}_k] \leqslant 1, \quad \hat{Q}_k = \frac{\bar{Q}_k}{1-\beta_Q} + \frac{Q_k}{\beta_Q}$$

$$[\hat{\mathbf{v}}_{k+1}]^{\mathrm{T}} \hat{R}_{k+1}^{-1}[\hat{\mathbf{v}}_{k+1}] \leqslant 1, \quad \hat{R}_{k+1} = \frac{\bar{R}_{k+1}}{1-\beta_R} + \frac{R_{k+1}}{\beta_R}$$

where $\bar{Q}_k, \bar{R}_{k+1}$ are found using interval analysis and $\beta_Q, \beta_R \in (0,1)$.

The centre of the ellipsoid and the ellipsoid itself can now be propagated forward in time using linearized set membership equations [16]. The algorithm is implemented as a standard recursive estimator with a prediction step and update step. The prediction step is an addition of two ellipsoids (previous state and process noise), and requires an optimization to be performed at each time step, such as minimizing the volume of the ellipsoid. The update step is an intersection of two sets (the predicted state and the projected state from the sensors), and also

requires a optimization. The algorithm is summarized as follows:

ESMF prediction step:

$$\hat{\mathbf{x}}_{k+1,k} = f(\hat{\mathbf{x}}_{k,k})$$
$$\Sigma_{k+1,k} = A_k \frac{\Sigma_{k,k}}{1-\beta_k} A_k^{\mathrm{T}} + \frac{\hat{Q}_k}{\beta_k} \tag{9}$$

ESMF update step:

$$\hat{\mathbf{x}}_{k+1,k+1} = \hat{\mathbf{x}}_{k+1,k} + K_{k+1}(\mathbf{y}_{k+1} - h(\hat{\mathbf{x}}_{k+1,k}))$$
$$\Sigma_{k+1,k+1} = (1-\delta_{k+1})\left[\frac{\Sigma_{k+1,k}}{1-\rho_{k+1}} - \frac{\Sigma_{k+1,k}}{(1-\rho_{k+1})} C_{k+1}^{\mathrm{T}} W_{k+1} C_{k+1} \frac{\Sigma_{k+1,k}}{(1-\rho_{k+1})}\right]$$

where

$$A_k = \left.\frac{\partial f(\mathbf{x}_k)}{\partial \mathbf{x}}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k,k}}, \quad C_{k+1} = \left.\frac{\partial h(\mathbf{x}_k)}{\partial \mathbf{x}}\right|_{\mathbf{x}_k=\hat{\mathbf{x}}_{k+1,k}}$$
$$W_{k+1} = \left[C_{k+1} \frac{\Sigma_{k+1,k}}{1-\rho_{k+1}} C_k^{\mathrm{T}} + \frac{\hat{R}_{k+1}}{\rho_{k+1}}\right]^{-1}$$
$$K_{k+1} = \Sigma_{k+1,k} C_{k+1}^{\mathrm{T}} W_{k+1}$$
$$\delta_{k+1} = [\,y_{k+1} - h(\hat{x}_{k+1,k})]^{\mathrm{T}} W_{k+1} [\,y_{k+1} - h(\hat{x}_{k+1,k})]$$

The actual state $\mathbf{x}_{k+1}$ is then bounded, and written as

$$[\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}]^{\mathrm{T}} [\Sigma_{k+1,\,k+1}]^{-1} [\mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}] \leqslant 1$$

The $\beta_k \in (0,1)$ and $\rho_{k+1} \in (0,1)$ parameters are used in the process of determining outer bounding of summation of ellipsoids (see more details in Reference [17]). For any value of β, ρ between 0 and 1, the ellipsoids are outer bounding. It follows that the trace of volume of the output ellipsoids can be minimized through optimization in order to find smallest ellipsoid. Reference [18] develops several closed form solutions for the selection of β and ρ.

4. COOPERATIVE ESTIMATION

There are distinct advantages to having multiple friendly vehicles in the application domain, with the ability to share information. First, overall system performance can be increased. For example, reconnaissance can be accomplished more quickly and reliably because of the larger number of vehicles. Second, target tracking is less conservative, making it easier for the path planner. To gain more insight into these effects, consider the case of two friendly aircraft tracking one adversary. The radar measurement from a single aircraft produces an uncertainty ellipsoid that is larger in the direction perpendicular to the line of sight (LOS). The path planner must plan around this awkward obstacle that is moving and rotating. However, the other friendly aircraft also generates an uncertainty ellipsoid for the same obstacle, but with a different orientation due to a different LOS direction. If two aircraft are tracking the same adversary, then it must lie within both ellipsoids. By finding an ellipsoid that bounds the intersection of these two ellipsoids, a smaller uncertainty location (and obstacle size) can be found.

The algorithm to compute an outer bounding ellipsoid of two intersecting ellipsoids is developed here. In Reference [17], an analytical solution for a bounding ellipsoid of an intersection is given. The solution works well for the case when the two uncertainty ellipsoids are nearly perpendicular. However, it tends to be more conservative and particularly sensitive for different orientations. More intensive computationally methods, such as on-line numerical optimization are difficult to implement in a dynamic, real time environment which requires on-line estimation.

4.1. Intersecting ellipsoid

A simple closed form solution which takes advantage of the unique shape of the radar observation based uncertainty ellipsoid is considered here. Because the measurement uncertainty ellipsoid is usually 'thin', the intersecting area is not significantly impacted by effect of the curvature of each ellipsoid. As an approximation, each two-dimensional ellipsoid is represented as two linear constraints, or tangent lines parallel to the major axis that bound the ellipsoid. Given two thin uncertainty ellipsoids and two sets of constraints, any orientation other than those with a relative 0° or 180° yields on intersecting area of a polyhedron. The resultant closed convex set allows a bounding ellipsoid for this polyhedron to be determined in closed form. It is noted that the procedure can easily be extended to three-dimensional ellipsoids.

The algorithm consists of two steps. First, the polyhedron that approximates the intersecting area is found. Second, an outer bounding ellipsoid is found.

Step 1: *Finding a bounding polyhedron*: The objective of this step is to find a set of linear constraints which bounds each ellipsoid and subsequently defines the polyhedron. Each constraint is determined based on the minor axis length and the orientation of the ellipsoid, as shown in Figure 4. Each ellipsoid is defined as $\mathscr{E}(\mathbf{c}_i, Q_i)$

$$\mathscr{E}(\mathbf{c}_i, Q_i) = \{\mathbf{x} \,|\, (\mathbf{x} - \mathbf{c}_i)^{\mathrm{T}} Q_i^{-1} (\mathbf{x} - \mathbf{c}_i) \leqslant 1\}$$

Let β_i be the angle of the major axis of the ith ellipsoid with respect to horizontal, $\lambda_{\min}$ be the smallest eigenvalue of Q_i and $V_{\lambda_{\min}}$ be a 2×1 normalized eigenvector corresponding to the

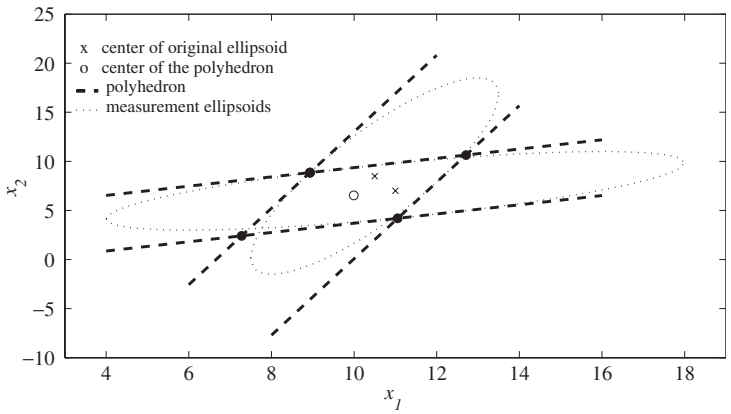

Figure 4. Polyhedron bounding of intersecting area.

smallest eigenvalue. Each ellipsoid is bounded on the intersecting area by two linear constraints written as

$$\underbrace{\begin{bmatrix}\tan(\beta_i)_{-1}\\ -\tan(\beta_i)_1\end{bmatrix}}_{A_i}\begin{Bmatrix}x_1\\ x_2\end{Bmatrix}\leqslant\underbrace{\begin{Bmatrix}b_i^+\\ b_i^-\end{Bmatrix}}_{\mathbf{b}_i}$$

where

$$\beta_i=\cos^{-1}(-V_{\lambda_{\min}}(1))\cdot\operatorname{sgn}(V_{\lambda_{\min}}(2))$$

$$b_{+,i}=-\sqrt{\lambda_{\min}}\cos(\beta_i)-\mathbf{c}_i(2)+\tan(\beta_i)(\mathbf{c}_i(1)+\lambda_{\min}\sin(\beta_i))$$

$$b_{-,i}=-\sqrt{\lambda_{\min}}\cos(\beta_i)+\mathbf{c}_i(2)-\tan(\beta_i)(\mathbf{c}_i(1)-\lambda_{\min}\sin(\beta_i))$$

The intersection for two ellipsoids, $\mathscr{E}(\mathbf{c}_1,Q_1)$ and $\mathscr{E}(\mathbf{c}_2,Q_2)$, is then approximated by a set $\mathscr{P}$

$$\mathscr{P}=\{\mathbf{x}\mid A\mathbf{x}\leqslant B\}$$

$$A=\begin{bmatrix}A_1\\ A_2\end{bmatrix},\quad B=\begin{Bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{Bmatrix}$$

Although, this process involves solving an eigenvalue problem, the computational load is small for two- or three-dimensional problems.

Step 2: *Finding the outer bounding ellipsoid*: In Reference [19], inner and outer bounding ellipsoids for a convex set are found. Given a known volumetric centre (ω). The inner and outer bounding ellipsoids for the n-dimensional polyhedron $\mathscr{P}$ are

$$\mathscr{E}(\omega,H^{-1})\subset\mathscr{P}\subset\mathscr{E}\left(\omega,\sqrt{n/\mu_{\min}}\cdot H^{-1}\right)\tag{10}$$

where

$$G=\operatorname{diag}(A\omega-B)$$

$$\mu=\operatorname{diag}(G^{-1}A(A^{\mathrm{T}}G^{-2}A)^{-1}A^{\mathrm{T}}G^{-1})$$

$$H=A^{\mathrm{T}}G^{-2}A$$

From this result, an outer bounding ellipsoid is found. The bounding ellipsoid must then be scaled in order to minimize its volume, which in turn forces it to be tight around the intersecting area. The inner and outer bounding ellipsoids defined above have identical orientations, based on the orientation of the polyhedron. This fact is used to obtain the smallest volume ellipsoid.

Defining a scaling factor, m, which guarantees that $\mathscr{P}\subset\mathscr{E}(\omega,(H/m^2)^{-1})$, m is maximized in order to find the smallest outer bounding ellipsoid. It is desirable to find a closed form solution in order to alleviate the computational load. The four indices of the polyhedron, as defined by the intersection of the linear constraints, are denoted as $\mathbf{x}_{e,i}$ for $i=1\cdots4$. The scaling m is computed based on the fact that at least one of these extreme points of the set $\mathscr{P}$ must lie on the ellipsoid. Scaling factors are computed for each of the four intersecting points, and the maximum scaling factor among these defines the smallest volume ellipsoid. This is written more

compactly as

$$\mathscr{P} \subset \left\{ \mathscr{E}\left(\omega, \left(\frac{H}{m_{\text{opt}}^2}\right)^{-1}\right) \;\middle|\; m_{\text{opt}} = \arg\min_{m_i} \mathscr{E}\left(\omega, \left(\frac{H}{m_i^2}\right)^{-1}\right) \right\}$$

Considering the ellipsoid $\mathscr{E}(\omega, (H/m^2)^{-1})$ that passes through $\mathbf{x}_{e,i}$, the scaling m_i is

$$m_i = \sqrt{(\mathbf{x}_{e,i} - \omega)^{\mathrm{T}} H (\mathbf{x}_{e,i} - \omega)} \;\Rightarrow\; m_{\text{opt}} = \max_{i=1\cdots4} m_i$$

By combining step 1 and 2, the ellipsoid that bounds the intersecting area of two radar measurement uncertainty ellipsoids is shown in Figure 5. Due to the assumption that the curvature of intersection has little effect, the approach works better in cases where the ellipsoid centers are close to each other and the ellipsoids are relatively thin. As the centers of the ellipsoids move apart, the algorithm's performance decreases. In the unlikely event that the two ellipsoids do not intersect, it is clear that one of the base assumptions of the ESMF has been violated (such as a spurious sensor measurement). These rare cases can be checked by evaluating intersection and either (a) using only one or the other ellipsoid, or (b) scaling the ellipsoids to be larger until they intersect.

This approach can be extended to the three-dimensional case. The intersecting area is bounded by a box instead of a polyhedron; and an outer bounding ellipsoid is determined using a similar approach. Extension to higher than three dimensions becomes complicated because it is almost impossible to evaluate all extremals analytically. This can be accomplished most easily using optimization tools.

4.2. *Orientations of interest*

It is important to understand the useful range of relative orientations of cooperative vehicles in order to know when to make use of vehicle cooperation. In order to do this, the analytical form of the volume of the intersecting ellipsoid is defined. Consider two ellipsoids $\mathscr{E}(\mathbf{c}_1, Q_1)$ and $\mathscr{E}(\mathbf{c}_2, Q_2)$ of equal volume in Figure 6. The volume of each of the original ellipsoids is denoted as $V_{\text{o}} = \sqrt{\pi\,|\,Q_1\,|} = \sqrt{\pi\,|\,Q_2\,|}$.

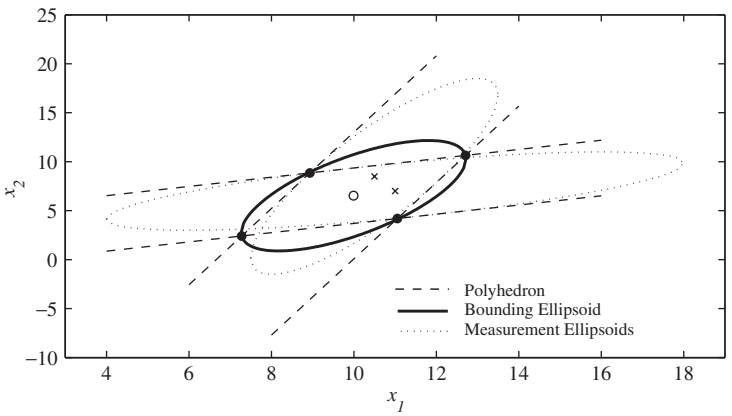

Figure 5. Outer bounding ellipsoid.

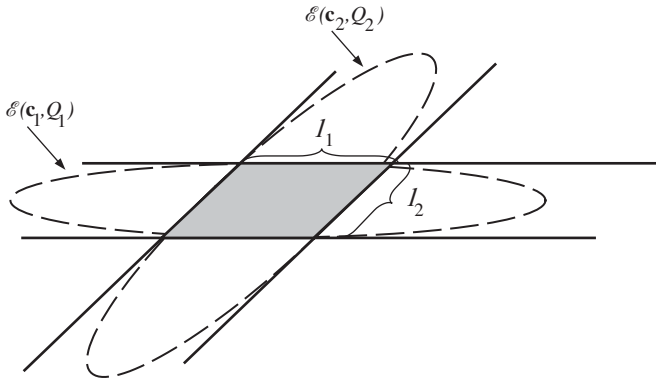

Figure 6. Area of the polyhedron.

The length of each side of a polyhedron $\mathscr{P}$ can be defined using two parameters: the length of minor axis of each ellipsoid, given by the minimum eigenvalue of Q_i matrix ($\lambda_{i,\min}$), and the angle between the major axes orientation, $\Delta\beta = \beta_2 - \beta_1$.

$$l_1 = \frac{\sqrt{\lambda_{1,\min}}}{\sin(\Delta\beta)}, \quad l_2 = \frac{\sqrt{\lambda_{2,\min}}}{\sin(\Delta\beta)}$$

The area of the polyhedron is then written as

$$A_{\mathscr{P}} = l_1 l_2 \cdot \sin(\Delta\beta) = \frac{\sqrt{\lambda_{1,\min}\lambda_{2,\min}}}{\sin(\Delta\beta)}$$

The volume of the outer bounding ellipsoid (V_I) is then defined as the following:

$$V_I = 2\pi \frac{\sqrt{\lambda_{1,\min}\lambda_{2,\min}}}{\sin(\Delta\beta)} \tag{11}$$

It can be seen that the ellipsoid has larger volume for smaller magnitude of $\Delta\beta$. This relation is obviously not linear; in fact for large values of $\Delta\beta$, the difference of volume for different orientations is relatively small. Figure 7 shows the ratio between volume of intersecting ellipsoid and original ellipsoids at different $\Delta\beta$ and different eccentricities. Let f_e be a parameter representing flatness of an ellipsoid, $f_e = \sqrt{\frac{\lambda_{\min}}{\lambda_{\max}}}$. It can be seen that the volume of the intersecting ellipsoid grows larger as $\Delta\beta$ gets smaller. At particular relative angles of orientation $\Delta\beta$, the intersecting volume is larger than original volume. This yields the critical value of relative orientation, $\Delta\beta_{crit}$, where cooperation is useful.

The insights developed from Figure 7 can be used to (1) task pairs of friendly vehicles to track adversaries and (2) understand when cooperative estimation is a performance enhancing task.

5. EVOLUTIONARY COMPUTATION BASED PATH PLANNING

The estimation architecture is designed to be integrated into many different types of path planners. The path planner used here is based on work by Capozzi [7], who developed an

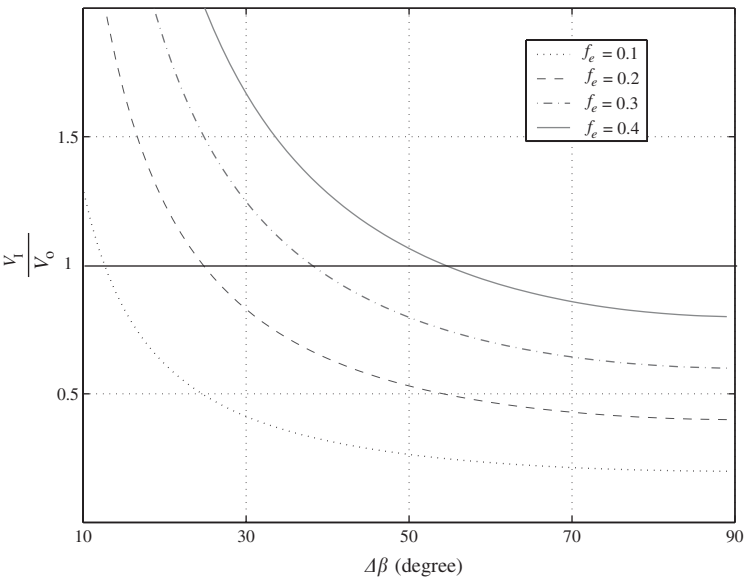

Figure 7. Ratio between the volume of the intersecting ellipsoid and original ellipsoids for the example shown in Figure 6.

evolutionary computation based path planning with stationary obstacles. Evolutionary computation (such as Genetic Algorithms), is a population based search methodology, and is usually based on a cost, or fitness, that allows those paths with the best fitness to continue.

For the evolutionary computation based planner, trajectory primitives for aircraft are defined and integrated in a 'gene' sequence of length Δt. Given that K_a and K_γ are the maximum acceleration and turning rate of the vehicle(s), respectively, the general categories of trajectory primitives for this work are defined based on the following moves:

Primitive 1. Hold present course and heading

$$\Delta x = V\cos(\gamma)\cdot\Delta t, \quad \Delta h = V\sin(\gamma)\cdot\Delta t$$

Primitive 2. Change speed

$$\Delta x = (V \underline{+} K_a\Delta t\cos\gamma)\cdot\Delta t + \tfrac{1}{2}\cos(\gamma)\Delta t^2, \quad \Delta h = (V \underline{+} K_a\Delta t\sin\gamma)\cdot\Delta t \underline{+} \tfrac{1}{2}\sin(\gamma)\Delta t^2$$

Primitive 3. Change heading

$$\Delta x = \underline{+}\frac{V}{K_\gamma}(\cos(\gamma) + \cos(K_\gamma\Delta t + \gamma)), \quad \Delta h = \underline{+}\frac{V}{K_\gamma}(\sin(\gamma) + \sin(K_\gamma\Delta t + \gamma))$$

Primitive 4. Change speed and heading—the trajectory is defined similarly to primitive 3, but with speed that changes according to primitive 2.

The $\underline{+}$ indicates speed up/slow down and turn left/right motion. The trajectory primitives can be defined more finely by using the same definition with different acceleration and/or turning rate.

An initial seed population is selected, based usually on random or *a priori* information. The fitness of each trajectory of the population is evaluated using a cost (fitness) function. The following steps then occur:

1. Mutate/change/adapt genes (or gene sequences) to generate new sequences.
2. Evaluate fitness of new sequences.
3. Rank order sequences, and select next generation.

The first step can be done in various ways, such as reorder the sequence, replace items at various time steps or combine sequences from two different sequences (parents) as shown in Figure 8.

The work here extends the planner to include dynamic obstacles (i.e. uncertainty ellipsoids of the adversarial vehicles) into the path planning algorithm for the friendly vehicles. Because evolution based planner is a search based algorithm, computational effort can be a burden. In contrast, the trajectory must be updated quickly because the obstacles or adversarial vehicles are moving. The approach proposed here is to generate a reachable space over time horizon ΔT_k for each adversarial vehicle using the predictor form of the ESMF, as shown in Figure 9(c). The reachable space (ellipsoids) are then used as obstacles to plan the path over the full time horizon. Ideally, ΔT_k should be small in order to track and use small and accurate adversary information. However this can be difficult because of time required for the search. The smallest time window horizon is selected within the constraint of the computational load.

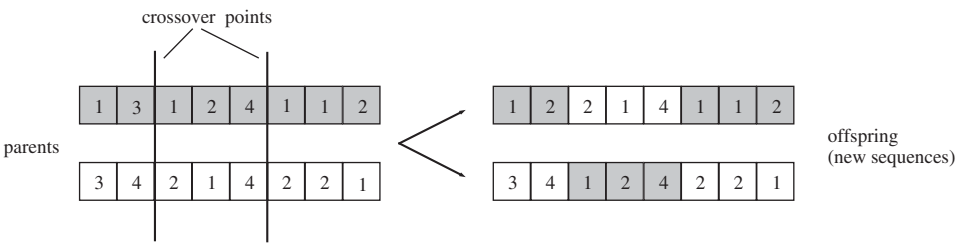

Figure 8. Generating new sequences.

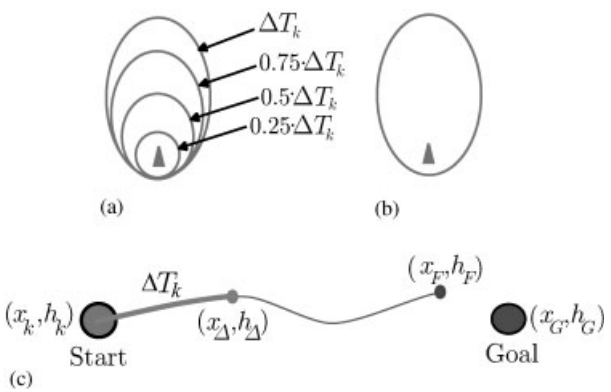

Figure 9. Integrating evolutionary computation based path planning and ESMF.

Figure 9(a) shows a reachable set that grows over time ΔT_k, where as Figure 9(b) shows the largest reachable space. The examples demonstrated here utilize only the largest predicted reachable space created by the predictor step of the ESMF, as shown in Figure 9(b). In this case, the position of obstacles are static over the time window ΔT_k. However, it was found during testing that including an estimate of the velocity is very important to the overall success of the methodology. Therefore, velocity is also included in the fitness function.

The fitness function (F) used to evaluate each of the trajectories in the planner is defined with an objective of moving the vehicle from a start point to a goal quickly, while avoiding collisions in the immediate time horizon ΔT_k and in near future. The fitness function is based on several parameters (J_i) with different weights (w_i). In this work, it is defined as

$$F = \sum_{i=1}^{5} w_i J_i$$

Let (x_Δ, h_Δ) be the location of the vehicle at the end of the time window ΔT_k and (x_F, h_F) be the last location in the sequence as shown in Figure 9(c). The parameters, J_i, are defined as following:

1. Distance to the goal
$$J_1 = \sqrt{(x_F - x_G)^2 + (h_F - h_G)^2}$$
2. Obstacle penetration, which is computed using a minimally enclosing rectangle (MER) as discussed in Reference [7]. The parameter is then,
$$J_2 = \begin{cases} 1 & \text{if collision occurs,} \\ 0 & \text{otherwise} \end{cases}$$
3. Change of path angle, γ—to keep a smooth trajectory. Let N_F be the index at the end of the time window where the first index at the time window starts at 1.
$$J_3 = \sum_{i=k}^{k+N_F} |\gamma_i - \gamma_{i-1}|$$
4. Proximity to obstacles, D (see Figure 10), where a penalty is applied if the relative distance between the vehicle and the predicted obstacle is closer than a 'Safe Distance',

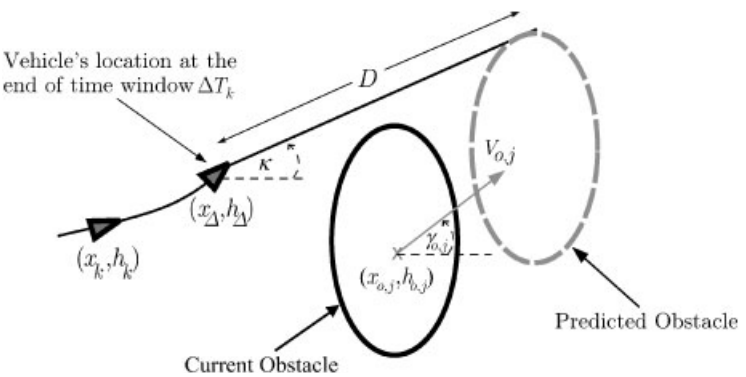

Figure 10. Predicted obstacle avoidance.

D_s. The propagating location of the obstacles forward in time while assuming a constant velocity and heading direction.

$$D = \sqrt{(x_\Delta - x_{o,j} - V_{o,j}\cos(\gamma_{o,j})\Delta T_{k+1})^2 + (h_\Delta - h_{o,j} - V_{o,j}\sin(\gamma_{o,j})\Delta T_{k+1})^2} - R_{o,j}$$

$$J_4 = \begin{cases} 1 & \text{if } D < D_s \\ 0 & \text{otherwise} \end{cases}$$

where $(x_{o,j}, h_{o,j})$ is the current location of obstacle jth, $R_{o,j}$ is the maximum radius of the obstacle jth, $V_{o,j}, \gamma_{o,j}$ are velocity and heading direction of the obstacle jth, respectively.

5. The angle between the vehicle and predicted obstacles, κ (see Figure 10)

$$J_5 = \kappa = \begin{cases} 0 & \text{if } |h_{o,j} - h_\Delta| > R_{o,j}, \\ \tan^{-1}\frac{h_{o,j} + \text{sgn}(h_{o,j} - h_\Delta)R_{o,j}}{|x_{o,j} - x_\Delta|} & \text{otherwise} \end{cases}$$

The parameters, J_4 and J_5, are used to maintain spacing from obstacles. Although obstacles are static over a time horizon of the planner, maintaining spacing using relative position, velocity, and angles in the next time horizon. This prevents the vehicle from getting too close to obstacles such that it cannot avoid collisions.

The weight, w_i is selected based on the maneuverability of the vehicles and the desired movement characteristics. For example, a vehicle with high maneuverability could have relatively low w_3 and w_4 compared to w_1, because it has the ability to change directions quickly. In order to prevent collisions, w_2 must be extremely large. The sensitivity of the performance to each weight varies. The path angle weight, w_3, is generally more sensitive than others because the angle has relatively small scaling. Conversely, w_1 is related to the distance which has larger scale, therefore it is less sensitive. The predicted obstacle parameter weights, w_4 and w_5, are used as an additional safety factor. Therefore they have a wider range of values that work well. The values used in this work are as following: $w_1 = 5$, $w_2 = 10000$, $w_3 = 1$, $w_4 = 20$, and $w_5 = 20$.

6. SIMULATION RESULTS

The estimation theory developed in this work is quite general, and can be applied to many applications. This is a direct result of using decoupled aircraft dynamics while only the individual sensor measurements are coupled. Thus, as aircraft enter and exit the scene, the state tracking vector (Equation (5)) can add or subtract elements (and thus add or subtract obstacles for the path planner). The examples presented here show friendly vehicles' attempt to track the adversary position using radar in the presence of sensor noise, radar loss and communication loss uncertainty. The partial state vector of the adversarial aircraft, as shown in Equation (4), is also estimated. Each vehicle is assumed to be an F-16 like aircraft, with nonlinear dynamics and parameters as described in Reference [20].

Three tracking estimation scenarios are presented, followed by several integrated estimation/path planning examples. First, the tracking ellipsoid case considers a single friendly and single adversarial aircraft with all sensors are working properly. Second, the tracking and predicting ellipsoid case considers the same scenario, but with a loss of radar lock for several seconds. This

is a common occurrence and an important practical case to evaluate the algorithms. The objective is then to continue to track and recover from the loss of contact. Third, a cooperative estimation case is shown where two friendly vehicles share information to reduce conservatism of the tracking ellipsoids. Finally, the integration of estimation architecture and the path planning simulation is presented, including simulations that show the benefit of using cooperative estimation in path planning.

Tracking ellipsoids: Figure 11 shows the actual trajectory of an adversarial aircraft, along with the tracking ellipsoids based on radar measurements and estimation on a friendly aircraft. All sensors are fully functional for this example, and the bounding ellipsoids are expanded by a factor of 20 in order to improve visibility. There are several items to take from Figure 11. First, the actual path of the adversary falls within the bounding ellipsoid at each time step. This validates that the algorithm bounds the solution. Second, as the simulation progresses, the ellipsoids rotate, and become smaller and fatter. These affects can be traced to the observability of the radar measurement. The non-symmetrical property is because the radar range sensor R_{ij} is more observable (less sensitive) than the azimuth angle $\theta_{R,ij}$. The rotation of the ellipse is always perpendicular to the radar line of sight. And, the uncertainty becomes smaller as the range R_{ij} becomes shorter.

Tracking and predicting ellipsoid: Figure 12 shows the same case as in Figure 11, but radar lock is lost for 5 s during the simulation. This case is easily handled with the filter that is proposed here. During the radar blackout, only the prediction step in Equation (9) is used, rather than both the prediction and update step. The tracking ellipsoids now grow during this time, and thus predict a worst case reachable set, given no knowledge from sensors. This is intuitively correct, as without a measurement, the adversary can turn and/or thrust in any direction, and the observer vehicle would have no knowledge. Once radar lock is reestablished, the filter quickly reduces the bounding ellipse (now with sensor information) and tracks the adversary again with little loss of performance. Because the reachable space ellipsoids grow

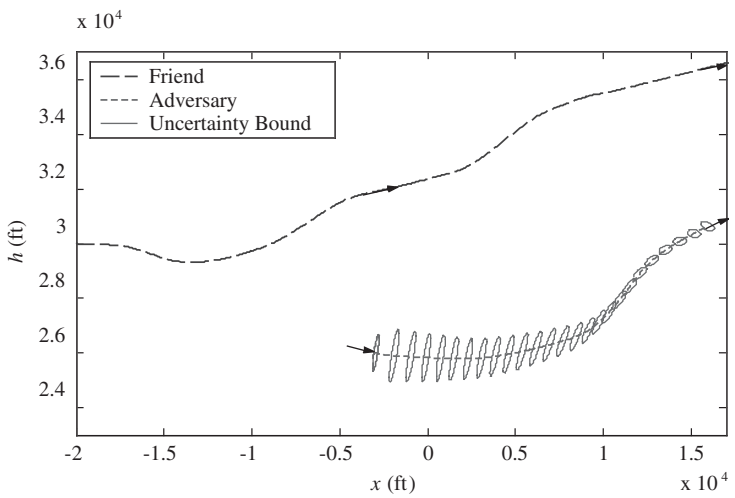

Figure 11. Result of the tracking ellipsoid case compared to the actual path of the adversary.

quickly over time during loss of radar, the state and tracking ellipsoids are removed from the estimator when they become very large, and alternative plans or maneuvers are generated.

Cooperative estimation: Cooperative estimation among friendly aircraft can easily be added into the tracking approach by using a communication link between friendly aircraft. The information (in three dimensions) that must be sent between friendly aircraft is the ellipsoid centre (three data products), along with the ellipsoid itself (six data products). Assuming a 32 bit word and 10 Hz sample rate, this equates to 2.88 kbps for each aircraft being tracked.

Figure 13 shows improvement in the size of tracking uncertainty ellipsoids. By sharing information between two friendly aircraft, the algorithm becomes less conservative, thus making subsequent functions such as, path planning and estimation of adversarial intent easier.

Path planning: Figure 14 shows the path of a friendly aircraft from a *start* to *goal* state using the ESMF to track the adversary and the evolution based planner to develop the trajectory. Shown are three different time snapshots. The trajectory is planned over a $\Delta T_k = 1$ second time period. The control technique is a gain scheduling based, LQR controller that maintains the vehicle on its proposed trajectory. Note that a more efficient control method could be applied, such as a nonlinear receding horizon control in Reference [21], without loss of generality. The obstacle is based on a tracking ellipsoid from the ESMF (Figure 9(b)), and are now used in the path planning segment as obstacles. Several different initial and final conditions are used, and in general the algorithm works quite well. The algorithm obviously has difficulty as the adversarial aircraft becomes more maneuverable during the planning horizon ΔT_k. This could be handled easily with a path planning algorithm that plans over a shorter horizon using the same estimation approach proposed here.

Path planning with cooperative estimation: In more critical situations, cooperative estimation becomes important in path planning. Figure 15 shows an aircraft attempting to maneuver from a Start point to Goal 1, and then on to Goal 2, with a static obstacle near the start and a moving adversarial aircraft flying around Goal 1, with radius of 4000 ft and speed of 1047 ft/s. In this situation, the planner must plan over shorter time horizon, $\Delta T_k = 0.4$ s, because the moving

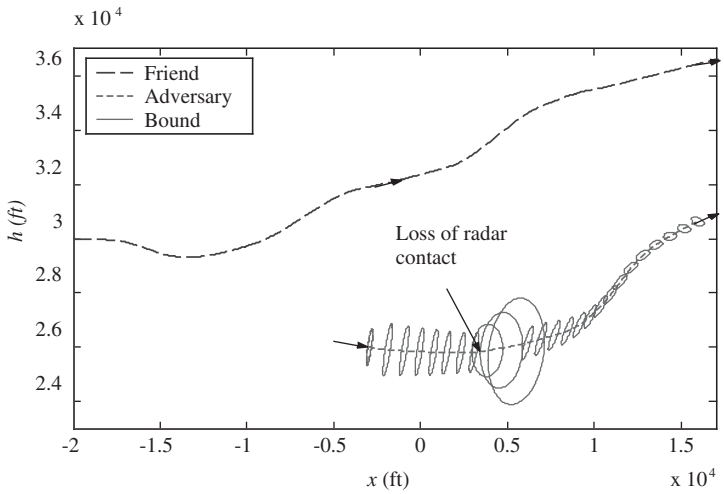

Figure 12. Result of the tracking and predicting ellipsoid case.

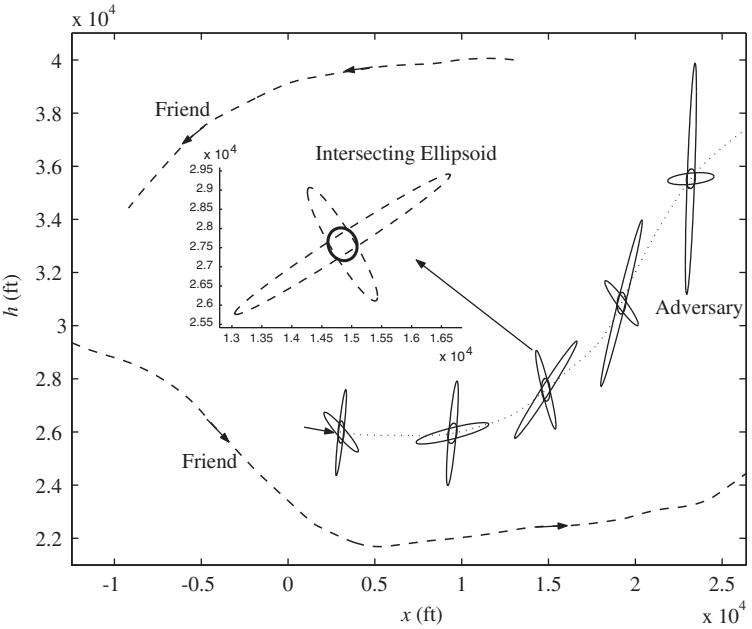

Figure 13. Result of cooperative estimation.

adversary has fast dynamics. A second friendly vehicle is added to the scenario, outside the area of engagement but within radar distance. The second friendly aircraft attempts to reduce the size of the adversary uncertainty using cooperative estimation, and allows the first aircraft to maneuver from the start to the goal(s) more quickly. This scenario was run 100 times, both with and without the second friendly aircraft. Results from simulations indicate that the travel distance for the first friendly aircraft is reduced by 15% and the travel time is reduced by 14.7%.

Figure 16 shows a second example with both a moving goal and static obstacles. The objective is to track and reach the goal (a target in pursuit) within the radius of 70 ft from its actual position. Two cases are again run: a single vehicle tracking and planning, and the same scenario but with a second friendly vehicle in the environment to allow cooperative estimation. Because it is assumed that cooperative estimation provides a smaller uncertainty of the goal, the aircraft can more accurately track and pursue the goal. Results from 100 simulations with and without cooperative estimation show that the planning aircraft reach the moving target using cooperative estimation 18% more accurately than without using it.

These simulation results show benefit in using cooperative estimation. It is important to realize that there are cases when cooperative estimation does not have a significant effect in reducing uncertainty or path planning. If one of the observer vehicles is very close to the adversary, and the noise level is reasonably low, the tracking uncertainty is already small and cooperative estimation only marginally improves the situation. A second example is when there are very few obstacles and a large space to maneuver, such that uncertainty in the obstacle position can be large without affecting path planning. A third case is when the time horizon is large with respect to the maneuver ability of the adversarial aircraft. In this case, the obstacle uncertainty (reachable space) is dominated by the vehicle's dynamics rather than individual

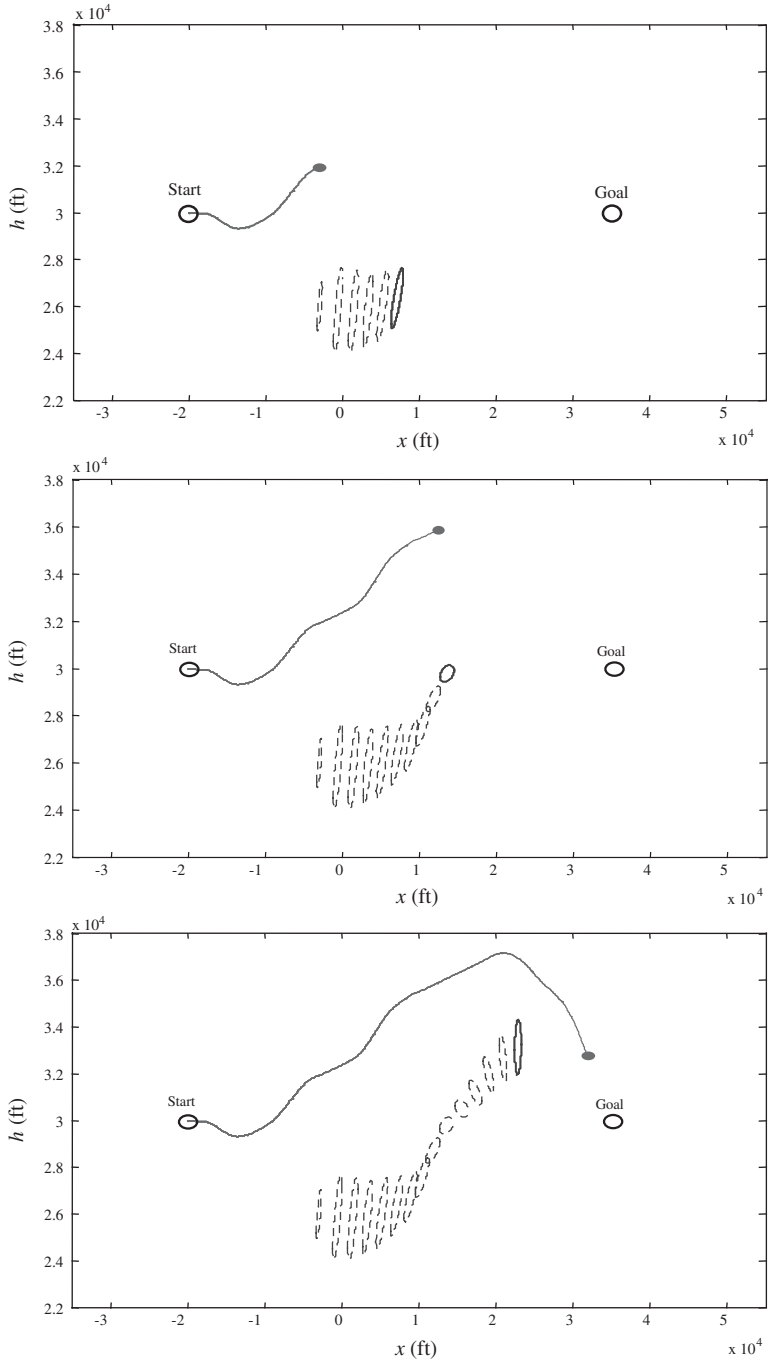

Figure 14. Example of path planning with moving obstacle.

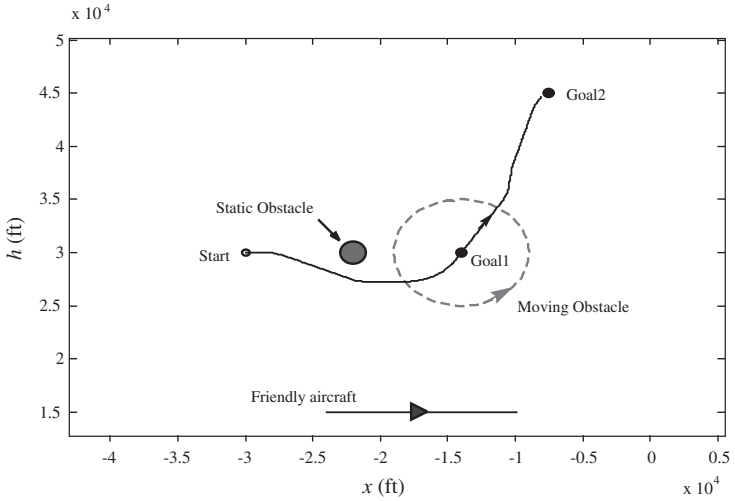

Figure 15. Example of integration of path planning and cooperative estimation.

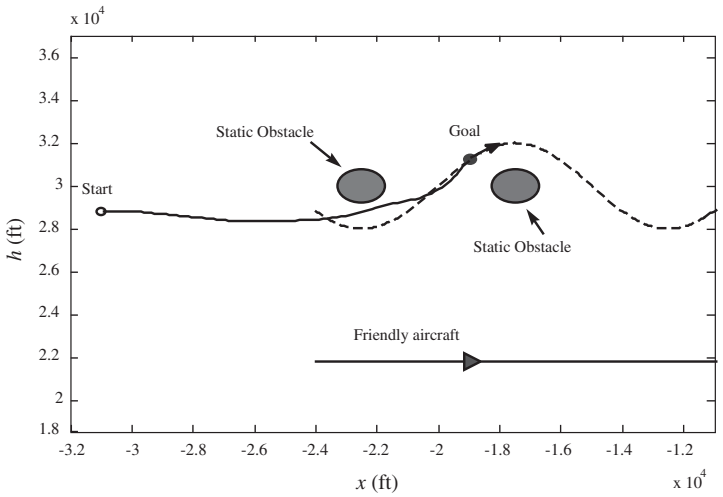

Figure 16. Example of integration of path planning and cooperative estimation with a moving goal.

radar measurements. Cooperative estimation is most applicable for tasks that require high accuracy, such as, path planning in dense environment with tight maneuvering space.

7. CONCLUSION

A full architecture of integrating bounded and cooperative estimation approaches and path planning to generate collision-free paths from one point to another has been presented. The

focus is on multiple aircraft systems, and full nonlinear aircraft models are used in estimation. The bounded nonlinear Set membership filter is used to estimate uncertainties in the presence of nonlinearities in the model. A novel cooperative estimation approach is developed, where information is shared among friendly aircraft. Single and cooperative estimation approaches are then combined with evolutionary computation based path planning. Simulations using F-16 like model show promising results in each of various cases. A single vehicle was able to track an adversary in the presence of noise and loss of radar lock. Cooperative estimation was shown to improve planning distances and time to goal by at least 14.7% in cases with a dense environment and tight maneuvering are required.

APPENDIX: UNCERTAINTY MEASUREMENT ELLIPSOID

This section shows the transformation of the uncertainty from polar co-ordinate measurement (radar) to cartesian co-ordinate. In Reference [13], the approximation using the linearization result. In this work, the uncertainty ellipsoid based on the radar measurement is obtained by using basic probability knowledge.

$$x_{\mathrm{A},j} = R_{ij} \cdot \cos(\theta_{R,ij} + \theta_{\mathrm{F},i}) + x_{\mathrm{F},i}$$

$$h_{\mathrm{A},j} = R_{ij} \cdot \sin(\theta_{R,ij} + \theta_{\mathrm{F},i}) + h_{\mathrm{F},i}$$

In order to be consistent with the concept of set membership filter, it is assumed that measurement noise, $(v_{R,i}, v_{\theta_R,i})$, is bonded by an ellipsoid.

$$\{v_{R,i} \;\; v_{\theta_R,i}\} \begin{bmatrix} \sigma_{R,i}^2 & 0 \\ 0 & \sigma_{\theta_R,i}^2 \end{bmatrix}^{-1} \begin{Bmatrix} v_{R,i} \\ v_{\theta_R,i} \end{Bmatrix} \leqslant 1$$

Because R_{ij} and $\theta_{R,ij}$ are independent, the joint probability density function is defined as:

$$p(R, \theta_R) = \frac{1}{\pi \sigma_{R,i} \sigma_{\theta_R,i}}$$

Denote the expected value, covariance and cross covariance of $x_{\mathrm{A},j}$ and $h_{\mathrm{A},j}$ as $\tilde{x}_{\mathrm{A},j}$, $\tilde{h}_{\mathrm{A},j}$, $\tilde{P}_{xx}$, $\tilde{P}_{hh}$, and $\tilde{P}_{xh}$, respectively, are computed to get description of the measurement ellipsoid.

$$\tilde{x}_{\mathrm{A},j} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{\mathrm{A},j} \cdot p(R, \theta_R)\, \mathrm{d}R\, \mathrm{d}\theta_R = R_{ij} \cos(\theta_{R,ij} + \theta_{\mathrm{F},i}) \cdot \frac{\sin(\sigma_{\theta_R,i})}{\sigma_{\theta_R,i}} + x_{\mathrm{F},i}$$

$$\tilde{h}_{\mathrm{A},j} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_{\mathrm{A},j} \cdot p(R, \theta_R)\, \mathrm{d}R\, \mathrm{d}\theta_R = R_{ij} \sin(\theta_{R,ij} + \theta_{\mathrm{F},i}) \cdot \frac{\sin(\sigma_{\theta_R,i})}{\sigma_{\theta_R,i}} + h_{\mathrm{F},i}$$

$$\tilde{P}_{xx} = E(x_{\mathrm{A},j}^2) - \tilde{x}_{\mathrm{A},j}^2$$

$$E(x_{\mathrm{A},j}^2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{\mathrm{A},j}^2 \cdot p(R, \theta_R)\, \mathrm{d}R\, \mathrm{d}\theta_R$$

$$\tilde{P}_{xx} = \frac{C}{3} \cdot (6R_{ij}^2 \sigma_{R,i} + 2\sigma_{R,i}^3) \left(\sigma_{\theta_R,i} + \frac{S2}{4} \right) - 4 \cdot C^2 S1^2 (R_{ij}^2 \sigma_{R,i}^2)$$

$$\tilde{P}_{hh} = E(h_{\mathrm{A},j}^2) - \tilde{h}_{\mathrm{A},j}^2$$

$$E(h_{\mathrm{A},j}^2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_{\mathrm{A},j}^2 \cdot p(R,\theta_R)\,\mathrm{d}R\,\mathrm{d}\theta_R$$

$$\tilde{P}_{hh} = \frac{C}{3} \cdot (6R_{ij}^2\sigma_{R,i} + 2\sigma_{R,i}^3)\left(\sigma_{\theta_R,i} - \frac{S2}{4}\right) - 4 \cdot C^2 C1^2 (R_{ij}^2\sigma_{R,i}^2)$$

$$\tilde{P}_{xh} = E(x_{\mathrm{A},j}h_{\mathrm{A},j}) - \tilde{x}_{\mathrm{A},j}\tilde{h}_{\mathrm{A},j}$$

$$E(x_{\mathrm{A},j}h_{\mathrm{A},j}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{\mathrm{A},j}h_{\mathrm{A},j} \cdot p(R,\theta_R)\,\mathrm{d}R\,\mathrm{d}\theta_R$$

$$\tilde{P}_{xh} = \frac{-C}{12} \cdot (6R_{ij}^2\sigma_{R,i} + 2\sigma_{R,i}^3)C2 - 4 \cdot C^2 C1 S1 (R_{ij}^2\sigma_{R,i}^2)$$

where

$$C = \frac{1}{\pi\sigma_{R,i}\,\sigma_{\theta_R,i}}$$

$$C1 = \cos(\theta_{R,ij} + \sigma_{\theta_R,i} + \theta_{\mathrm{F},i}) - \cos(\theta_{R,ij} - \sigma_{\theta_R,i} + \theta_{\mathrm{F},i}) = -2\sin(\theta_{R,ij} + \theta_{\mathrm{F},i})\sin(\sigma_{\theta_R,i})$$

$$S1 = \sin(\theta_{R,ij} + \sigma_{\theta_R,i} + \theta_{\mathrm{F},i}) - \sin(\theta_{R,ij} - \sigma_{\theta_R,i} + \theta_{\mathrm{F},i}) = 2\cos(\theta_{R,ij} + \theta_{\mathrm{F},i})\sin(\sigma_{\theta_R,i})$$

$$C2 = \cos[2(\theta_{R,ij} + \sigma_{\theta_R,i} + \theta_{\mathrm{F},i})] - \cos[2(\theta_{R,ij} - \sigma_{\theta_R,i} + \theta_{\mathrm{F},i})] = -2\sin[2(\theta_{R,ij} + \theta_{\mathrm{F},i})]\sin(2\sigma_{\theta_R,i})$$

$$S2 = \sin[2(\theta_{R,ij} + \sigma_{\theta_R,i} + \theta_{\mathrm{F},i})] - \sin[2(\theta_{R,ij} - \sigma_{\theta_R,i} + \theta_{\mathrm{F},i})] = 2\cos[2(\theta_{R,ij} + \theta_{\mathrm{F},i})]\sin(2\sigma_{\theta_R,i})$$

The measurement uncertainty ellipsoid from $\mathrm{AC}_{\mathrm{F},i}$ to $\mathrm{AC}_{\mathrm{A},j}$ then can be described by

$$\tilde{\mathscr{E}}_{ij} = \mathscr{E}\left(\left\{\begin{matrix}\tilde{x}_{\mathrm{A},j}\\ \tilde{h}_{\mathrm{A},j}\end{matrix}\right\}, \underbrace{\begin{bmatrix}P_{xx}P_{xh}\\ P_{xh}P_{hh}\end{bmatrix}}_{\tilde{P}}\right)$$

REFERENCES

1. Bar-Shalom Y, Li XR. *Estimation and Tracking*: *Principles*, *Techniques and Software*. YBS Publishing: Connecticut, USA, February 1998.
2. Daeipour E, Bar-Shalom Y. IMM tracking of maneuvering targets in the presence of glint. *IEEE Transactions on Aerospace and Electronic Systems* 1998; **34**:996–1003.
3. Van Keuk G. MHT Extraction and track maintenance of a target formation. *IEEE Transactions on Aerospace and Electronic Systems* 1999; **35**:242–54.
4. Hue C, Le Cadre JP, Perez P. Tracking multiple objects with particle filtering. *IEEE Transactions on Aerospace and Electronic Systems* 2002; **38**:791–812.
5. Farina A, Ristic B, Benvenuti D. Tracking a ballistic target: comparison of several nonlinear filters. *IEEE Transactions on Aerospace and Electronic Systems* 2002; **38**:242–54.

6. Murray RM, Hauser J, Jadbabaie A, Milam MB, Petit N, Dunbar WB, Franz R. Online control customization via optimization based control. In Samad T, Balas G. (eds). *Software Enabled Control*: *Information Technology for Dynamical Systems*, John Wiley and Sons, New York, 2002.
7. Capozzi BJ. Evolution based path planning and management for autonomous vehicles. *Doctoral Dissertation*, University of Washington, 2001.
8. Brooks RA. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation* 1986; **RA-2**:14–23.
9. Milam MB, Mushambi K, Murray RM. A new computational approach to real-time trajectory generation for constrained mechanical systems. *39th IEEE Conference on Decision and Control*, December 2000, Sydney.
10. Stentz A. Optimal and efficient path planning for partially-known environments. *1994 International Conference on Robotics and Automation*, San Diego, USA, 1994; **4**:3310–3317.
11. Parunak H, Brueckner S. Entropy and self-organization in multi-agent systems. *International Conference on Autonomous Agents* (*Agents 2001*), Montreal, Canada, 2001.
12. Frazzoli E, Dahleh MA, Feron E. Real-time motion planning for agile autonomous vehicles. *AIAA Guidance*, *Navigation*, *and Control Conference*, Denver, CO, 2000.
13. Blackman S, Popoli R. *Design and Analysis of Modern Tracking Systems*. Artech House: Norwood, MA, 1999.
14. Scholte E, Campbell ME. On-line nonlinear guaranteed estimation with application to a high performance aircraft. *American Control Conference*, Anchorage, AK, 2002.
15. Campbell ME, Udrea B. Collision avoidance in satellite clusters. *American Control Conference*, Anchorage, AK, 2002.
16. Schweppe FC. *Uncertain Dynamic Systems*. Prentice-Hall: Englewood Cliffs, NJ, 1973.
17. Kurzhanski A, Valyi I. *Ellipsoidal Calculus for Estimation and Control*. Birkhauser: Basel, 1997.
18. Chernousko FL. *State Estimation for Dynamic Systems*. CRC Press: Boca Raton, FL, 1994.
19. Anstreicher KM. Ellipsoidal approximations of convex sets based on the volume barrier. *Mathematics of Operations Research* 1999; **24**(1):193–203.
20. Stevens BL, Lewis FL. *Aircraft Control and Simulation*. Wiley: New York, 1992.
21. Bhattacharya R, Balas GJ, Kaya A, Packard A. Nonlinear receding horizon control of F-16 aircraft. *Proceedings of the American Control Conference*, 2001; **1**:518–522.

