43rd IEEE Conference on Decision and Control
December 14-17, 2004
Atlantis, Paradise Island, Bahamas

Low Observability Path Planning for an Unmanned Air Vehicle Using Mixed Integer Linear Programming

Atif Chaudhry, Kathy Misovec, and Raffaello D'Andrea

Abstract— **Detection of an Unmanned Air Vehicle by radar is dependent on many variables including range, altitude, and relative orientation. Given a radar location and appropriate model for the likelihood of detection, a path plan can be created for an Unmanned Air Vehicle which constrains the probability of detection. In this paper such an approach is taken using a linearized detection model. The detection model and the Unmanned Air Vehicle's dynamics are represented as a linear program subject to mixed integer constraints. This mixed integer linear program is then solved with commercial software which has been traditionally used by the Operations Research community. This approach searches for all feasible solutions and produces the best path plan based on the user specified parameters.**

I. INTRODUCTION

Optimal path planning for Unmanned Air Vehicles (UAVs) is a difficult task due to the intrinsically non-convex nature of path optimization. Recently, new approaches have been used which utilize techniques traditionally used in the field of Operations Research. By using linearized dynamic models and mixed integer constraints, researchers have used Mixed Integer Linear Programming (MILP) for various path planning and controls applications [1] [2] [3] [4] [5] [6] [7]. MILP problem formulations can be solved using commercial software that employs a branch and bound algorithm. This algorithm, by looking at all path possibilities, returns the optimal solution, if one exists.

Previous efforts [1] and [2] have concentrated on path planning for multiple vehicles and multiple goals. Constraints have included time to waypoints and total effort, or fuel. In this paper the MILP approach for path planning is expanded to include constraints derived not only from the relationship, but also the interactions, between the UAV and adversaries.

Radar guided surface to air missiles (SAMs) are a major threat to UAVs. While the loss of a UAV does not result in a human casualty, it can jeopardize the mission and results in the loss of an important battlefield resource. Knowledge of a SAM's location and a good model of its attack capabilities can be used to create a UAV path plan with acceptable risk. A good model for a SAM's probability of detecting a UAV takes many factors into consideration. The most basic is distance between the SAM and the UAV. Another important factor in determining whether a SAM site detects a UAV is the attitude dependent signature presented by the UAV to the SAM's radar. Once detected, it is possible for a UAV to avoid destruction by taking advantage of the possibility of lock loss. Lock loss occurs when radar detects a UAV but is unable to continue detection, or lock on, long enough for a SAM to be fired at the UAV.

Non-convexities, path dependencies, and sharp gradients are introduced to the path generation problem when taking into account UAV flight dynamics, the probability of radar detection based on UAV state, and the occurrence of lock loss. Such problems have been previously approached using a gradient descent based method which can result in local minima [8]. Discrete approximations to continuous shortest paths have also been investigated in relation to SAM avoidance [9]. The approach outlined in this paper uses a linearized model of the probability of detection, lock loss phenomenon, and UAV dynamics to create a large MILP formulation. Once solved, this MILP formulation results in the global solution. Examples of possible path plans, as a function of acceptable probabilities of detection, are also presented.

II. GENERAL MODEL

The model is presented in this section in three parts: the vehicle dynamics, the probability of detection model, and the lock loss model. The model is a simplified version of the Open Experimental Platform (OEP) developed by Boeing [10]. Throughout this report, it is assumed that the aircraft maintains a fixed altitude and that the radar is on the ground. Once the model is described, the MILP formulation that was developed for this model is given in the next section.

A. Vehicle Dynamics

The inputs to the aircraft model include measurements of the aircraft position and velocity as well as the destination waypoint position. A constant speed for travel between waypoints is also an input. The outputs for the aircraft model are the aircraft position and attitude in inertial coordinates (north, east, up). The model is highly simplified and does not accurately portray the true physics of the aircraft. However, the model was developed for research in mission planning systems, and it is assumed that these planning controls will be designed and tested in conjunction with more complex models of the vehicles and their flight management systems. The model assumptions include constant altitude (2D) flight; instantaneous changes in speed, heading and bank angle; and simplified turning. During turns the bank angle is $\pm 45^\circ$, while during steady level flight, the bank angle is zero. The following equations are used for the aircraft state for $t_i \leq t \leq t_{i+1}$ where t_i denotes when the UAV is at waypoint i

$$\begin{aligned} x(t) &= x(t_i) + U(t_i)\cos(\psi(t_i))(t - t_i) \\ y(t) &= y(t_i) + U(t_i)\sin(\psi(t_i))(t - t_i) \\ h(t) &= h(t_i) \\ \psi(t) &= \psi(t_{i+1}) \\ \phi(t) &= \frac{\pi}{4}\delta_{pt} - \frac{\pi}{4}\delta_{nt} \end{aligned} \tag{1}$$

where x, y, and h are the aircraft positions along the north, east, and up axes, respectively, and $U(t)$ is the speed. The heading and bank angles are ψ and ϕ, respectively. The integer variables, δ_{pt} and δ_{nt}, are defined for positive (right) and negative (left) turns as follows.

$$\begin{aligned} \delta_{pt} = 1 \leftrightarrow t \leq t_i + T_{turn} \quad \text{and} \quad \psi(t_{i+1}) - \psi(t_i) \geq \varepsilon \\ \delta_{nt} = 1 \leftrightarrow t \leq t_i + T_{turn} \quad \text{and} \quad \psi(t_{i+1}) - \psi(t_i) \leq -\varepsilon \end{aligned}$$

where ε is a small positive constant. The value of ε is chosen to be larger than any computer rounding or precision errors that may be present which could give an erroneous positive or negative value. The heading, $\psi(t_i)$, is the angle between the nose and north and

is positive clockwise about the up axis and is defined below.

$$\psi(t_i) = \tan^{-1}\left(\frac{y(t_{i+1}) - y(t_i)}{x(t_{i+1}) - x(t_i)}\right)$$

The steady level flight turn rate equation is used to compute the turn time, T_{turn}, as follows, where g is the acceleration due to gravity,

$$T_{turn} = \frac{\mid \psi(t_{i+1}) - \psi(t_i) \mid}{\frac{g\tan\left(\frac{\pi}{4}\right)}{U(t_i)}}$$

To compute the magnitude of the turn angle, the dot product between the velocity vector, $v(t_i) = v_x(t_i)[\underline{x}] + v_y(t_i)[\underline{y}]$, where $[\underline{x}]$ and $[\underline{y}]$ are unit vectors, and the difference between the destination waypoint and the current waypoint

$$\Delta R = (x(t_{i+1}) - x(t_i))[\underline{x}] + (y(t_{i+1}) - y(t_i))[\underline{y}]$$

is used as follows

$$\cos(\mid \psi(t_{i+1}) - \psi(t_i) \mid) =$$

$$\min\left\{1, \frac{v_x(t_i)(x(t_{i+1}) - x(t_i)) + v_y(t_i)(y(t_{i+1}) - y(t_i))}{\sqrt{(v_x(t_i))^2 + (v_y(t_i))^2}}\right.$$

$$\left. \cdot \frac{1}{\sqrt{(x(t_{i+1}) - x(t_i))^2 + (y(t_{i+1}) - y(t_i))^2}}\right\} \quad (2)$$

For the sign of the turn angle, the cross product is used. Specifically if $y(t_{i+1})v_x(t_i) - x(t_{i+1})v_y(t_i) > 0$, then the change in turn angle is negative, $\psi(t_{i+1}) - \psi(t_i) = -|\psi(t_{i+1}) - \psi(t_i)|$. This is a "left" turn, and is achieved by rolling the aircraft with roll angle $\phi = -45^\circ$, with the left wing down. Similarly if $y(t_{i+1})v_x(t_i) - x(t_{i+1})v_y(t_i) < 0$, then the change in turn angle is positive, $\psi(t_{i+1}) - \psi(t_i) = |\psi(t_{i+1}) - \psi(t_i)|$. This is a "right" turn, and is achieved by rolling the aircraft with roll angle $\phi = 45^\circ$, with the right wing down.

Note that there could be cases where there is no turn. This happens if the turn angle is less than a small amount specified in the OEP. Also, there could be cases where the turn time is greater than or equal to time between waypoints $t_{i+1} - t_i$. In this case, the OEP model assumes turning for the entire segment.

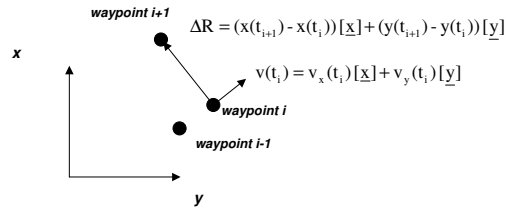

Fig. 1. The turn angle is computed from the velocity vector and the vector between the destination and current waypoints.

B. Probability of Detection

The second component of the model is the detection model. This model computes the probability of detection of a UAV by an opponent SAM radar. The inputs of the detection model are the outputs of the aircraft model, specifically, the aircraft position and attitude. The positions of opponent radars are also input if the aircraft is within an engagement range of the radar. For each radar, the signature and probability of detection of the aircraft is included in the output of the detection model. Signature is an intermediary variable that is related to radar cross section.

TABLE I
SIGNATURE IS A FUNCTION OF AZIMUTH AND ELEVATION

		Az					
		0	± 30	± 31	± 159	± 160	± 180
El	90°	1e0	1e0	1e0	1e0	1e0	1e0
	45°	5e-3	5e-3	1e0	1e0	5e-3	5e-3
	20°	5e-4	5e-4	5e-1	5e-1	5e-4	5e-4
	0°	5e-5	5e-5	5e-1	5e-1	5e-5	5e-5
	−20°	5e-4	5e-4	5e-1	5e-1	5e-4	5e-4
	−45°	5e-3	5e-3	1e0	1e0	5e-3	5e-3
	−90°	1e0	1e0	1e0	1e0	1e0	1e0

TABLE II
PROBABILITY OF DETECTION TABLE: THIS TABLE RELATES SIGNATURE, RANGE IN KM, AND PROBABILITY OF DETECTION.

	Probability of Detection			
Signature	.99	.5	.1	.01
1	380	481.2	555.6	656.6
1e-1	213.7	270.6	312.5	369.2
1e-2	120.2	152.2	175.7	207.6
1e-3	67.6	85.6	98.8	116.8
1e-4	38	48.1	55.6	65.7
1e-5	21.4	27.1	31.2	36.9
1e-6	12	15.2	17.6	20.8

For each radar, a vector from the aircraft to the radar is transformed to aircraft body axis. This vector is then transformed to spherical coordinates for azimuth (Az), elevation (El) and range (R).

The signature is computed with a table look-up as a function of azimuth and elevation. The table values are given in Tables I and II. Note that the signature model has sharp changes between azimuth magnitudes between 30° and 31° and also between 159° and 160°. When the nose of the aircraft points directly to the radar or directly away from the radar, and the elevation magnitudes are relatively low, there is a region of low signature. The probability of detection is computed as a function of signature and range using another table look-up. For the probability of detection model, note that probability of detection increases with increasing signature and decreasing range. To determine probability of detection as a function of signature and range, interpolation is used. The maximum and minimum probability of detection are 0.99 and 0.01, respectively.

In the OEP, a random number is generated with uniform probability between 0 and 1. If this number is greater than or equal to the probability of detection, the UAV is detected. If the generated number is less than the probability of detection, the UAV is not detected. We assume that the problem is deterministic, and that a detection occurs if the probability of detection is greater than a threshold value.

C. Lock Loss

A mixed logical dynamical (MLD) representation of the lock loss model is presented here and shown in Figure 2. Define four states: disengage, engage, launch, and damage. The integer, binary variables δ_e, δ_{LL}, δ_{launch}, δ_{damage} are used to signal events

that transition the opponent SAM radar system from one state to another. When $\delta_e = 1$, the opponent SAM engages the UAV. The variables δ_{LL}, δ_{launch}, δ_{damage} are used in conjunction with timers described below for lock loss, launch and time-to-target. When a timer surpasses a threshold value, a change in state is triggered. When the lock loss timer exceeds threshold T_{LL}, the variable δ_{LL} is set to 1 and the state transitions to disengage. When the engage timer exceeds T_{eng}, the variable δ_{launch} is set to 1, indicating that the SAM is launching a missile at the UAV. When the time-to-target timer exceeds T_{target}, the variable δ_{damage} is set to 1, indicating that the missile has had enough time to reach the target. These conditions are detailed by the following logical equations described below:

$$\begin{aligned} \delta_e &= 1 \leftrightarrow \text{engagement conditions met} \\ \delta_{LL} &= 1 \leftrightarrow x_{LL} \geq T_{LL} \\ \delta_{Launch} &= 1 \leftrightarrow x_{eng} \geq T_{eng} \\ \delta_{damage} &= 1 \leftrightarrow x_{launch} \geq T_{target} \end{aligned} \tag{3}$$

The variables x_{LL}, x_{eng}, and x_{launch} are the timer variables and are updated using the equations below.

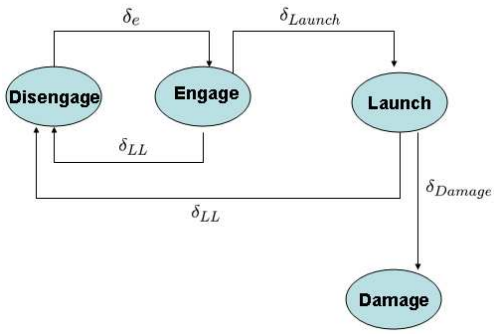

Fig. 2. Lock Loss four state model.

A clock is defined with the equation below.

$$\begin{aligned} t(j+1) &= t(j) + T_s \qquad j = 0, 1, 2, ... \\ t(0) &= 0 \end{aligned} \tag{4}$$

Where T_s is the sample time.

For the lock loss timer, if there is no detection, the lock loss timer is incremented. If there is a detection, the lock loss timer is reset to zero. The lock loss threshold, T_{LL}, in (3) is specified in the OEP model.

$$x_{LL}(t(j+1)) = x_{LL}(t(j))(1 - \delta_e) + (1 - \delta_e) \tag{5}$$

The engage timer increments when the system is engaged and is reset to zero if there is a lock loss event.

$$x_{eng}(t(j+1)) = (x_{eng}(t(j)) + \delta_e)(1 - \delta_{LL}) \tag{6}$$

Similarly, the launch timer increments when a launch has occurred and is reset to zero if there is a lock loss event.

$$x_{launch}(t(j+1)) = (x_{launch}(t(j)) + \delta_{launch})(1 - \delta_{LL}) \tag{7}$$

III. MILP Problem Formulation

In this section, the MILP representation developed for the problem is presented. This MILP representation captures the essence of the model described in section II using only linear equations and binary variables. The first subsection pertains to developing a vehicle dynamics model that is suitable for the MILP approach. The second subsection addresses the detection model, and the third pertains to the lock loss model.

In order to represent the model described in the previous section as a Mixed Integer Linear Programming (MILP) problem, several simplifications are made. The MILP formulation, like the model, assumes the UAV maintains constant altitude. In addition, the bank angle of the UAV is always zero. This simplification is valid if the turn time is small compared to the size of the discrete time steps used in MILP.

A. Dynamics Constraints

The total flight time is represented throughout the formulation as T discrete time steps, which is a parameter the user specifies in the input data file. The UAV is given T time steps to reach the desired end point.

There are T values for the UAV's x, y position. These T values are indexed with variable k. At any individual time step, k, the UAV's position at that time step, $x[k]$ and $y[k]$, is calculated to be the starting x and y position, called x_0 and y_0, plus the sum of all velocity components, $v_x[k]$ and $v_y[k]$, multiplied by the constant time step length T_s, for all values of k from zero to the current time step. Written as equations this gives:

$$x[k] = x_0 + \sum_{j=0}^{k} (v_x[j] \cdot T_s) \tag{8}$$

$$y[k] = y_0 + \sum_{j=0}^{k} (v_y[j] \cdot T_s) \tag{9}$$

Velocity components v_x and v_y must be constrained so that the magnitude of the velocity vector is not greater than the maximum velocity of the UAV, v_{max}.

Calculating the total velocity using v_x and v_y results in a non-linear function, due to the square root. For a MILP representation a linearized approach must be taken to constrain v_x and v_y. One such approach, outlined in [2], is to constrain v_x and v_y separately to be less than some v_m.

$$| v_x | \leq v_m \quad \text{and} \quad | v_y | \leq v_m \tag{10}$$

Plotted on a v_x versus v_y graph, as in Fig. 3, these constraints create a box that is inscribed in the circle corresponding to all feasible v_x and v_y values. This box is a poor approximation to the circle.

To achieve a better approximation to the circle, more constraints can be added.

$$v_x[k] \sin\left(\frac{2\pi m}{M}\right) + v_y[k] \cos\left(\frac{2\pi m}{M}\right) \leq v_{max} \cos\left(\frac{\pi}{M}\right) \tag{11}$$

This constraint has to hold for all integer values of m in the set from 1 to M, where M is the number or sides of the polygon approximation of the circle. Since the polygon approximation is inscribed in the circle, v_{max} must be multiplied by the cosine factor. Without this factor, the circle would fall within the polygon and there would be v_x and v_y combinations allowed by the

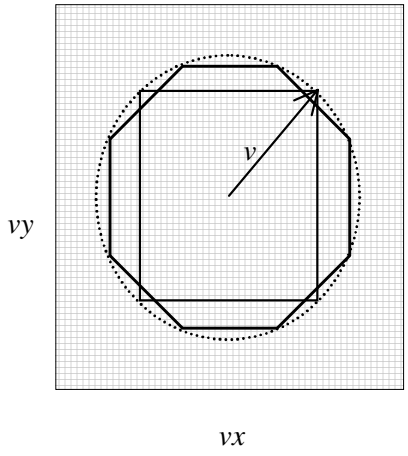

Fig. 3. Polygon circle approximation for M = 4 and M = 8.

approximation which produce a total $v > v_{max}$. This is easiest to see in the case where $M = 4$ since there is a large difference between the box inscribed in the circle and the box that contains the circle.

The above constraints only ensure that v_{max} is not violated. To find the actual velocity at any time step k, $v[k]$, the v_x and v_y components must be used. Normally, calculating the magnitude of a vector from its components relies on the nonlinear square root.

$$v[k] = \sqrt{(v_x[k])^2 + (v_y[k])^2} \tag{12}$$

A linear approximation uses three constraints and several binary variables, b_v. For any time step k and for all m in the range of 1 to M, where L is an arbitrarily large number:

$$v_x[k]\sin\Big(\frac{2\pi m}{M}\Big) + v_y[k]\cos\Big(\frac{2\pi m}{M}\Big) \leq v[k] \tag{13}$$

$$\Big(v_x[k]\sin\Big(\frac{2\pi m}{M}\Big) + v_y[k]\cos\Big(\frac{2\pi m}{M}\Big)\Big)\cdot 1.01 \geq v[k] - L(1 - b_v[k,m]) \tag{14}$$

and

$$\sum_{j=1}^{M} b_v[k,j] = 1 \quad , \quad b_v[k,j] \in \{\,0,1\} \tag{15}$$

The true value of $v[k]$, corresponding to the UAV velocity, will satisfy these conditions for only one value of m. First, (13) constrains $v[k]$ to be a value that is bigger than any possible combination of $v_x[k]$ and $v_y[k]$, for any value m. Equation (14) also constrains $v[k]$ to be smaller than any combination multiplied by 1.01, or else the corresponding binary variable, $b_v[k]$, must be zero. With no other constraints, many values of $v[k]$ could satisfy (13) and (14) by having all zero values of $b_v[k]$. To prevent this, (15) constrains $b_v[k]$ such that it must be non-zero for one m value. This m corresponds to one of the corners of the M sided polygon approximation. When $b_v[k]$ does equal 1, (13) and (14) can only be satisfied with a $v[k]$ which is a close approximation to the actual velocity value.

The heading value is important in determining the radar signature of the aircraft. The heading angle is tracked with a variable which will later be constrained to ensure that a small enough cross section is presented to any radar to avoid detection. Given the model for UAV dynamics, it can be assumed that the UAV always faces the direction it is traveling. Therefore the heading is related to the velocity vector. In finding $v[k]$ there are m binary variables, $b_v[k]$, for each time step k. These variables all equal zero except for the one that corresponds to the m which relates how $v_x[k]$ and $v_y[k]$ combine to form $v[k]$. This value of m is directly related to the heading angle of the UAV by

$$\sum_{j=1}^{M} b_v[k,j]\cdot j \cdot \frac{360}{M} = \text{heading angle} \tag{16}$$

To force the UAV to go from its starting position to the final, desired position, more constraints and binary variables are necessary. Again assuming that L is an arbitrary large value we can write the following:

$$x[k] - x_f \leq L\cdot(1 - b_f[k]) \tag{17}$$
$$x[k] - x_f \geq -L\cdot(1 - b_f[k]) \tag{18}$$
$$y[k] - y_f \leq L\cdot(1 - b_f[k]) \tag{19}$$
$$y[k] - y_f \geq -L\cdot(1 - b_f[k]) \tag{20}$$

and

$$\sum_{j=1}^{T} b_f[j] = 1 \quad , \quad b_f[j] \in \{\,0,1\} \tag{21}$$

The first four constraints force $b_f[k]$ to be 0 unless the UAV is at the final location, x_f and y_f, in which case $b_f[k]$ can be 1. The final constraint says that when the binary variable b is summed over the entire time of the flight, it must equal 1. These constraints force the UAV to the final position.

None of the constraints presented so far force the UAV to reach the final position as quickly as possible. To do this the final step is to create a definition for what makes a path optimal. This is done with MILP by creating a metric that characterizes each path and either minimizing or maximizing that metric. Note that in the constraints to force the UAV to the final position, $b_f[k]$ only equals 1 when the UAV is at the final position and that k is a time step index. The optimal path is one for which the k is the smallest. This can be represented by defining the cost as:

$$\text{cost} := \sum_{k=1}^{T}\left(b_f[k]\cdot\sum_{1}^{k} T_s\right) \tag{22}$$

Thus, the optimal path is the one that satisfies all other constraints and has the lowest cost.

B. SAM Constraints

Given a model for a SAM's ability to detect a UAV, several more constraints can be placed on the UAV's path to reduce the probability of detection to a predetermined acceptable level.

As described in the model, two important parameters in determining a SAM's impact on the UAV's path are the distance between the two and the UAV's signature to the SAM's radar. The calculations in the model for distance and azimuth are

nonlinear. The following MILP formulation is an equivalent linear approximation.

Distance between a UAV and SAM is an important factor in determining the probability of detection. The distance can be found, and then constrained, in a similar fashion as maximum velocity, using constraints and binary variables. For any time step k, where L is an arbitrarily large number the distance, $dist[k]$, is constrained as follows,

$$x[k]\sin\Big(\frac{2\pi m}{M}\Big) + y[k]\cos\Big(\frac{2\pi m}{M}\Big) \leq dist[k] \tag{23}$$

$$\Big(x[k]\sin\Big(\frac{2\pi m}{M}\Big) + y[k]\cos\Big(\frac{2\pi m}{M}\Big)\Big)\cdot 1.01 \geq dist[k] - L(1 - b_d[k,m]) \tag{24}$$

and

$$\sum_{j=1}^{M} b_d[k,j] = 1 \quad , \quad b_d[k,j] \in \{\,0,1\} \tag{25}$$

$x[k]$ and $y[k]$ are combined so that they are less than or equal to $dist[k]$ yet are greater than or equal $dist[k]$ when slightly increased, in this case by multiplying them by 1.01. Instead of 1.01, another value, such as 1.001, could be used to create a better distance approximation, but this would increase computation time. These constraints must hold for all values of m, where m ranges from 1 to M. The binary variables are used to pick out the one true value of distance that makes the two constraints true.

Another factor in determining whether a SAM detects a UAV is if the UAV is nose in or nose out relative to the SAM. Nose in refers to when the UAV's exposure to radar is minimal. Nose out refers to when the exposure is greater. This is determined by comparing the UAV's heading angle and the line of sight (LOS) angle between the SAM and the UAV. The LOS angle can be found from the x, y, and distance in the same way that heading was found from v_x, v_y, and velocity. Using (25), (26), and (27) to find the non-zero value of $b_d[k]$ the LOS is given by

$$\sum_{j=1}^{M} b_d[k,j]\cdot j \cdot \frac{360}{M} = \text{Line of Sight angle} \tag{26}$$

With the line of sight angle and the previously calculated heading angle, the azimuth angle at any time step k, can be found by comparing the two. The LOS angle and the heading angle are each one of M discrete values, where M is the size of the polygon approximation. A static M by M table can be used to find the azimuth angle as a function of both the line of sight angle and the heading angle. To illustrate, if the UAV has a heading of 0°, heading north, the azimuth table entry would be 180° if the line of sight value is also zero; the UAV is north of the SAM and heading away. However, if the LOS value is 180°, than the azimuth angle would be 0°,the UAV is south of the SAM and heading directly toward it. To access the correct table entry binary variables b_{los} and b_h are used as indices.

The next factor to consider is elevation angle between the UAV and the SAM. Since a constant altitude is assumed, the elevation angle is simply a function of range.

$$dist\cdot\tan(\text{elevation}) = \text{altitude} \tag{27}$$

However tangent is a non-linear function. Since the elevation table given in the model for SAM detection is discrete, a discrete linear approximation can be used for the tangent function in this case. For any time step, k, an index d, that ranges from 1 to the size of the tangent approximation array, and the binary variables b_{el} can be used to create the following constraints:

$$alt \leq dist[k]\cdot el_{table}[d+1] + M\cdot(1 - b_{el}[k,d]) \tag{28}$$

$$alt \geq dist[k]\cdot el_{table}[d] - M\cdot(1 - b_{el}[k,d]) \tag{29}$$

and

$$\sum_{d=1}^{el_{size}} b_{el}[k,j] = 1 \quad , \quad b_{el}[k,j] \in \{\,0,1\} \tag{30}$$

$$\sum_{d=1}^{el_{size}} b_{el}[k,d]\cdot d\cdot\frac{180}{el_{size}} = el[k] \tag{31}$$

Using elevation, azimuth, and acceptable probability of detection set by the user, the tables given in section II can be used to find the minimum safe distance between the UAV and the SAM radar. In the MILP formulation these two tables are combined into one three dimensional table. First the elevation angle is used to find the correct azimuth versus probably of detection subtable. Then the azimuth variable is used with the user supplied, discrete, probability of detection threshold variable to look up the minimum distance. The final MILP constraint is that the UAV distance from the SAM is greater than this look up distance. This constraint is consistent with the assumption that if a path violates the probability of detection threshold, a detection is assured. This simplification is necessary to eliminate the random element of SAM radar detection. Due to this assumption, for any path to be valid it must be true that for any given time the distance between the UAV and the SAM is greater than the look up distance based on all the factors.

C. Lock-Loss Constraint

The MILP formulation of lock loss assumes that the first time the UAV is detected, the lock loss timer is activated. During the lock loss time, if the vehicle continues to be detected, a SAM is fired and the UAV is destroyed. If the UAV is not detected for as long as the lock loss time, than the engage timer is reset and it takes another detection to start it again. This means that a UAV can take advantage of lock loss by flying a path that may allow it to be detected, but not for longer than the lock loss time limit. In MILP this can be specified using binary variables. Suppose the lock loss time is 4 time steps. The distance constraint is changed to include b_{ll}, binary variables for the lock loss such that at any time step k and for L being an arbitrary large number:

$$dist[k] + (L\cdot b_{ll}[k]) \geq \text{table value for distance} \tag{32}$$

If the distance at time k, is less than the table value for distance, this constraint can only be true if $b_{ll}[k]$ is 1, otherwise it can be 0. To ensure lock loss the final constraint is added, assuming the lock loss time is 4:

$$\sum_{j=0}^{3} b_{ll}[k+j] = 3 \quad , \quad b_{ll}[k+j] \in \{\,0,1\} \tag{33}$$

Which states that for any time, k, and the next 3 consecutive time steps, b_{ll} must be equal to zero once. This can only happen if for any 4 consecutive time steps, the distance between the UAV

and the SAM is greater than the table value at least once. This ensures that the UAV is not detected for any longer than the lock loss time.

IV. Implementation

The AMPL modeling language is well suited for representing the optimizations presented above. With AMPL two files are used. The model file details the various constraints, while another file holds the model data. Thus data can be changed, such as starting point or UAV velocity, without changing the constraints. The optimization problem coded in the two AMPL files is solved with ILOG CPLEX [11]. The output values are saved to a third file, which can be opened in Matlab for data analysis. The following examples were solved on a 1.8 GHz Pentium II computer with 512 MB of RAM, running the Windows XP operating system.

V. Examples

A. Single UAV vs. Single SAM

Using the data in Tables I and II a single UAV was started at position x_o = 350 and y_o = 125. The UAV was given a final destination of x_f = 340 and y_f = 0. A single SAM was placed at the origin and an acceptable probability of detection of 0.5 was used. The MILP formulation of this path planning problem contains 15363 binary variables, 72 linear variables and 10359 linear constraints. As shown in Fig. 4, the UAV does not fly directly to the final position, but instead must stay far enough away from the SAM until it reaches a position where it can turn toward the SAM site and approach the final position. The circle in the figure represents the minimum distance the UAV must maintain if it has a high signature, corresponding to a nose out orientation. The circle appears elliptical due to the unequal scaling of x and y. The UAV only turns toward the destination when doing so will provide a small cross section to the SAM's radar, allowing it to get closer without violating the acceptable probability of detection.

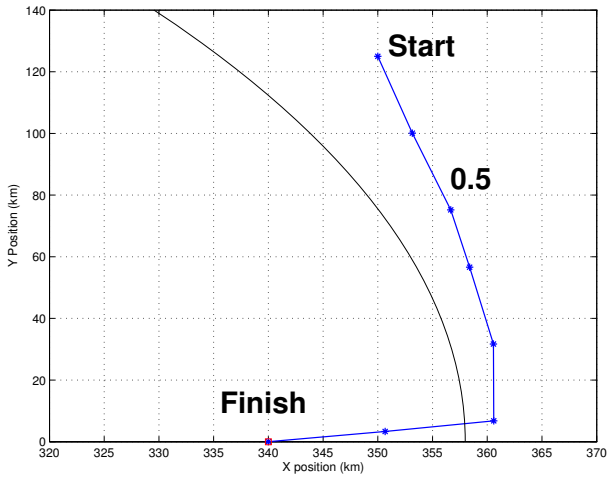

Fig. 4. UAV path, SAM at origin and Probability of Detection = 0.5

For size comparison, a two UAV scenario, with the second UAV at x_o = -350 and y_o = 125, resulted in a MILP formulation containing 25037 binary variables, 116 linear variables and 14354 linear constraints.

B. Effects of Probability of Detection

By varying the acceptable probability of detection, a more direct path plan can be generated. Two additional path plans were calculated using all the same parameters as the example given above, except that the probability of detection threshold values of 0.6 and 0.7 were used. All three paths are plotted in Fig. 5, which clearly shows that when a higher risk of detection is acceptable, the UAV can use a more direct and risky path. The additional circles correspond to the smaller minimum distance for high signature approaches. The smaller the probability threshold, the closer the UAV can be to the SAM at the origin while presenting it with a high signature.

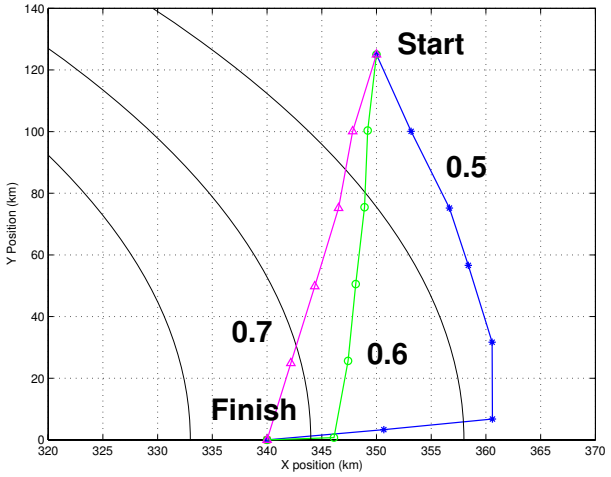

Fig. 5. UAV paths, SAM at origin and Probability of Detection = 0.5, 0.6, and 0.7

C. Calculation Time

Each path given in Fig. 5 took approximately 3 minutes to compute. This time was needed by the CPLEX solver to find all feasible solutions. Due to the large and complex nature of the MILP formulation there are many branches for the solver to traverse in search of the optimal solution.

D. Effects of Lock Loss

Shown in Fig. 6 is the same path plan from Fig. 4, with a 0.5 probability of detection. Also shown is another path that takes advantage of the lock loss, where the lock loss time is set to four time steps. This means as long as the UAV is not detected for four consecutive time steps, the UAV will not be destroyed. Due to the extra complexity that the lock loss modeling adds to the MILP formulation the computation time increased to approximately 3.5 minutes.

VI. Conclusion

As the use of UAVs increases, so does the importance of efficient path plans. In complex situations, such as the threat of a SAM, finding a feasible path plan can be difficult, knowing it is also optimal, even more so. The MILP approach outlined in this paper finds the most efficient path. Due to the computation time, the approach outlined above is best suited as a top level path planning algorithm. It may also be possible to use this algorithm in a receding horizon fashion, where new path plans are generated for shorter intermediary waypoints and recalculated as new sensor information is available.

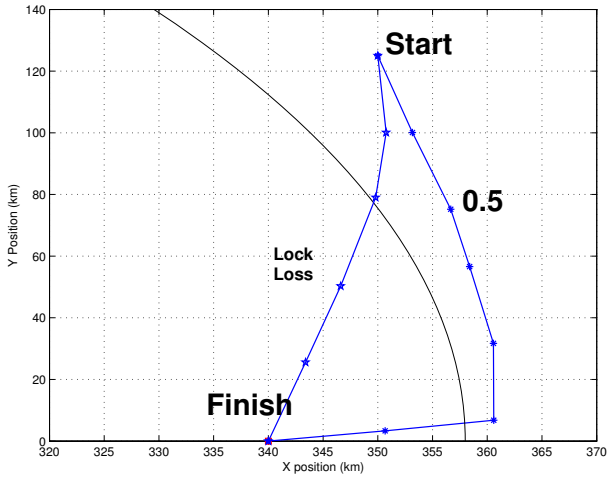

Fig. 6. UAV paths, lock loss and non-lock loss with Probability of Detection = 0.5, SAM at origin.

VII. ACKNOWLEDGMENTS

This research is based upon work supported by the DARPA Advanced Research Projects, Information Exploitation Office (DARPA/IXO) and the United States Air Force Research Laboratory under Contract No. F33615-01-C-3149. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the United States Air Force. The authors gratefully acknowledge the helpful comments of the reviewers and DARPA and AFRL management.

REFERENCES

[1] M. G. Earl and R. D'Andrea, A Study in Cooperative Control: The RoboFlag Drill, *in Proc. American Control Conference, Anchorage, AK*, 2002

[2] A. G. Richards and J. P. How, Aircraft Trajectory Planning With Collision Avoidance Using Mixed Integer Linear Programming,*in Proc. American Control Conference, Anchorage, AK*, 2002

[3] M. G. Earl and R. D'Andrea, Modeling and Control of a Multi-Agent System Using Mixed Integer Linear Programming,*IEEE CDC*, Los Vegas, Nevada, Dec. 2002

[4] Y. Kuwata, Real-time Trajectory Design for Unmanned Aerial vehicles using Receding Horizon Control,*Master's thesis*, Massachusetts Institute of Technology, June 2003

[5] C. Tomlin, G. J. Pappas, and S. Sastry, Conflict Resolution for Air Traffic Management: A Case Study in Multi-Agent Hybrid Systems,*IEEE Transactions on Autonomous Control*, vol. 43, no. 4, pp. 509-521, April 1998.

[6] A. Bemporad, F. Borrelli, and M. Morari, Piecewise Linear Optimal Controllers for Hybrid Systems, *American Control Conference*, Chicago, USA, Vol. 2, pp. 1190-1194, June 2001.

[7] A. G. Richards, J. P. How, T. Schouwenaars, and E. Feron, Plume Avoidance Maneuver Planning Using Mixed Integer Linear Programming,it AIAA Guidance, Navigation and Control Conference, Montreal, August 2001.

[8] K. Misovec, T. Inanc, J. Wohletz, R.M. Murray, Low Observable Nonlinear Trajectory Generation for Unmanned Air Vehicles,*IEEE CDC*, Maui, Hawaii, Dec. 2003

[9] J Kim, J Hespanha, Discrete Approximations to Continuous Shortest-Path: Application to Minimum-Risk Path Planning for Groups of UAVs, *IEE CDC*, Maui, Hawaii, Dec. 2003

[10] J. Knutti and D. Corman, OEP documentation, The Boeing Company, 2002

[11] ILOG, Inc. CPLEX 7.1 http://www.ilog.com/products/cplex/