Abstract—For many navigation scenarios, it is known that the accuracy of navigation state estimates depends on the path traveled, particularly when access to external navigation aids such as the Global Positioning System (GPS) is not available. In this work, we present a path planning method that attempts to minimize the navigation uncertainty of a pair of autonomous vehicles traveling from known initial locations to desired goal locations. For the scenario considered in this study, each vehicle has an onboard odometer for measuring relative changes in position and heading. The vehicles also have sensors for measuring the range between the vehicles. Navigation state estimates are obtained using a centralized batch factor graph approach implemented with the Georgia Tech Smoothing and Mapping (GTSAM) library.

In previous work on this problem, candidate vehicle trajectories were chosen from a restricted class of trajectories referred to as pseudo-zigzagging. An exhaustive search of all possible trajectories in this class revealed that the final position uncertainty could be reduced by a factor of 5 when compared with the case of each vehicle traveling straight to its goal location. In the work we present here, the trajectories are no longer restricted to such a limited class. Instead, each path is constructed from a small set of waypoints that may be placed anywhere. We have devised a method that takes an arbitrary set of waypoint locations and adjusts their locations so that the resulting path meets travel time and maneuverability constraints. Using just 3 waypoints for each vehicle path, and applying a simple random search optimization algorithm to the waypoint locations, the final position uncertainty can be reduced by another factor of 3. Thus, we achieve a total factor of 15 reduction in position uncertainty when compared to the straight path (i.e. worst) case. Also, the results indicate that the final position uncertainty can be different for each vehicle. Furthermore, the navigation certainty does not necessarily improve with increased travel time. Thus, travel time should be considered a free parameter (i.e. the time constraint is not necessarily an active constraint).

As is the case for most practical nonlinear optimization problems, there is no guarantee that the results obtained in this work are globally optimal. Nevertheless, it is quite clear that path planning can be used to significantly reduce navigation uncertainty for the scenario under consideration.

I. INTRODUCTION

For many navigation scenarios, it is known that the accuracy of navigation state estimates depends on the path traveled. For example, it has been observed that for a single vehicle using an inertial measurement unit (IMU) and a camera to travel autonomously to a goal, sinusoidal trajectories result in a more accurate navigation solution than straight trajectories [4]. In fact, navigation accuracy tends to improve with increasing frequency and amplitude of the trajectory oscillations [4]. However, there are practical limits to the maneuverability that a vehicle can achieve. In a subsequent study, it was shown that a sinusoidal trajectory can be constructed that satisfies vehicle motion and travel time constraints and minimizes navigation uncertainty [7]. The unique combination of frequency and amplitude that achieves the best performance depends on the relative accuracy of the navigation sensors. High amplitude, low frequency trajectories are best when the IMU is of relatively high quality compared to the camera, and vice versa.

We refer to the concept of planning and executing a path (as realized by the guidance and control systems of a vehicle) that minimizes position uncertainty (i.e. maximizes estimation performance) as integrated control and estimation. In addition to the scenario above, integrated control and estimation can be applied to a single vehicle using wheel odometry and a camera to navigate. Paths that minimize navigation uncertainty for this problem have been discovered using a particle swarm optimization approach [3]. Even for GPS-aided inertial navigation, it has been observed that certain types of paths can help reduce the uncertainty of various navigation states [1].

Integated control and estimation can also be applied to cooperative, multi-vehicle navigation scenarios. One such scenario that has been studied is the case of two ground vehicles autonomously navigating to goal locations using wheel odometry and a range measurement between the vehicles [8]. In this work, the vehicle paths were restricted to a class of paths described as pseudo zigzagging and defined by a set of waypoints. At each waypoint, the vehicle was allowed to travel in only one of two possible directions, thereby allowing only two possible choices for the next waypoint location. The path for each vehicle was constructed using 10 waypoints, which, when considering the additional constraint that the last waypoint must be placed at the goal location, resulted in 252 possible paths for each vehicle. Optimal paths were found using an exhaustive search of all 63504 possible pairs of trajectories. Using this approach, the standard deviation of the estimated final position error was reduced by a factor of 5 compared to the case where the vehicles travel straight to their goal locations.

The work presented here expands on the previous work of [8] by allowing a more general class of vehicle paths to be studied. Specifically, each vehicle is allowed to travel in any direction after reaching a waypoint, given that a direction is chosen such that the vehicle can still reach the goal in the allotted time. Since it is generally not possible to perform an exhaustive search of all possible paths, a simple random
search algorithm was implemented. Furthermore, this work examines how navigation performance depends on the allotted travel time.

II. METHODS

The problem scenario is shown in Figure 1. Two vehicles are constrained to move in a 2D plane, and the goal is for each vehicle to reach its goal location as accurately as possible within an allotted time. Each vehicle has wheel encoders that detect changes in heading and position of the vehicle as it moves. The vehicles also have sensors for measuring the range between the vehicles. Both measurement types are corrupted by a small amount of noise, thus the true state (i.e. position and heading) of the vehicles is never known exactly.

Fig. 1. Problem scenario. The goal is for each vehicle to reach its target as accurately as possible within an allotted time using onboard odometers and an inter-vehicle range sensor.

A. Vehicle Kinematics

The vehicle kinematics are modeled using a simple unicycle model with no sideslip. In continuous time, this model is as follows, where \( x \) and \( y \) are the position components of the vehicle relative to a fixed inertial reference frame, and \( \theta \) represents the vehicle heading.

\[
\dot{x} = v \cos(\theta) \tag{2}
\]
\[
\dot{y} = v \sin(\theta) \tag{3}
\]

If it is assumed that the turn rate of the vehicle, \( \omega \), and the speed, \( v \), are constant during a sample period of duration \( \Delta t \), then the vehicle kinematics can be described analytically in discrete time [6]. The following equations describe the discrete time kinematics at timesteps indexed by \( k \).

\[
\theta_k = \theta_{k-1} + \omega_{k-1} \Delta t \tag{4}
\]
\[
x_k = x_{k-1} + v_{k-1} \cos \left( \theta_{k-1} + \frac{\omega_{k-1} \Delta t}{2} \right) \tag{5}
\]
\[
y_k = y_{k-1} + v_{k-1} \sin \left( \theta_{k-1} + \frac{\omega_{k-1} \Delta t}{2} \right) \tag{6}
\]

In the above equations, \( v \) is the magnitude of the displacement of the vehicle between successive timesteps, as defined by (7), and \( \text{sinc}(q) \) is defined by (8) for a generic argument \( q \).

\[
\nu_{k-1} = v_{k-1} \Delta t \text{sinc} \left( \frac{\omega_{k-1} \Delta t}{2} \right) \tag{7}
\]
\[
\text{sinc}(q) = \begin{cases} \sin(q)/q & \text{if } q \neq 0 \\ 1 & \text{otherwise} \end{cases} \tag{8}
\]

B. Measurement Models

Wheel encoders measure the rotation of the wheels as the vehicle moves. These measurements can be used to derive changes in heading and position of the vehicle during a sample period. To avoid an unnecessarily detailed model of the wheel encoders, we simply assume that these sensors measure vehicle speed and turn rate corrupted by a small amount of zero-mean Gaussian noise. Expressions for the measured values of turn rate, \( \tilde{\omega} \), and speed, \( \tilde{v} \), are given below, where \( \sigma_\omega \) and \( \sigma_v \) are the standard deviations of turn rate and speed, respectively, and \( N(\mu, \sigma^2) \) represents the normal distribution with mean \( \mu \) and standard deviation \( \sigma \) (variance of \( \sigma^2 \)).

\[
\tilde{\omega}_{k-1} = \omega_{k-1} + \eta_\omega \quad \eta_\omega \sim N(0, \sigma^2_\omega) \tag{9}
\]
\[
\tilde{v}_{k-1} = v_{k-1} + \eta_v \quad \eta_v \sim N(0, \sigma^2_v) \tag{10}
\]

These measurements are used to formulate odometry measurements (i.e. discrete changes in heading and position relative to the body fixed frame at timestep \( k - 1 \)) as follows, where \( \tilde{v}_{k-1} \) is computed using (7) by replacing \( \omega \) and \( v \) with \( \tilde{\omega} \) and \( \tilde{v} \), respectively.

\[
\tilde{\Delta} \theta_k = \tilde{\omega}_{k-1} \Delta t \tag{11}
\]
\[
\tilde{\Delta} x_k^b = \tilde{v}_{k-1} \cos(\tilde{\Delta} \theta_k/2) \tag{12}
\]
\[
\tilde{\Delta} y_k^b = \tilde{v}_{k-1} \sin(\tilde{\Delta} \theta_k/2) \tag{13}
\]

For input to the navigation state estimator (described in section II-D), odometry noise is approximated from the wheel encoder noise in the following manner, and it is assumed the noise is unbiased (i.e. zero mean) for each component of odometry.

\[
\sigma_{\Delta \theta}^b = \sigma_\omega \Delta t \tag{14}
\]
\[
\sigma_{\Delta x}^b = \sigma_v \Delta t \tag{15}
\]
\[
\sigma_{\Delta y}^b = 0 \tag{16}
\]

A more precise formulation of the wheel odometry noise could be derived from the odometry equation, but it seems to make little practical significance. Note that the standard
deviations of the odometry components are given in the body frame, as required by the navigation state estimator (see section II-D). When transformed to the global frame, the standard deviations of all odometry components will usually all be nonzero.

A range sensor also measures the range, ρ, between the vehicles, where the subscripts indicate the vehicle number.

\[ \dot{\rho} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} + \eta_\rho \quad \eta_\rho \sim N(0, \sigma_\rho^2) \]  

\[ (17) \]

C. Path Planning and Guidance

The trajectory of each vehicle is defined by waypoints denoted by \( w_{ij} \), where \( i \) is the vehicle number and \( j \) is the waypoint number. It is convenient to define the waypoints in a scaled and rotated coordinate system, which we refer to as the trajectory frame. This coordinate system is centered at the initial vehicle location and scaled by the distance, \( d_T \), from the initial vehicle location, \( p_{i0} = [x_{i0} \ y_{i0}]^T \) to the target location \( p_{iT} = [x_{iT} \ y_{iT}]^T \). Thus, in the trajectory frame, the initial vehicle location is \([0 \ 0]^T\) and the target location is \([1 \ 0]^T\).

The notation \( \bar{w}_{ij} \) is used to indicate a normalized interior waypoint, and \( W^- \) indicates the ordered set of all interior waypoints for all vehicles. The components of the waypoints in \( W^- \) are defined in the trajectory frame and normalized with respect to \( d_T \), and they are referred to as interior because they do not include the initial or target location. The interior waypoints are distinguished from the full set because later in section II-E, the initial and target waypoints will remain fixed while the interior waypoints will be adjusted to solve an optimization problem. The full set of normalized waypoints, with elements denoted by \( \bar{w}_{ij} \), is created simply by including the initial and target locations as in

\[ \bar{w}_{ij} \in W^- \]  

\[ (18) \]

\[ \bar{w}_{ij} = \begin{cases} 
[0 \ 0]^T & \text{if } j = 0 \\
\bar{w}^-_{ij} & \text{if } 1 \leq j < n \\
[1 \ 0]^T & \text{if } j = n 
\end{cases} \]  

\[ (19) \]

Normalized waypoints in the trajectory frame are transformed into waypoints in the global frame using (20), where \( C \) is a combined rotation and scaling matrix as defined in (21), and \( S \) is an additional scaling matrix defined in (22) that effectively acts only on the \( y \) component of each waypoint and is used for the path optimization described in section II-E. In particular, the parameter \( s \) is used to create a waypoint plan from a normalized set of interior waypoints such that the travel time to the target is within the allotted time.

\[ w_{ij} = C_i S_i \bar{w}_{ij} + p_{i0} \]  

\[ (20) \]

\[ C_i = \begin{bmatrix} x_{iT} - x_{i0} & -(y_{iT} - y_{i0}) \\ y_{iT} - y_{i0} & x_{iT} - x_{i0} \end{bmatrix} \]  

\[ (21) \]

\[ S_i = \begin{bmatrix} 1 & 0 \\ 0 & s_i \end{bmatrix} \]  

\[ (22) \]

The turn rate command is generated by a simple proportional law on the heading error using (23), where \( K \) is the proportional gain on turn rate, \( \omega_{\text{max}} \) is the magnitude of the maximum allowable turn rate, and \( \beta \) is the angle between the vehicle’s velocity and the relative position vector from the vehicle to the current waypoint.

\[ \omega = \begin{cases} 
-\omega_{\text{max}} & \text{if } K\beta < -\omega_{\text{max}} \\
K\beta & \text{if } |K\beta| \leq \omega_{\text{max}} \\
\omega_{\text{max}} & \text{if } K\beta > \omega_{\text{max}} 
\end{cases} \]  

\[ (23) \]

Once a vehicle becomes closer to a waypoint than the waypoint critical distance, \( r_w \), the current waypoint is updated to the next waypoint in the list. Perfect tracking of the turn rate command is assumed.

Since the turn rate of the vehicle is under digital control, the heading error can be corrected in a single time step if \( K = 1/\Delta t \). This is what we use throughout the paper. Note that for stability, \( K \) must be less than \( 2/\Delta t \). Also, notice that the vehicle will turn at the maximum turn rate magnitude unless \( |\beta| \leq \omega_{\text{max}}/K \) (i.e. when the magnitude of the waypoint relative bearing is fairly small). Thus, most of the time, the vehicle is either driving (nearly) straight, turning to the right at maximum turn rate, or turning to the left at maximum turn rate.

It was apparent while developing the path optimization procedure in section II-E that the waypoint guidance law needed to be modified slightly to help avoid the case where the vehicle will continuously loop around a waypoint without actually passing through the waypoints. An additional condition was added such that the vehicle is considered to have visited a waypoint if the vehicle crosses the boundary defined by a line perpendicular to the line connecting the current waypoint to the previous waypoint (represented by the vector \( \Delta w \)). Mathematically, if (24) is satisfied, the current waypoint is updated to the next waypoint in the list.

\[ (w - p)^T \Delta w < 0 \]  

\[ (24) \]

\[ \Delta w_{ij} = w_{ij} - w_{i(j-1)} \]  

\[ (25) \]

\[ w_{i0} = p_{i0} \]  

\[ (26) \]

Here, \( p \) is the position vector of the vehicle, \( w \) is the position vector of the waypoint, and \( \Delta w \) is a vector pointing from the previous waypoint to the current waypoint.

Since the vehicle must actually intercept the goal location at the final waypoint, the half-plane condition in (24) cannot be used at the last waypoint. However, the algorithm must still protect against the case where the vehicle could end up in an infinite loop around the final waypoint. To avoid this situation, it is checked if the final waypoint is inside the current turn radius of the vehicle. If (27) is satisfied, then the vehicle is commanded to travel straight until the waypoint is no longer within the turn radius of the vehicle. Then, the vehicle turns toward the final waypoint until intercept.
\[ \|p \pm (v_{\perp}/v)r_{\min} - w\| \leq r_{\min} \quad (27) \]
\[ v_{\perp} = [-v_y \ v_x]^T \quad (28) \]
\[ r_{\min} = v/\omega_{\max} \quad (29) \]

Even with implementing the half-plane condition of (24) and the turn radius condition of (27), there are still cases where the travel time of the vehicle will not converge on the desired travel time. In other words, the travel time will be shorter than the desired time for a given value of the scaling parameter \( s \) in (22), and longer than the desired travel time for an infinitesimally larger value of \( s \). Several of the alternative conditions for incrementing the current waypoint given in [2] also exhibit this bifurcation behavior in certain cases. This bifurcation is difficult to avoid in general. It seems difficult, and perhaps impossible, to construct a guidance law that will avoid bifurcation without moving one or more of the waypoints independently of the rescaling operation. The current workaround for this situation is to implement a timeout on the number of iterations used to solve for \( s \). If the scaling parameter is updated more than 30 times and the trajectory still does not satisfy the travel time constraint, then the trajectory is set to a case that is known to perform poorly so that the optimization algorithm will essentially reject it as a useful candidate solution.

D. State Estimation

The state estimator generates an estimate of the position and orientation of a vehicle, as well as an estimate of the uncertainty in the state estimate. In this work, we implement a centralized state estimator, meaning that the estimator has knowledge of all the measurements taken by all the vehicles. The estimated covariance matrix generated by the state estimator is used to formulate the cost function considered by the path optimization routine in section II-E.

For this work, we use the Georgia Tech Smoothing and Mapping (GTSAM) library to construct a state estimator. GTSAM implements a factor graph approach to estimation over a batch of sensor measurements. For the cooperative navigation scenario considered in this work, this estimator gives more consistent results than a standard Extended Kalman Filter (EKF). In other words, the uncertainty reported by GTSAM tends to agree with Monte-Carlo simulation, whereas an EKF tends to underestimate the uncertainty. This is not surprising, since GTSAM also tends be more consistent for vision-aided navigation [10].

E. Path Optimization

In general, the problem of constructing a set of trajectories that minimize position estimation uncertainty while satisfying vehicle motion constraints can be posed as a nonlinear constrained optimization problem. The objective function must be a scalar value, but the uncertainty of each vehicle is typically characterized by an uncertainty ellipse (defined in 2D by at least 3 parameters), and there is more than one vehicle. Thus, there are several ways that one can choose to design the objective function for each vehicle, and for the group of vehicles, and the chosen method can have a significant effect on the resulting trajectory shapes (results not shown). For each vehicle, we choose to define the position uncertainty (i.e. the uncertainty between the true and estimated vehicle location) as the median position estimation error, meaning the final position error will be less than this number 50% of the time. We refrain from calling this circular error probable (CEP) since CEP is often used to statistically characterize the actual miss distance (i.e. the distance between the actual vehicle position and a desired vehicle position). Another option would be to use the root sum of squares of the position error components, but this has a less intuitive geometric interpretation.

The objective function \( J_i \) for each vehicle \( i \) is the median position error as computed from the estimated final covariance matrix, \( \hat{P}_i \), given by the state estimator. Using the exact formulation given in [9], the median position error is defined according to (30).

\[ J_i = \sqrt{\frac{\chi^2}{\nu}} \quad (30) \]
\[ \hat{P}_i = VA V^T \quad (31) \]
\[ \chi^2 = 2\Gamma^{-1}(0.5, \nu/2) \quad (32) \]

In (32), \( \Gamma^{-1} \) is the functional inverse of the incomplete Gamma function, and

\[ \nu = \frac{\text{tr}(\Lambda)^2}{\text{tr}(\Lambda^2)} \]

In the above, \( \hat{P}_i \) is the \( 2 \times 2 \) estimated covariance matrix of the x and y components of vehicle \( i \), and \( \Lambda \) is a diagonal matrix containing the eigenvalues of the matrix \( \hat{P}_i \).

For the group of vehicles, one way to define the position uncertainty is to take the mean position error over all the vehicles. Another more greedy method is to simply define the error of the group as the error of the vehicle with the smallest error, as follows.

\[ J = \min J_i \quad (33) \]

This is the objective function we use in the results. A third possibility would be to find a trajectory set that minimizes the maximum position uncertainty over all the vehicles (i.e. the minimax solution, \( J = \max J_i \)).

In this work, we use a simple stochastic search algorithm to design trajectories that minimize the objective function. The algorithm works by solving a zero-finding (i.e. root solving) problem (35) nested inside a minimization problem (34). Note that we use the nonstandard notation “argzero” to imply the argument that solves the zero-finding problem. For a set of normalized interior waypoints, \( \mathcal{W}^- \), the scaling parameter \( s \)
is solved using a bisection search to satisfy the travel time constraint, where \( t_f \) is the travel time for a given scaling parameter and normalized interior waypoint set, and \( t_{max} \) is the allotted travel time. To solve (34), the interior waypoints are perturbed a small random amount at each iteration of the optimization routine. The currently best known set of waypoints (i.e. the set of waypoints that gives the lowest value of \( J \) so far) is used as the starting point for the perturbation. Thus, if a better set of waypoints is found on a particular iteration, it becomes the new best set, otherwise, the optimization routine continues perturbing from the best known set until a better set is found.

\[
W^* = \arg\min_{W^-} J(W^-, s^*(W^-)) \tag{34}
\]

\[
s^* = \arg\min_{s} t_f(W^-, s) - t_{max} \tag{35}
\]

III. RESULTS

Unless otherwise stated, the parameters listed in Table I are used throughout the results section.

TABLE I
SIMULATION PARAMETERS.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>vehicle speed ((v))</td>
<td>30 m/s</td>
</tr>
<tr>
<td>max turn rate ((\omega_{max}))</td>
<td>5 degrees/s</td>
</tr>
<tr>
<td>initial target distance ((d_f))</td>
<td>4 km</td>
</tr>
<tr>
<td>allotted flight time ((t_{max}))</td>
<td>150 seconds</td>
</tr>
<tr>
<td>waypoint critical distance ((r_w))</td>
<td>200 meters</td>
</tr>
<tr>
<td>odometry measurement frequency</td>
<td>10 Hz</td>
</tr>
<tr>
<td>speed measurement standard deviation ((\sigma_v))</td>
<td>0.01 m/s</td>
</tr>
<tr>
<td>turn rate measurement standard deviation ((\sigma_\omega))</td>
<td>0.57 degrees/s</td>
</tr>
<tr>
<td>range measurement frequency</td>
<td>10 Hz</td>
</tr>
<tr>
<td>range measurement standard deviation ((\sigma_\rho))</td>
<td>0.01 meters</td>
</tr>
</tbody>
</table>

As a baseline for comparison, the position estimation performance of the vehicles traveling straight to their goals is studied. Note that in this case, the allotted travel time \( t_{max} \) is only 133.3 seconds. The resulting trajectories and final position covariance ellipses are shown in Figure 2. The covariance ellipses are shown at the 20\(\sigma\) level so they can be easily seen in the figure and in results presented later in this section. For this case, the median position error of both vehicles is the same at 40.18 meters.

Next, we repeat the best case that was discovered in [8] using an exhaustive search of pseudo zigzagging trajectories. Note that for this case, the maximum turn rate was set to 57 degrees/s and the waypoint critical distance \( r_w \) was set to 30 meters. The results are shown in Figure 3. Here, the median position error ranges between 9.13 and 9.23 meters for the two vehicles. This is a factor of 4.4 reduction in median position error compared to the straight line case. In [8], a factor of 5 reduction was reported, but a different method of quantifying the position uncertainty was used.

Next, using the parameters in Table I, only 3 waypoints per vehicle, and the optimization procedure described in section II-E, we discovered the trajectories shown in Figure 4. For this case, vehicle 1 had a median position uncertainty of 3.78 meters, and vehicle 2 had a median position uncertainty of 6.73 meters. Thus, for the better of the two vehicles, this results in an additional factor of 2.4 reduction in median position error compared to the best case from [8].

Another run of the optimization procedure with the exact same parameters produced the results shown in Figure 5. Here, the resulting trajectories are qualitatively quite different than those of Figure 4. For this case, vehicle 1 had a median
position error of 4.85 meters and vehicle 2 had a median position error of 3.97 meters. Thus, the better of the two vehicles performed slightly worse in this case than in the previous case, but the worse of the two vehicles performed better than in the previous case. In general, all of the optimization attempts tended to converge on trajectory sets similar to either Figure 4 or Figure 5.

As a final experiment, the allotted flight time was varied over a range from 135 seconds to 160 seconds. Interestingly, with $t_{\text{max}}$ set to 140 seconds, an even better result can be achieved than with $t_{\text{max}}$ set to 150 seconds. Here, vehicle 1 had a median position error of just 2.75 meters, while vehicle 2 had a median position error of 5.54 meters. Overall, this is a factor of 14.6 reduction in position uncertainty compared with the straight line case and a factor of 3.3 reduction compared to the best case from [8].

A. Trajectory Bifurcation

As stated in the Methods, there are certain instances when the trajectory exhibits a bifurcation (i.e. a large change in behavior for a small change in a parameter). One such example is shown in Figure 7. This particular case of trajectory bifurcation was observed during the optimization process which eventually led to the solution for vehicle 1 in Figure 4.
increased by only $3 \times 10^{-10}$ causing waypoint 2 to move a distance of only $3 \times 10^{-6}$ meters (i.e. 3 microns), once the vehicle approaches within the critical distance of waypoint 2, it cannot reach waypoint 3 even at maximum turn rate. Thus, the vehicle satisfies condition (27) whereby it travels straight until the final waypoint is within its turn radius. This leads to an increase of 18.4 seconds in travel time and violation of the travel time constraint of 150 seconds. Interestingly, if $s$ is increased by $6.5 \times 10^{-5}$ causing waypoint 2 to move a distance of 0.7 meters, the vehicle never gets within the critical distance of waypoint 2. Instead, the vehicle continues to circle waypoint 2 until condition (24) is met (i.e. it crosses the half-plane boundary defined by waypoints 1 and 2), at which point it then heads to waypoint 3. Even though $s$ is larger in this case, the resulting flight time is 4 seconds shorter. Clearly, this is an extremely sensitive trajectory plan, and in practice, it would be impossible to predict which of the 3 paths the vehicle would actually follow, especially when unmodeled dynamics such as winds will have an impact on the trajectory.

IV. CONCLUSION

We have presented a method of using path planning to reduce navigation position uncertainty. For the cooperative navigation scenario considered in this work, the method resulted in nearly a factor of 15 reduction in vehicle position uncertainty when compared to the case of the vehicles traveling straight to their goal locations. This is an additional factor of 3 reduction in uncertainty compared to previous results. The method is general enough to be applied to other integrated control and estimation problems. All that is required is the selection of a state estimator for the problem of interest, and a well defined objective function for the position uncertainty.

ACKNOWLEDGMENT

We thank Dr. Robert Murphey for his suggestion to use the half-plane condition defined in (24) to avoid continuously looping around a waypoint. We also thank Andrew Kondrath from the Air Force Institute of Technology for finding a cleaner representation of the vehicle kinematics using the sinc function, which avoids needing special cases for whether or not the vehicle is turning.

REFERENCES