

Sensor Placement for Improved Robotic Navigation

Michael P. Vitus

Department of Aeronautics and Astronautics
Stanford University
Email: vitus@stanford.edu

Claire J. Tomlin

Department of Electrical Engineering and Computer Sciences
University of California at Berkeley
Email: tomlin@eecs.berkeley.edu

Abstract—Consider a set of fixed sensors used to estimate the state of a vehicle (e.g. position, orientation and velocity) while it attempts to follow a pre-planned trajectory. Since the sensor can only provide a measurement to the vehicle when it is within range, the deployment of the sensors will have a major impact on the ability of the vehicle to follow the trajectory. The problem addressed here is to optimally place the sensors in the environment such that the weighted function of the estimation error at each time-step is minimized. An optimization formulation is proposed that accounts for the uncertainty of the vehicle’s state in determining whether it can receive a measurement from a sensor. A confidence level is introduced as a tuning parameter that controls the conservativeness of the solution. Consequently, the resulting solution increases the likelihood of the vehicle successfully following its intended trajectory. Finally, due to the interdependence between the sensors’ positions, a novel incremental optimization algorithm is presented which significantly outperforms a standard nonlinear optimization procedure. Experimental and simulation results are shown which characterize the performance of the proposed algorithm.

I. INTRODUCTION

Effective sensor deployment has most notably been studied with application to the Global Positioning System (GPS) [1]. In this system, the satellites effectively provide range measurements to the end user which are used to triangulate their position. The satellites’ configuration heavily impacts the quality of the estimate of the user’s position and has been extensively studied [2] [3].

Optimal sensor deployment strategies have also been studied for improving robotic localization. For example, Jourdan et al. [4] considered the case of deploying a network of static sensors that provide range measurements to the agent for localization. Most notably, they developed a locally optimal algorithm, significantly outperformed Simulated Annealing, to position the sensors on the boundaries of buildings to minimize the average position error bound over multiple agent locations. Zhang [5] examined the optimal orientation of sensors in 2D where the sensors can have different but constant measurement variances. A necessary condition was derived for the optimal orientation through minimizing the joint covariance matrix. This condition was then used to develop an $M - 3$ step algorithm that converged to the globally optimal solution, where M is the number of sensors. Zhang’s formulation, however, does not consider the position of the sensors which could have a major impact on the quality of the solution.

A related problem to optimal sensor deployment is the generation of trajectories for mobile sensor platforms to improve target localization. In [6], the authors studied the case of an unmanned aerial vehicle (UAV) taking 3D-bearings-only measurements for the task of target localization. They setup an optimization problem that optimized the trajectory of the

UAV to enhance the estimation performance characterized by the Fisher Information Matrix (FIM). Similarly, Sinha et al. [7] maximized the FIM as well as survivability and detection probability for a group of UAVs performing surveillance of several ground targets. Their solution technique used a gradient descent algorithm coupled with a genetic algorithm to search for the global minimum. Martinez et al. [8] investigated sensor placement and motion coordination strategies through the determinant of the FIM and characterized the global minima in the 2D case. They used the results to develop a motion coordination algorithm to dynamically control the mobile sensor network to an optimal deployment around the target.

Finally, the idea of using sensor optimization for simultaneous localization and mapping (SLAM) has also been considered. Strasdat et al. [9] explored the problem of SLAM for computational and/or memory limited systems, which can incorporate only a limited number of landmarks. Specifically, they proposed a landmark selection policy that identifies which landmarks are valuable for the robot to efficiently complete its navigation tasks. Through simulations and experimental demonstrations, they showed their algorithm outperformed handcrafted heuristics.

The sensor deployment problem considered in this paper is most similar to those considered in [4] and [5]. The problem involves a vehicle attempting to follow a pre-planned trajectory through the environment. Although [4] briefly touched on incorporating uncertainty, the proposed work presents a unifying framework for handling uncertainty of the vehicle’s execution of the pre-planned trajectory. To increase the vehicle’s likelihood of successfully following the trajectory and reaching its goal location, sensors are deployed in the environment to provide the vehicle with measurements when within range. A novel solution to this sensor deployment problem is presented which is applicable to any linear system with linear measurements.

There are many motivating examples for this work. One example is the placement of sensors to minimize the uncertainty of a vehicle following a pre-planned trajectory. There are several applications for which a pre-planned trajectory exists and is repeatedly used including automated supply chains, autonomous construction and disaster site cleanup. In these applications, the proposed work can provide better performance with fewer sensors than existing solutions. Another potential application is tracking building occupants’ behavior for energy efficient control.

Another class of motivating problems is that of landmark placement in an environment, with the vehicle carrying a sensor to observe the landmarks for localization. For example, this work could be applied to feature selection for SLAM

with computationally limited systems in which the exponential growth of requirements with number of landmarks is prohibitive. This problem is formulated in [9]. If the environment is dense in features, then the proposed algorithm could be applied without much change. With a small number of features, however, the expected distance to future landmarks plays a critical role and would need to be factored into the algorithm.

The work presented has several contributions. First, the sensor placement problem is formulated as an optimization program which minimizes the estimation error while the vehicle follows the pre-planned trajectory; this will maximize the likelihood of the vehicle successfully reaching its intended destination. In the original problem formulation, the discrete nature of the measurement region results in insufficient gradient information. Consequently, a continuous approximation is used that is more suitable for a numerical optimization solver. To ensure that the final solution is conservative, the uncertainty of the vehicle is taken into account when determining whether the vehicle can receive a measurement from the sensor. In this design, a confidence level is introduced as a tuning parameter that controls the degree of conservativeness of the resulting sensor deployment. Finally, an incremental optimization procedure is proposed that significantly outperforms both a simple, large nonlinear optimizer and Simulated Annealing.

The paper proceeds as follows. Section II describes the standard sensor placement problem formulation. In Section III, a continuous approximation of the measurement regions is presented. A solution that accounts for the uncertainty of the vehicle around the pre-planned trajectory is developed in Section IV. Simulation and experimental results are presented in Sections V and VI, respectively. The paper concludes with directions of future work and with an Appendix that analyzes the optimal measurement time for a simple system.

II. PROBLEM FORMULATION

Consider the following linear stochastic system defined by,

$$x(k+1) = Ax(k) + Bu(k) + w(k), \forall k \in T_N \quad (1)$$

where $x(k) \in \mathbb{R}^n$ is the state of the system, $u(k) \in \mathbb{R}^p$ is the input of the system, $w(k) \in \mathbb{R}^n$ is the process noise and $T_N = \{0, \dots, N-1\}$ is the horizon. The initial state, $x(0)$, is assumed to have a zero mean Gaussian distribution with covariance Σ_0 i.e., $x(0) \sim \mathcal{N}(0, \Sigma_0)$, and the process noise, $w(k)$, is assumed to have a zero mean Gaussian distribution, $w(k) \sim \mathcal{N}(0, \Sigma_w)$.

The system may estimate its own state using M sensors that can be placed in the environment. Let \mathbb{M} be defined as $\{1, \dots, M\}$. Each sensor, $i \in \mathbb{M}$, has position $s_i \in \mathbb{R}^d$, sensor measurements $C_i \in \mathbb{R}^{r \times n}$ and a maximum sensing radius $R_i \in \mathbb{R}$. The i^{th} sensor's measurement is defined by $y_i(k) = C_i x(k) + v_i(k)$, where $y_i(k) \in \mathbb{R}^r$ and $v_i(k) \in \mathbb{R}^r$ are the measurement output and measurement noise of the i^{th} sensor at time-step k , respectively. The measurement noise has a zero mean Gaussian distribution, $v_i(k) \sim \mathcal{N}(0, \Sigma_{v_i})$, $\forall i \in \mathbb{M}$.

By a standard result of linear estimation theory, the Kalman filter is the minimum mean square error estimator for the

system considered. Let $\hat{\Sigma}_{k|k}$ be the covariance matrix of the optimal estimate of $x(k)$ given all of the measurements up to time-step k , let $\hat{\Sigma}_{k+1|k}$ be the covariance matrix of the predicted state, $x(k+1)$, given all of the measurements up to time-step k , and let $\hat{\Omega}_{k|k} = \hat{\Sigma}_{k|k}^{-1}$ be the information form of the standard Kalman filter. The information filter form has an important advantage over the standard Kalman filter recursion in that the measurement update is a simple sum over all the measurements, but this advantage could be negated by the increased computational cost in the motion update for large dimensional systems. The information form of the estimator recursion is as follows,

$$\hat{\Omega}_{k+1|k} = (A\hat{\Omega}_{k|k}^{-1}A^T + \Sigma_w)^{-1} \quad (2)$$

$$\hat{\Omega}_{k+1|k+1} = \hat{\Omega}_{k+1|k} + \sum_{j=1}^m f_w(x(k+1), s_j, R_j, \hat{\Omega}_{k+1|k}) C_j^T \Sigma_{v_j}^{-1} C_j \quad (3)$$

with initial condition $\hat{\Omega}_{0|0} = \Sigma_0^{-1}$. The weighting function, $f_w(\cdot)$, indicates whether a sensor can provide a measurement or not. The general sensor placement problem is now stated.

<p style="margin: 0;">minimize $V(s) = \sum_{k=1}^N \text{tr}(\hat{\Omega}_{k k}^{-1})$ subject to Eqn. (2) $\forall k \in T_N$ Eqn. (3) $\forall k \in T_N$</p>	(P2.1)
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------

There are several choices for the objective function, $V(s)$. Since the problem involves minimizing a matrix, it is desirable to find a scalar function for the objective to simplify the problem without significant loss of information. Three possible functions are the determinant, the maximum eigenvalue, or the trace. The main disadvantage of the determinant is that a small determinant can correspond to a very elongated ellipse. As opposed to the maximum of the eigenvalues, the trace represents the uncertainty in all directions equally and was chosen. To ensure a fair comparison between different algorithms, the summation over all time-steps is used since it balances the uncertainty of the vehicle throughout the trajectory.

The optimization problem P2.1 only has an analytical solution for very simple systems, such as the one presented in the Appendix; consequently a numerical optimization solver must be used. To aid the numerical solver in finding a locally optimal solution, the analytical gradient can be computed for this problem as stated in Algorithm 1, where $J(j)$ is the gradient of the j -th component of the concatenated sensor positions. A similar procedure can be used to calculate the analytical Hessian. Algorithm 1 is presented in a form that is easy to read but not necessarily the most efficient.

For the remainder of the paper, the problem is specialized for the case of a vehicle with state representing position, velocity, etc. Let $p(k) \in \mathbb{R}^d \subset x(k)$ be the position of the vehicle at the k^{th} time-step. The true sensor measurement

Algorithm 1 Computation of the Gradient

```
1: for  $j = 1, \dots, m$  do
2:    $J(j) = 0, \hat{\Omega}_{0|0} = \Omega_0, \frac{\partial \hat{\Omega}_{0|0}^{-1}}{\partial s_j} = 0$ 
3:   for  $k = 1, \dots, N$  do
4:      $\bar{\Omega}_{k|k-1} = A \hat{\Omega}_{k-1|k-1}^{-1} A^T + \Sigma_w$ 
5:      $\hat{\Omega}_{k|k} = \bar{\Omega}_{k|k-1}^{-1} +$ 
6:        $\sum_{i=1}^m f_w(x(k), s_i, R_i, \hat{\Omega}_{k|k-1}) C_i^T \Sigma_{v_i}^{-1} C_i$ 
7:      $\frac{\partial \bar{\Omega}_{k|k-1}}{\partial s_j} = A \frac{\partial \hat{\Omega}_{k-1|k-1}^{-1}}{\partial s_j} A^T$ 
8:      $\frac{\partial \hat{\Omega}_{k|k}}{\partial s_j} = -\bar{\Omega}_{k|k-1}^{-1} \frac{\partial \bar{\Omega}_{k|k-1}}{\partial s_j} \bar{\Omega}_{k|k-1}^{-1} +$ 
9:        $\frac{\partial f_w(x(k), s_j, R_j, \hat{\Omega}_{k|k-1})}{\partial s_j} C_j^T \Sigma_{v_j}^{-1} C_j$ 
10:     $\frac{\partial \hat{\Omega}_{k|k}^{-1}}{\partial s_j} = -\hat{\Omega}_{k|k}^{-1} \frac{\partial \hat{\Omega}_{k|k}}{\partial s_j} \hat{\Omega}_{k|k}^{-1}$ 
11:     $J(j) = J(j) + \text{tr} \left( \frac{\partial \hat{\Omega}_{k|k}^{-1}}{\partial s_j} \right)$ 
12:   end for
13: end for
```

weighting function in Eqn. (3) is defined as,

$$f_w(x, s, R, \hat{\Omega}) = \begin{cases} 1, & \text{if } \|p - s\| \leq R \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

There are two concerns with this formulation. First, the weighting function defined in Eqn. (4) produces an ill-formed optimization problem for numerical solvers because the gradient is 0 everywhere except when $\|p - s\| = R$, where it is undefined. Consequently, an approximate weighting function will have to be devised to provide more information to the numerical optimization solver. Second, only the pre-planned trajectory, x_{plan} , is known *a-priori* and the vehicle may stray away from it if the vehicle's process noise is large. Thus the vehicle's position, p , in the true sensor measurement weighting function cannot be readily evaluated.

III. APPROXIMATING THE WEIGHTING FUNCTION

To address the lack of gradient information in the true sensor weighting function, the function has been approximated by,

$$f_w(x, s, R, \hat{\Omega}) \approx g_w(d) = 1 - \frac{1}{1 + \exp(-(\alpha d + \beta))} \quad (5)$$

where $d = \|p - s\|$ is the distance between the sensor and the vehicle. The function is a transformed sigmoid decay function with two parameters that control the shape. The parameter α affects how quickly the function decays to zero beyond the maximum sensing radius R and is calculated through defining the value of the function at the maximum sensing radius. The parameter β controls the width of the shoulder. As β increases, the weighting function approaches the true sensing region. Figure 1 illustrates the weighting function for a maximum sensing radius of $R = 2$ meters, for an α parameter defined

by $g_w(2) = 0.01$, and for values of $\beta = [5, 10, 15, 20, 25, 30]$. The weighting function is approximately 0 when the vehicle is outside the maximum sensing region and less than 1 when the vehicle is inside the sensing region. Therefore, this form of the weighting function is a conservative approximation of the true sensor weighting function which is a necessary condition to obtain a realistic solution. Since the function is continuous and smooth everywhere, it has a well defined derivative which is a major benefit for the numerical optimization solver.

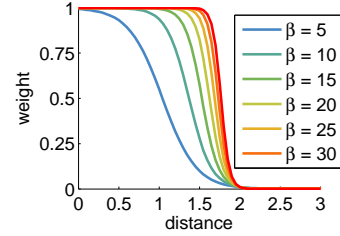


Fig. 1. The weighting function, $g_w(d)$, for various values of the parameter β with a maximum sensing radius of 2 meters and α is defined as $g_w(2) = 0.01$. As β increases, the weighting function approaches the true weighting function.

IV. ACCOUNTING FOR UNCERTAINTY

For certain systems with large process noise, the pre-planned trajectory is not a good estimate of the vehicle's path if the system is not provided with enough measurements to accurately estimate its state. Consequently, Eqn. (5) cannot be used in its current form. Rather, the uncertainty in the vehicle's position, p , at each time-step will need to be taken into account. For the system considered, the estimate of the position of the vehicle at each time-step has a Gaussian distribution. Unfortunately the Gaussian distribution has infinite support, but luckily most of the probability distribution is centered closely around the mean. Therefore, only a region in the neighborhood of the pre-planned positions will have to be considered for conservatively estimating whether the vehicle will receive a measurement from a sensor. This region is represented by the ellipsoid \mathcal{E}_ρ . Consequently, d in Eqn. (5) can be replaced with $\Theta(p, s, \mathcal{E}_\rho) = \max_{o \in \mathcal{E}_\rho} \|o - s\|$, which is defined as the maximum distance from the sensor position, s , to any point contained within the ellipsoid, \mathcal{E}_ρ .

For a multivariate Gaussian random variable there is a natural choice for the ellipsoid that represents the uncertainty in the vehicle's position. Consider the d -D position of the vehicle, $p \sim \mathcal{N}(\mu, \Sigma)$, where $p \in \mathbb{R}^d$. A confidence level, δ , can be specified that results in a corresponding confidence ellipsoid, \mathcal{E}_ρ , such that, $\text{p}(p \in \mathcal{E}_\rho) = \delta$. Given this definition, there are an infinite number of choices for the ellipsoid, but it will be restricted to be centered around the mean μ .

Define the random variable, $z \in \mathbb{R}$ such that $z = (p - \mu)^T \Sigma^{-1} (p - \mu)$, which is a measure of the distance of p from μ . It is well known that z has a Chi-square distribution, \mathcal{X}_d^2 . The confidence ellipsoid, parameterized by ρ , is then defined as, $\mathcal{E}_\rho = \{p \in \mathbb{R}^d | z \leq \rho\}$. Once the confidence level, δ , is specified, the confidence ellipsoid parameter is calculated via $\rho = F_{\mathcal{X}_d^2}^{-1}(\delta)$ where $F_{\mathcal{X}_d^2}$ is the chi-square cumulative

distribution function. In this formulation, the confidence level is a tuning parameter that controls the conservativeness of the solution.

A. Example Objective Function

Several important properties of the objective function can now be demonstrated through a simple example. Consider the double integrator system with 1D position defined by,

$$A = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}, \quad \Sigma_w = 1 \times 10^{-4}I, \\ C_i = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \Sigma_{v_i} = 0.5, \quad R_i = 0.5, \quad \forall i \in \{1, 2\}$$

where $\Delta t = 0.25$ seconds, $N = 200$, $\delta = 0.95$ and the pre-planned trajectory has equally spaced points between 0 and 10 meters. The logarithm of the objective function for this scenario is depicted in Figure 2, where s_1 and s_2 are the positions of the sensors. Note that the objective function is symmetric about the line $s_1 = s_2$ since the sensors are identical. The optimal sensor positions for the two sensors are $s_1 = 1.41$ and $s_2 = 4.54$ meters with an objective function value of 98.1.

Since the uncertainty of the vehicle position is included in determining when a measurement is allowed, the sensor positions are not very robust to positive perturbations to their positions. This is due to the fact that the process noise always increases the uncertainty of the vehicle. Consequently, if the sensor is pushed too far along the trajectory the vehicle's confidence ellipsoid will never be contained within the sensor footprint. This trend can be clearly seen in Figure 2 by the darker region located in the upper-right of the plot. If both sensors are perturbed by 0.05 meters into this region, the value of the objective function explodes to 2479 which is an increase of over 25 times. In contrast, if both sensors are perturbed by -0.1 the objective function value is 104.2 which is only a 1.06% increase. Therefore, one should be conservative in positioning the sensors since a small perturbation might result in a large detriment to the final solution.

This examples also illustrates that sensors closer to the start of the trajectory have a larger impact on the objective function. For example, if sensor s_1 is perturbed by $+0.05$ meters, then the objective function increases to 2479. However, if only sensor s_2 is perturbed by $+0.05$ meters, then the objective function only increases to 470. The reason behind this trend is due to the fact that if the first sensor is not allowed to perform any measurements then the subsequent sensors will be significantly less likely to provide a measurement. This property is dramatic for this example due to the small number of sensors. Small perturbations will have less of an impact if there are a large number of sensors over a short trajectory, since the interdependence is smaller.

B. Solution Methodology

The presented algorithm assumes the sensors are homogeneous to eliminate the impact on the solution due to the ordering of the sensors. Solving the optimization problem in P2.1 is difficult for two reasons. First, as stated in the previous section,

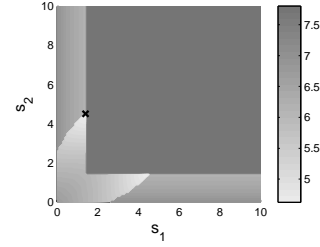


Fig. 2. The logarithm of the objective function for two identical sensors with the uncertainty of the position of the vehicle taken into account. The colormap ranges from light to dark for small to large values of the objective function. The optimal sensor positions are marked with an x. Note that since the sensors are identical the function is symmetric about the line $s_1 = s_2$.

the sensors' positions are highly dependent on each other and during the optimization procedure their position updates may have to be coordinated. Determining this level of coordination or interdependence strictly from the gradient information is difficult, if not infeasible, thus an intelligent program must be designed. Second, the optimization program P2.1 is not convex; consequently, finding the globally optimal solution is difficult and the final solution is sensitive to the initial conditions since the optimizer will tend to find local minimas.

Due to the interdependence of the sensors, an incremental sensor placement algorithm is proposed; each additional sensor is placed in the environment while the previously placed sensors' positions are fixed. The motivation behind this type of algorithm lies within the observation in Section IV-A that sensors further along the trajectory are heavily dependent on the previous sensors' positions. For example, consider the objective function shown in Figure 2. If sensor s_1 is initially placed at 1 meter then the furthest that sensor s_2 can be placed is 4.1 meters. However, if s_1 is slightly perturbed to 1.1 meters then s_2 can be placed at 4.2 meters.

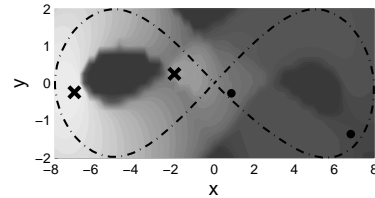


Fig. 3. This figure illustrates the objective function for the incremental addition of another sensor after two sensors have already been placed in the environment. The colormap ranges from light to dark for small to large values of the objective function. The pre-planned trajectory is shown as the dash-dotted line and the previous two sensor locations are indicated by the dots. There are two local minima for the additional sensor shown by the black x's.

While the incremental algorithm significantly reduces the complexity of the optimization algorithm, the objective function is still plagued by many local minima. Consequently, a poor choice in the initial condition of the sensor position can lead to a poor solution. Unfortunately, there is no clear simple choice in choosing the initial condition for the optimization solver. Consider the objective function that is depicted in Figure 3 corresponding to the value of placing an additional

sensor after two sensors have already been deployed in the environment. There are two local minima, highlighted by the x 's, with objective value of 16.3 and 27.1 corresponding to a 66.7% difference. Since local minima are difficult for an optimizer to circumvent, the initial condition for each incremental sensor addition will be instrumental to the quality of the overall solution. In general, random sampling of the initial sensor position helps to prevent getting stuck in local minima but the state space to sample from could be quite large. Fortunately, the best sensor positions are typically located near the pre-planned trajectory as seen in Figure 3. Thus, the sampling space can be effectively limited to points along the pre-planned trajectory.

The overall algorithm is stated in Algorithm 2. The first step in the algorithm is to sample q positions from the pre-planned trajectory into S_{sample} . Then for each additional sensor, up to M sensors, the position from S_{sample} which minimizes the cost function $V(\{s, s_{\text{fixed}}\})$ is chosen. This position is then optimized through gradient descent while keeping all of the previous sensor positions fixed. This procedure is repeated until all of the sensors are placed. Since the incremental placement of sensors does not guarantee a locally optimal solution, a final optimization procedure is performed.

Algorithm 2 Solution Method

- 1: $S_{\text{sample}} = \text{sample } q \text{ positions from } x_{\text{plan}}$
 - 2: **for** $i = 1, \dots, M$ **do**
 - 3: $s_i = \arg \min_{s \in S_{\text{sample}}} V(\{s, s_{\text{fixed}}\})$
 - 4: Optimize s_i through gradient descent
 - 5: $s_{\text{fixed}} = \{s_{\text{fixed}}, s_i\}$
 - 6: **end for**
 - 7: Optimize s_{fixed} through gradient descent
-

V. SIMULATION RESULTS

Several simulation studies were performed to evaluate the performance of the presented sensor placement algorithm.

A. Scenarios

The following examples have double integrator dynamics for a 2D vehicle position with,

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} 0.5\Delta t^2 & 0 \\ 0 & 0.5\Delta t^2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix}. \quad (6)$$

where $\Delta t = 0.25$ seconds. For all of the examples, the chosen parameters for the weighting function are $g_w(R) = 0.01$ and $\beta = 50$. In all the figures, the sensors are shown as numbered circles and the pre-planned trajectory is shown as the solid blue line with a subset of the uncertainty ellipsoids.

1) *Position Sensing*: The scenarios depicted in this section use position sensors with dynamics,

$$C_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \Sigma_{v_i} = 0.1I$$

and a confidence level of $\delta = 0.85$. First, simple cases are considered with only a few sensors to aid understanding of the algorithm's choices. Figures 4 and 5 show the solution after the incremental sensor placement on the left and the final solution on the right.

Figure 4 shows an example with five sensors with a radius of sensing of 2 meters and process noise $\Sigma_w = 0.001I$. In this scenario, the final optimization results in moving sensor 1 further along the trajectory because sensor 2 was placed in the center. The objective value after the incremental placement step is 52.6 as compared to 36.9 after the final optimization.

For the scenario displayed in Figure 5, there are four sensors with a radius of sensing of 2 meters and process noise $\Sigma_w = 0.0001I$. After the final optimization, sensor 3 and 4 are pushed further back toward the start of the trajectory. Sensor 4 is positioned to sense more positions along the final bend of the pre-planned trajectory. In addition, sensor 1 is moved up to sense all of the positions in the first bend of the trajectory. The objective value after the incremental sensor placement step is 63.7 and after the final optimization it is 40.2.

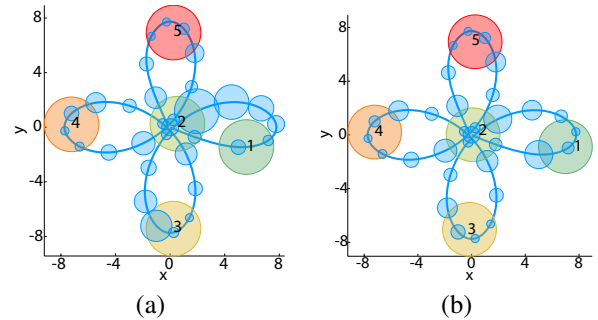


Fig. 4. The pre-planned trajectory is shown as the blue line, the uncertainty ellipsoids for a subset of the vehicle's pre-planned positions are shown as the blue ellipsoids and the sensor footprints are the numbered circles. (a) The solution after incrementally placing all of the sensors. (b) The solution after the last optimization step is performed. The main difference is that sensor 1 is allowed to move to the right since sensor 2 was placed in the center. The objective values for the two solutions are 52.6 and 36.9, respectively.

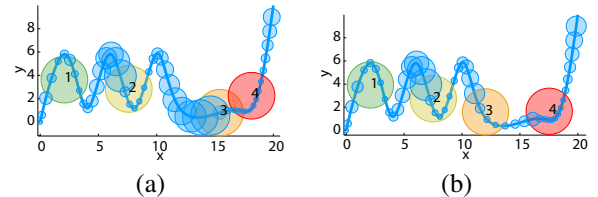


Fig. 5. (a) The solution after incrementally placing all of the sensors (b) The solution after the last optimization step is performed. The main difference between the two solutions is that sensor 3 and 4 are moved further back along the trajectory. The objective values for the two solutions are 63.7 and 40.2, respectively.

Figure 6 shows an example for a larger sized problem with 16 sensors and an extremely overlapping trajectory. The sensors have a radius of sensing of $R_i = 3$ meters. The process noise of the vehicle was set to $\Sigma_w = 0.01I$. In retrospect, the algorithm makes some fairly intelligent choices on where to place the sensors. In particular, sensors 2, 3, 6, 10, 11 and 15 all take measurements at multiple parts of the trajectory. Also,

sensors 5, 9 and 12 perform measurements for the entire time the vehicle is performing its loop.

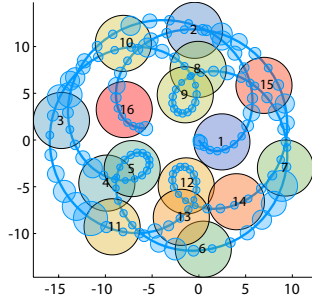


Fig. 6. The sensor placement solution for a sinuous trajectory for sixteen sensors with sensor range of 3 meters.

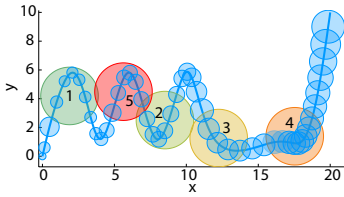


Fig. 7. The sensor placement solution for five velocity sensors.

2) *Velocity Sensors*: The last example shown in Figure 7 is for a system with velocity sensors with the same dynamics as the previous examples except,

$$\Sigma_w = 0.0001I, \quad C_i = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\Sigma_{v_i} = 0.001I, \quad R_i = 2, \quad \forall i \in \{1, \dots, 5\}$$

and a confidence level of $\delta = 0.95$. The sensor placements are similar to the previous example with position sensors but more sensors are required since the velocity sensors don't provide as much information.

B. Performance Analysis

To characterize the performance of the algorithm, the presented solution is compared with a nominal strategy which does not account for the position uncertainty. Three pre-planned trajectories were selected and the vehicle was simulated 500 times for each strategy. The first two use position sensors and the last one uses velocity sensors. Both sensor placement solutions for each trajectory are shown in Figure 8. The solution when accounting for the uncertainty of the vehicle is shown as the orange, solid circles with an x at the center and the solution without is displayed as the green, dashed circles with a dot at the center. The solution when not accounting for uncertainty was obtained through a gradient descent optimization algorithm because the interdependence of the sensors is drastically reduced in this formulation. In all cases, the vehicle starts at $(0, 0)$.

In all scenarios, the two different solutions are fairly similar at the beginning of the trajectory but differ drastically at the end of the trajectory. This difference arises primarily because

the solution when planning with uncertainty has to shift the sensors closer to the beginning of the trajectory to account for the increasing uncertainty.

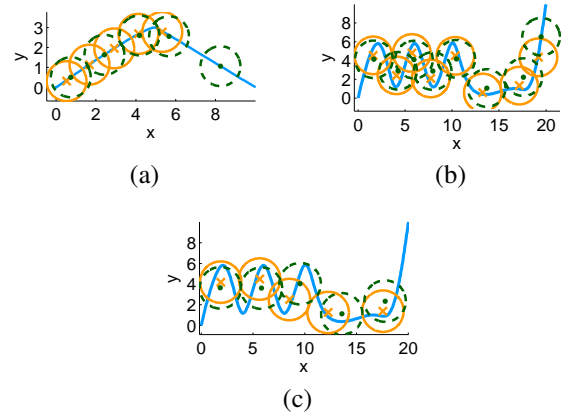


Fig. 8. Compares the sensor placement for three different scenarios with and without accounting for the uncertainty of the vehicle. The uncertainty included solution is depicted as the orange, solid circles with an x at the center and the solution not accounting for uncertainty is shown as the green, dashed circles with a dot at the center. For all scenarios, the major difference between the two solutions is the placement of the last sensors. (a) [Line] For a system with parameters: $\Sigma_w = 0.0005I$, $R_i = 1$ meter and $\Sigma_{v_i} = 0.05I$. (b) [Sinuous] The system parameters are: $\Sigma_w = 0.001I$, $R_i = 2$ meters and $\Sigma_{v_i} = 0.1I$. (c) [Vel. Sensors] The same parameters as previously used.

The performance results for the simulations are shown in Table I which presents several statistics for the positioning error at the final location. The statistics presented in the table are: the mean of the position error at the final time-step for all trajectories that didn't diverge, the maximum final positioning error and the percent of the trajectories that diverged. A divergent trajectory is classified as one in which the final position of the vehicle is outside of the uncertainty ellipsoid around the pre-planned trajectory. The uncertainty ellipsoid is calculated from the plan that accounts for the uncertainty of the vehicle since this is the most conservative prediction of the final position of the vehicle.

TABLE I
COMPARISON OF THE STATISTICS OF THE PERFORMANCE OF THE TWO DIFFERENT SOLUTIONS.

	With Uncertainty			Without Uncertainty		
	mean	max	% div.	mean	max	% div.
Line	0.95	2.73	0.0	0.27	25.10	6.8
Sinuous	0.70	2.19	0.2	0.37	22.92	3.4
Vel. Sensors	0.52	1.73	0.0	0.41	3.59	0.2

For all three scenarios, the mean performance of the vehicle is better when planning without uncertainty, but the max error and the percent diverged are significantly larger. The improved mean error performance when planning without uncertainty is expected since in this strategy, sensors can be placed further along the trajectory without penalty, thus reducing the final error for non-divergent trajectories. However, by not accounting for the uncertainty, the strategy is much more likely

to result in divergent trajectories. It is also interesting to note that for the velocity sensors scenario, the performance statistics are very similar for both planning strategies. This is due to the fact that velocity sensors accumulate a large amount of drift in the position estimate of the vehicle. Thus, there is less benefit in incorporating the uncertainty in the planning stage.

C. Comparison to Simulated Annealing

For comparison, the proposed algorithm was evaluated against Simulated Annealing (SA). Since SA is a stochastic algorithm it has the potential to avoid local minima and approach the globally optimal solution. The proposed algorithm and SA were evaluated on three different scenarios. Figure 9 shows a histogram of the relative difference, $\frac{V_{SA}-V}{V}$, for 50 trials of each scenario.

In the first scenario, the pre-planned trajectory is a simple 45° line and there are 3 sensors with a sensing radius of 6 meters. Figure 9(a) shows a histogram of the relative performance between the proposed algorithm and SA. Even for this simple scenario, SA has difficulty consistently finding a reasonable solution. In 28% of the trials SA had an objective value larger than 50 times the proposed solution. In the next scenario, SA was run for the example shown in Figure 4 and the histogram of the relative performance is shown in Figure 9(b). For the final scenario, both algorithms were evaluated on the sinuous trajectory displayed in Figure 5 and the results are shown in Figure 9(c). In this example, SA was able to find a 20% better solution than the proposed algorithm 30% of the time, but 70% of the solutions were over 6 times worse than the proposed algorithm. Consequently, even though SA sometimes can find a better solution, the proposed algorithm consistently finds reasonable solutions. In addition, SA ran slower than the proposed algorithm by a factor of 22, 16 and 13.

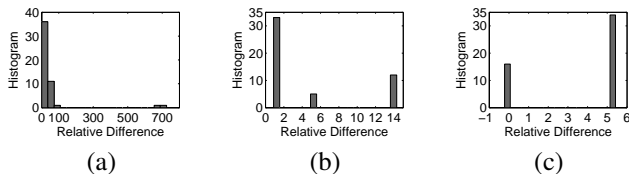


Fig. 9. Comparison of the relative performance between the proposed algorithm and Simulated Annealing for three different examples. A positive number means that Simulated Annealing found a worse solution than the proposed algorithm.

VI. EXPERIMENTAL RESULTS

The sensor placement strategies were also evaluated on a quadrotor unmanned aerial vehicle. The vehicle has an onboard inertial measurement unit which provides three-axis attitude, attitude rate and acceleration measurements. An external Vicon [10] positioning system is also used to provide measurements of the vehicle’s position with respect to a global coordinate frame.

With any aerial vehicle, the state (position, orientation and velocity) estimation is critical to a successful flight since small errors can easily lead to a disastrous crash. With this in mind, instead of withholding Vicon measurements when the vehicle

is outside of the sensing regions, the vehicle was provided with degraded measurements to prevent any major catastrophes.

In the following experiments, the sensor placement strategies were evaluated on the estimation of the quadrotor’s motion in the 2D x-y plane. For low speeds, the quadrotor’s planar translational dynamics can be approximated by standard linear point mass dynamics as shown in Eqn. (1), (6). To follow the pre-planned trajectory, a controller with both feed-forward and feedback terms was utilized. The parameters used for the experimental tests were: $\Delta t = 0.05$ seconds, $\Sigma_w = 0.00001I$, $\Sigma_{v_i} = 0.001I$ and $R_i = 0.25$ meters.

The results of the experimental trials are depicted in Figure 10, with the planning with and without uncertainty cases shown on the left and right, respectively. The pre-planned trajectory is shown by the blue, dash-dotted line; the trajectory starts at $(-1, 0)$ meters and ends at $(-0.25, 0)$ meters and requires the vehicle to stop at each corner. Ten experimental trials were conducted for each sensor placement strategy. When accounting for uncertainty, nine out of the ten tests were successfully flown by the quadrotor vehicle. The test that failed required human intervention because the vehicle initially missed the second sensor’s measurement region, causing the vehicle to oscillate around the trajectory. In contrast, the sensor placement strategy that didn’t account for the uncertainty of the vehicle required human intervention for all ten trials to prevent a crash. The main cause of the drastic performance differences between the two plans is the placement of the first sensor. In particular, the solution without accounting for uncertainty does not place a sensor at the starting position, causing the vehicle to consistently overshoot the pre-planned trajectory at the first corner and subsequently oscillate out of control.

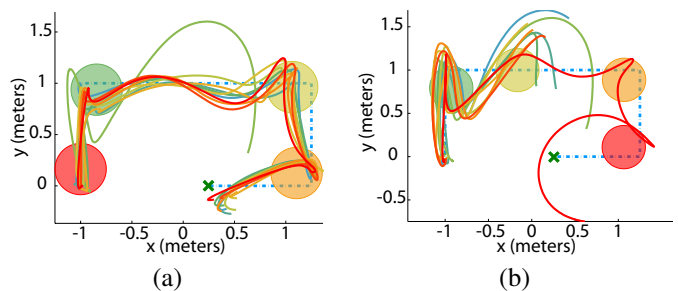


Fig. 10. Experimental data comparison for two different sensor placement strategies. The solid lines are the vehicle’s trajectory in the 2D plane, the dash-dotted line is the pre-planned trajectory, the colored circles are the sensor footprints and the x mark is the final goal location. The experimental trials for planning with and without uncertainty are shown in (a) and (b), respectively. (a) Successfully completes nine out of ten trials. (b) Pilot intervention is required for all ten trials to prevent a crash.

VII. CONCLUSIONS

A generalized solution to the sensor deployment problem for a linear system with linear measurements was presented. To ensure the conservativeness of the solution, the uncertainty of the vehicle’s position was accounted for when determining if the vehicle will receive a measurement. Finally, the sensor deployment strategy was demonstrated on a quadrotor platform,

where the proposed algorithm showed significant performance increases over other strategies.

There are several interesting areas of future work that the authors wish to explore. First, the case of non-radial sensing regions due to obstacle occlusions should be explored which will increase the applicability of the algorithm. Another extension to the algorithm is to incorporate nonlinear system dynamics as well as nonlinear sensor measurements. Lastly, the authors wish to extend the algorithm to handle heterogeneous sensors; this presents a challenging addition since the ordering of the sensors heavily impacts the final solution.

ACKNOWLEDGMENT

The authors wish to thank Deborah Meduna for numerous discussions during the preparation of the paper.

APPENDIX ANALYTICAL SOLUTION

For simple systems, an analytical solution to the optimization problem in P2.1 can be found. Consider a 1D system that has dynamics,

$$x_{k+1} = x_k + \Delta t u_k + w_k. \quad (7)$$

where $x_k \in \mathbb{R}$, $w_k \sim \mathcal{N}(0, \sigma_w)$ and $k \in \{1, \dots, N\}$. In the current scenario, the system can only take a single measurement, $y_s = x_s + v$ with $v \sim \mathcal{N}(0, \sigma_v)$, at time-step, s . The optimal measurement time can be solved for by differentiating the objective function and setting it equal to 0 which results in:

$$s^* = \left\{ -\frac{\sigma_0}{\sigma_w}, \frac{1}{\sigma_w} \left(-\frac{3}{4}(\sigma_0 + \sigma_v) + \frac{1}{4}(N+1)\sigma_w \pm \frac{1}{4} \sqrt{(\sigma_0 + 9\sigma_v + (N+1)\sigma_w)(\sigma_0 + \sigma_v + (N+1)\sigma_w)} \right) \right\}. \quad (8)$$

It is important to note that there is no guarantee that the value for s^* will be an integer. An integer solution, \bar{s}^* , can be determined by rounding s^* . The performance of this rounded solution is upper bounded away from the optimal solution by $V(\bar{s}^*) - V(s^*)$. Also, the optimal solution is not guaranteed to return a solution in the range $\{1, \dots, N\}$, but these cases can still provide valuable insight into the optimal sensing time.

PERTURBATION ANALYSIS

An interesting analysis to perform on the optimal sensing time is how the solution varies due to perturbations in the system parameters.

Time Horizon N :

One parameter of interest is the time horizon. The derivative of the optimal sensing time with respect to the time horizon is,

$$\frac{\partial s^*}{\partial N} = \frac{1 + 5\sigma_v + \sigma_0 + (N+1)\sigma_w}{4\sqrt{(9\sigma_v + \sigma_0 + (N+1)\sigma_w)(\sigma_v + \sigma_0 + (N+1)\sigma_w)}}. \quad (9)$$

The steady state change as N approaches ∞ is $\lim_{N \rightarrow \infty} \frac{\partial s^*}{\partial N} = \frac{1}{2}$. Consequently, as N increases the

optimal sensing time's dependence on the parameters of the system decreases; therefore, if the time horizon is increased by one time-step, the optimal sensing time will only change by 0.5.

Process Noise σ_w :

Characteristics of the optimal sensor measurement time with respect to changes in the process noise can also be determined analytically. Some of the important results from this analysis are as follows. As the process noise, σ_w , approaches 0 is $\lim_{\sigma_w \rightarrow 0^+} s^* = -\infty$, which is ill-formed because the sensing time is restricted to $s \in \{1, \dots, N\}$, but a trend can still be extracted from the result. As the process noise decreases, the optimal sensing time also decreases toward the first possible measurement time. As σ_w approaches ∞ is $\lim_{\sigma_w \rightarrow \infty} s^* = \frac{1}{2}(N+1)$. Consequently, as the process noise grows toward infinity the optimal sensing time isn't dependent on any of the system parameters and approaches the middle of the time horizon.

Measurement Noise σ_v :

The properties of the optimal sensing time can also be analyzed with respect to the measurement noise. As the measurement noise, σ_v , tends to 0,

$$\lim_{\sigma_v \rightarrow 0^+} s^* = \frac{(N+1)\sigma_w - 2\sigma_0}{2\sigma_w} \leq \frac{1}{2}(N+1) \quad (10)$$

which is bounded to the first half of the time horizon. As σ_v approaches ∞ ,

$$\lim_{\sigma_v \rightarrow \infty} s^* = \frac{2(N+1)\sigma_w - \sigma_0}{3\sigma_w} \leq \frac{2}{3}(N+1) \quad (11)$$

which is dependent on the other system parameters but can be bounded to the first two-thirds of the time horizon.

REFERENCES

- [1] P. Misra and P. Enge, *Global Positioning System: Signals, Measurements, and Performance*. Lincoln, MA, USA: Ganga-Jamuna Press, second ed., 2006.
- [2] M. Phatak, "Recursive method for optimum GPS satellite selection," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 37, pp. 751-754, April 2001.
- [3] M. Kihara and T. Okada, "A satellite selection method and accuracy for the global positioning system," *Navigation*, vol. 31, no. 1, 1984.
- [4] D. B. Jourdan and N. Roy, "Optimal sensor placement for agent localization," in *Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS 2006)*, (San Diego, CA), April 2006.
- [5] H. Zhang, "Two-dimensional optimal sensor placement," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, May 1995.
- [6] S. Ponda and E. Frazzoli, "Trajectory optimization for target localization using small unmanned aerial vehicles," in *AIAA Conf. on Guidance, Navigation, and Control*, (Chicago, IL), 2009.
- [7] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Optimal cooperative placement of GMTI UAVs for ground target tracking," in *Proceedings of the IEEE Aerospace Conference*, (Big Sky, MT), March 2004.
- [8] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, pp. 661-668, April 2006.
- [9] H. Strasdat, C. Stachniss, and W. Burgard, "Which landmark is useful? Learning selection policies for navigation in unknown environments," in *Proceedings of the IEEE International Conference on Robotics Automation (ICRA)*, 2009.
- [10] Vicon, *Vicon MX Systems*. visited January 2010. <http://www.vicon.com/products/viconmx.html>.