IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2019, accepted December 16, 2019, date of publication December 31, 2019,
date of current version February 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2963286

Single Vehicle Localization and Routing in GPS-Denied Environments Using Range-Only Measurements

SOHUM MISRA[1], BINGYU WANG[2], KAARTHIK SUNDAR[3], RAJNIKANT SHARMA[1], (Member, IEEE), AND SIVAKUMAR RATHINAM[2]

[1]Department of Aerospace Engineering, University of Cincinnati, Cincinnati, OH 45221, USA
[2]Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843, USA
[3]Information Systems and Modeling, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Corresponding author: Sohum Misra (misra.sohum@gmail.com)

This work was supported by the National Science Foundation (NSF) IIS Robust Intelligence Awards under Grant 1736087 and Grant 1527748.

ABSTRACT Path planning algorithms for unmanned aerial or ground vehicles rely on Global Positioning System (GPS) information for localization in many surveillance and reconnaissance applications. However, disruption of GPS signals, by intention or otherwise, can render these algorithms ineffective. This paper provides a way of addressing this issue by leveraging range information from additionally placed stationary objects in the environment called Landmarks (LMs). The placement of LMs and the route followed by the vehicle is posed as an integer program such that the total travel and LM placement cost is minimized. The proposed formulation of the optimization problem also allows for a limited field-of-view of the sensor on-board the vehicle. For instances that are hard to solve for optimal solutions using the integer program, we present two fast heuristics to find good feasible solutions. We provide a systematic framework and algorithms for the problem, and evaluate the system using numerical, simulation and experimental results.

INDEX TERMS GPS-denied environments, integer linear programming, localization, vehicle-routing.

I. INTRODUCTION

The use of Unmanned Vehicles (UVs) is growing at a rapid rate as a result of the versatility of UVs as a platform. UVs have found their way into a plethora of disparate applications spanning anywhere within personal (photography) [6], community (bridge inspection) [5], business (package delivery) [7] or military (surveillance, intelligence) [8], [11] domain. Improvements in the field of computer science, material science, electrical, and aerospace engineering have enabled expansion of the potential of these UVs drastically with researchers trying to push its boundaries even further [1], [3]. Most of the aforementioned applications concerning autonomous navigation of the UVs through a specified path or sequence of waypoints rely on GPS signal for localization. Autonomous navigation demands seamless knowledge of position and orientation of the UV to ensure proper decision-making. GPS is one such medium that can deliver accurate information of the UV's position and orientation. However, disruption of GPS signals either intentionally (e.g. signal jammer) or unintentionally (e.g. urban high-rise blockage) could potentially render these techniques not applicable [2], [12]. Furthermore, most indoor environments and many parts of terrain in an urban canyon of an outdoor environment do not have access to GPS; even if available, the access is intermittent and not reliable. Hence, localization in a GPS-denied or GPS-restricted environment is an absolute requirement and thus, is an active area of research.

GPS-denied navigation relies on local resources with known coordinates called Landmarks (LMs) for navigation. Whilst the knowledge of position and orientation is one important factor for autonomous navigation, its efficiency is certainly bolstered by path planning and resource planning techniques to utilize perishable and limited resources efficiently. For instance, a vehicle making repeated visits to a set of waypoints (WPs) will conduce poor utilization of time, fuel [29], [30], [33], [34] and other such resources [20], [22]. Thus, a natural problem of interest is to (i) plan a route via pre-specified WPs that will ensure efficient resource

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

utilization and (ii) minimize selection of LMs to enable position and orientation information retainment for successful navigation. Localization in GPS-denied environments require sensing, and correspondingly, localization procedures are dependent on the available of sensory measurements such as range and/or bearing measurements from LMs. Since sensory measurements are usually contaminated with noise, the problem of localization also requires filtering the noise in order to determine an accurate estimate of location and orientation.

Our initial formulation of the path planning problem was developed in [28]. However, our previous work lacked rigorous numerical, simulation and hardware results to support the theory. In this paper, we provide extensive simulation and hardware cases to compare different scenarios that clearly stresses the importance of optimal routing and navigation in uncertain environments. The hardware setup adheres to all the theoretical constraints and enables successful navigation of our unmanned vehicular platform (Turtlebot 2.0) using estimated states calculated with the help of optimally placed LMs. Further, the theory developed previously considered the sensor mounted on the UV to have a 360° Field-of-View (FOV). While some sensors such as RFIDs, LiDARs can potentially have circular FOV, many sensors such as non-scanning RADARs, most low-cost cameras do not have a circular FOV. This limited the robustness of our algorithm. So, keeping this practical consideration in our mind, we imposed a FOV constraint on our algorithm to encompass a wide range of sensors (especially vision based sensors such as cameras). In this paper, we generalize our formulation to have a FOV constraint and develop a multi-camera system (having limited FOV) to detect distances from multiple LMs simultaneously. The LMs are fixed points in the environment whose locations are known a priori. They are created using AR-Tag markers [18]. This setup helps us solve the data-association problem and enable simultaneous detection at the same time. The details regarding the markers, setup and hardware experiments are explained in later sections.

II. LITERATURE REVIEW

Several authors tried to tackle the problem of routing UVs through GPS-denied environment previously. D. Wong et. al. [37] used computer vision techniques to address the issue of vehicle localization in GPS-denied environments. Variants of Simultaneous Localization and Mapping (SLAM) techniques have proven to be quite effective for indoor navigation [15], [36]. Infrastructure aided localization for aerial and ground vehicles have been used in [17] where devices are pre-installed in the infrastructures that can provide range measurements to the vehicles. This was successfully carried out in California PATH's Automated Highway Systems (AHS) program. C. H. Ou [19] proposed a vehicular Ad-Hoc Network based localization approach where vehicles estimate their localization using periodic signals sent by a pair of Road Side Units (RSUs) which are very similar to what we are referring to as Landmarks (LMs) in our literature. A. Khattab et. al. [13] modified the previous approach using

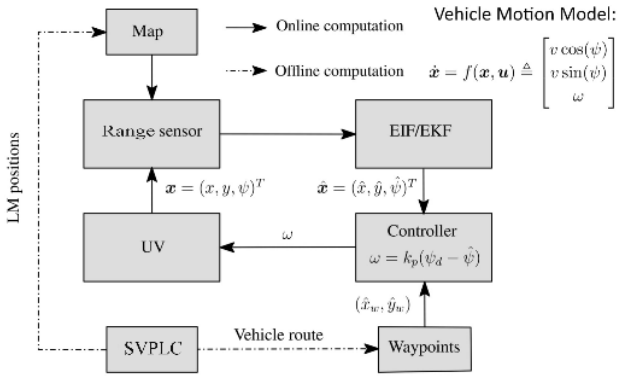

FIGURE 1. System architecture.

a 2-way time of arrival information to reduce the requirement of 2 RSUs to 1 per site for communication and therefore, localization. However, all techniques mentioned above concentrated on Landmark placement for localization without considering optimization of placement and / or selection of such ''Road Side'' information. All of these papers have an underlying assumption that information from LMs is available in abundance.

Given a path of a vehicle, the problem of optimally placing a minimum number of LMs was addressed in [21]. Given the placement of all the LMs, the problem of routing a UV through all the target locations reduces to the well known Traveling Salesman Problem (TSP), which is NP-Hard. This was further built upon and the problem of simultaneous LM placement and routing for UVs was formulated in [28].

This paper focuses on providing a systematic framework to include practical constraints (such as FOV) and validates it with hardware implementation. We use range-only measurements for estimation. We also provide comparative numerical, simulation and experimental results to show the performance and efficiency of our algorithms for several scenarios.

III. SYSTEM ARCHITECTURE

The overall workflow is shown in Fig. 1. The localization requirements for any path assigned to the vehicle must satisfy the following criteria: There must be at least 2 LMs in the sensing range of the vehicle from any point along the assigned path, or alternatively, there must be at least two edges incident on the vehicle's position at any time instant of the Relative Position Measurement Graph (Ref to Fig. 2). This requirement is according to Hermann-Krener criteria detailed in [10]. Song and Grizzle [27] stated that the bounds or uncertainty is related to the eigenvalues of the observability gramian which means,

- the vehicle should have path to at least 2 LMs (Fig 2) for the errors to stay bounded if a single sensor (range-only or bearing-only) is used, *i.e.*, the vehicle is able to sense its relative range or relative bearing angle from at least 2 LMs to relatively localize itself with respect to these Landmarks,

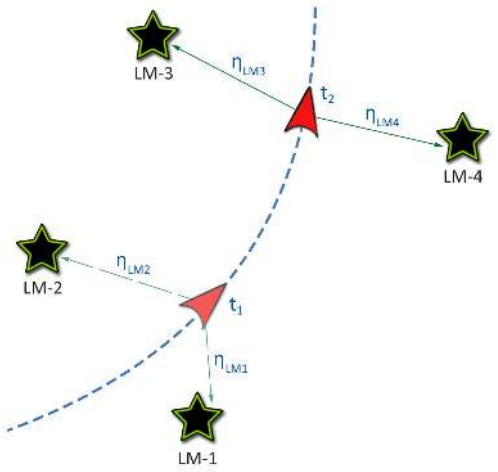

FIGURE 2. Relative position measurement graph.

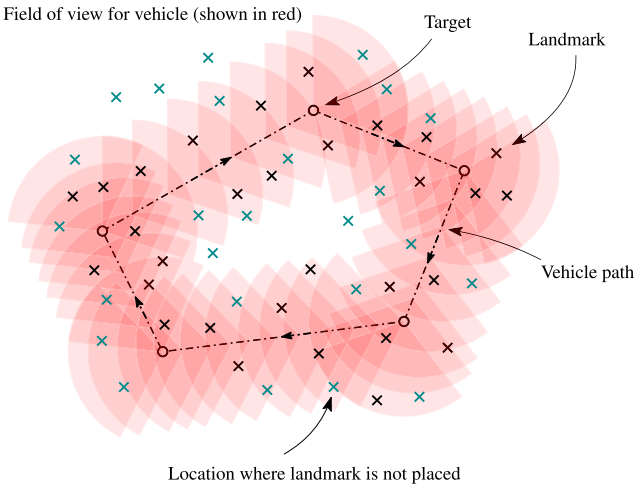

FIGURE 3. Problem scenario.

- the estimation algorithm will provide meaningful localization estimates if and only if the system is observable,
- an edge of RPMG is established only when the vehicle comes within the sensing range of the LM,
- path to more than 2 LMs quickens the convergence rate of the estimation algorithm.

The idea of using stationary landmarks to aid localization in a GPS-denied setting has been addressed previously in [25], [28], [32]. There are also a number of techniques that use so-called proxy landmarks for localization. Proxy landmarks can refer to either the use of additional UVs specifically for the purpose of localization or radio signals received from suitably positioned neighboring vehicles [14], [23], [24].

In scenarios where sufficient LMs are not available, additional LMs may be suitably placed to aid navigation. In this article, we allow for a discrete set of locations where one can place additional LMs. In addition to finding the LM placement, we also find the optimal tour for the vehicle so that each of the given targets is visited at least once. In the next section, we state the coupled landmark placement and path planning problem for the vehicle.

IV. PROBLEM STATEMENT

Given a vehicle starting from a depot whose coordinates are known, a set of targets or waypoints (WPs) to visit and a set of potential landmark (LM) placement locations, our goal is to find an optimal sequence for visiting WPs as well as a subset of LM locations such that:

1) The route starts and ends at the depot and visits each WP at least once;
2) There are at least 2 LMs within the sensing range and field of view of the vehicle from any point along the path;
3) The total travelling and LM placement cost is minimized (joint optimization).

This problem is referred to as Single Vehicle Path planning with Localization Constraints (SVPLC). The problem scenario is presented in Fig. 3. For mathematically formulating and modeling this problem, we use the following simplifying assumptions:

1) The vehicle travels at a constant velocity on a 2-dimensional plane;
2) The vehicle cannot move backwards;
3) The vehicle's on-board sensors have a constant sensing range with respect to every landmark;
4) The physical sizes of the vehicle and the LMs are small as compared to the size of the topographical area in which it is traversing;
5) The vehicle has a high turn rate.

In the ensuing section, we will discuss a joint optimization technique to reduce the combined cost of LM placement and travel, and present a formulation of the problem which takes the FOV constraint into account as well. We will also discuss heuristic approaches.

V. MATHEMATICAL MODELS, ALGORITHMS AND NUMERICAL RESULTS

In the absence of localization constraints, note that the SVPLC reduces to a standard TSP which is NP-Hard [21], [31], [32]. In the presence of localization constraints, it is possible that some TSP tours may not be feasible for the SVPLC. In this section, we will first mathematically formulate the SVPLC as an integer program. For hard instances or problems of larger size, we present heuristics that can provide high quality solutions for the problem. Next, we present numerical results to discuss the performance of the Integer Linear Programming (ILP) and the heuristics.

Overall, the problem is solved offline for the vehicle route and the optimal locations where landmarks need to be placed. This solution is then used with an online estimation algorithm to localize the UV in the GPS-denied environment as the UV traverses its route (Refer to Fig. 1).

A. AN INTEGER LINEAR PROGRAM

We define SVPLC on a set of targets V (including the depot location) and a set of potential LM locations K.

Let $G = (V, E)$ be a complete, directed graph where E is the set of directed edges joining any pair of vertices in V. For each edge $e = (i, j) \in E$, we define one binary variable x_{ij} which takes the value 1 if the edge e is traveled by the vehicle and 0 otherwise. The cost of edge $e = (i, j)$ is denoted by $c_{ij} = c_e$. We assume this travel cost is symmetric; that is for any edge $e = (i, j)$, $c_{ij} = c_{ji} = c_e$. For each potential LM location $k \in K$, we define one binary variable y_k which takes the value 1 if the location k is used to place a LM and 0 otherwise. The cost for placing a LM at k is denoted by d_k. Given a vertex subset, $S \subseteq V$, denote $\delta^+(S) := \{(i, j) : i \in S, j \in V \setminus S\}$ and $\delta^-(S) := \{(i, j) : i \in V \setminus S, j \in S\}$. Informally, $\delta^+(S)$ is the set of all the directed edges which start from a vertex in S and end at a vertex in $J \setminus S$, *i.e.*, $\delta^+(S)$ consists of edges leaving the set S. On the other hand, $\delta^-(S)$ is the set of all the directed edges which start from a vertex in $V \setminus S$ and end at a vertex in S, *i.e.*, $\delta^-(S)$ consists of edges entering the set S.

The formulation is as follows:

$$\min \sum_{e \in E} c_e x_e + \sum_{k \in K} d_k y_k$$

$$\text{subject to: } x(\delta^+(j)) = 1, \quad \forall j \in V, \tag{1}$$

$$x(\delta^-(j)) = 1, \quad \forall j \in V, \tag{2}$$

$$x(\delta^+(S)) \geqslant 1, \quad \forall S \subset V, \tag{3}$$

$$\sum_{k \in K_{e_s}} y_k \geqslant 2x_e, \quad \forall e \in E, e_s \in e, \tag{4}$$

$$x_e \in \{0, 1\}, \quad \forall e \in E, \tag{5}$$

$$y_k \in \{0, 1\}, \quad \forall k \in K. \tag{6}$$

Degree constraints in (1) and (2) ensure that all the targets are visited exactly once by the path. The subtour elimination constraints stated in inequalities (3) prevent solutions that include a subtour not connecting all the targets and the depot.

Inequalities (4) state the localization requirements of the path. These are also referred to as the edge-covering constraints in this article. To state the requirements, we first partition each edge into segments of fixed length. When the vehicle is traveling edge $e = (i, j)$ from i to j and on segment s, a LM location is considered to be in the sensing range of segment s or "covers" s if the location is in the field of view of the vehicle from both the endpoints of s (Fig. 4). For any segment s corresponding to edge e, let K_{e_s} denote the subset of potential LM locations that are in the field of view of segment s. Therefore, for the vehicle to localize itself as it travels from i to j, there must be at least two LMs placed in every subset K_{e_s} corresponding to the edge $e = (i, j)$. Refer to Fig. 5 for a feasible placement of LMs corresponding to an edge.

A key feature of the proposed model is that it is functionally versatile. It can handle non-uniformity among the potential LM locations, together with heterogeneity among the costs and covering capabilities associated with potential LMs. Moreover, additional, practical constraints can be readily added to the model. For clearance requirement between every two LMs placed, if locations i and j are too close to each other and cannot be selected together, we can apply $y_i + y_j \leq 1$; and for clearance requirement between every LM and every edge, if location i is too close to edge e and they cannot be selected together, we can enforce $x_e + y_i \leq 1$.

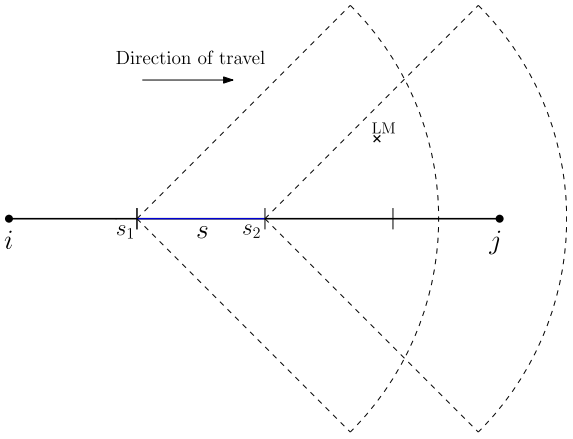

FIGURE 4. Edge (i, j) is split into four segments. For segment s, a LM is placed such that it is in the field of view from the end points of segment s.

B. HEURISTICS

Although the proposed ILP approach has several advantages, there are still scenarios where finding optimal solutions is time consuming, including (a) if the size of an instance is large, and a fast solution is desired; (b) if it is possible to place LMs almost anywhere in the field, instead of choosing from a discrete set of locations; and (c) if there is a constraint on the number of available LMs. To address these issues, in this subsection, we propose heuristics to find good feasible solutions relatively fast.

We propose a couple of heuristics that follow two key steps. In the first step, we ignore the localization constraints in the SVPLC and solve the single TSP spanning the targets and the depot. In the second step, we use the TSP tour obtained in the first step and place the LMs. Both the heuristics use the well known Lin-Kernighan-Helsgaun (LKH) algorithm [9], [16] to find a near-optimal TSP tour. In the first heuristic[1] (referred to as $Heuristic_{int}$), we use the integer program proposed in the previous section also for landmark placement; however, we fix the x_e values of the TSP tour obtained using the LKH algorithm and solve the simpler integer program. In the second heuristic (referred to as $Heuristic_{app}$), we use a greedy algorithm for LM placement proposed in [32]. It is globally suboptimal, but locally optimal with respect to every edge being traveled. This algorithm is of polynomial time complexity, thus, under certain circumstances, if the solution must be obtained within a very short period of time, it would be beneficial to utilize such an algorithm.

[1] $Heuristic_{int}$ involves solving the integer program with the x_e values fixed. Strictly speaking a heuristic is an algorithm that runs in polynomial time in the size of the input. But here, as we consider only 15 waypoints, once we fix the x_e values, we could solve each instance of the integer program in the order of a second. Therefore, with an abuse of notation, we refer to $Heuristic_{int}$ as a heuristic.

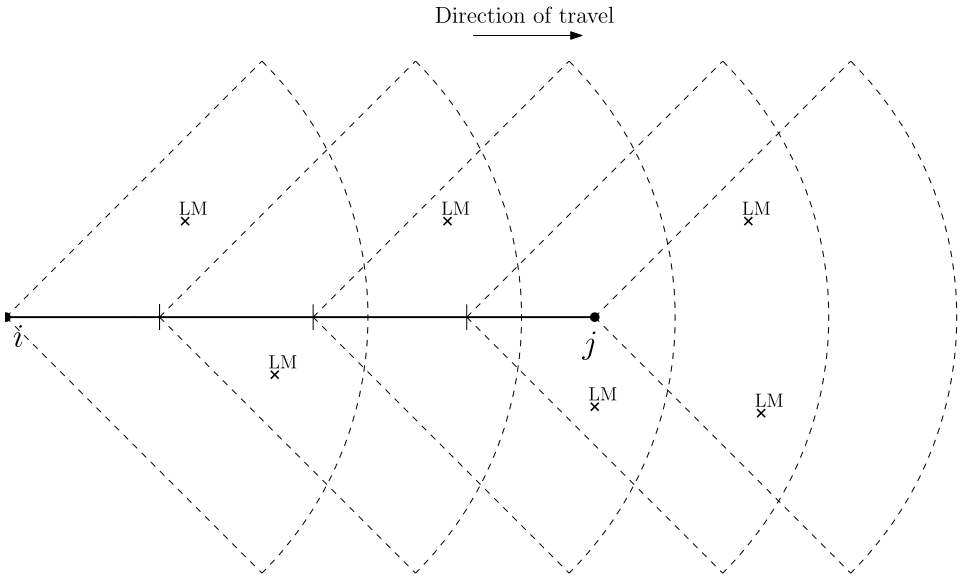

FIGURE 5. A feasible placement of LMs for edge (i, j).

C. COMPUTATIONAL PERFORMANCE AND ANALYSIS OF THE RESULTS

In order to solve the ILP, we implemented separation and branch and cut algorithms to find an optimal solution. Our algorithms were implemented in C++ (gcc version 4.6.3), using the elements of Standard Template Library (STL) in the CPLEX 12.7.1 framework. All the computations were performed on a Dell Precision T5500 workstation (Intel Xeon E5360 processor @2.53 GHz, 12 GB RAM).

We observed that when the cost for placing any LM is very small compared to the length of the path, the proposed ILP formulation can still be solved to exact optimum efficiently. Under this category, instances with 50 (or fewer) target locations and 250 (or fewer) potential locations for LM placement can be solved exactly within several minutes in general, some even within 5 seconds.

However, when the cost for placing a LM becomes larger, e.g. comparable to the length of the path, the time complexity of the computation increases rapidly. Under this category, some instances with 15 target locations and 150 potential locations for LM placement required around one hour to be solved exactly. No instance with 50 or more target locations and 250 or more potential locations for LM placement has been solved exactly within a time limit of two hours.

In addition, after analyzing the log data of the CPLEX solver, we observed that for many instances being exactly solved, it only takes 2-3 minutes to find a feasible solution relatively close to the exact optimum and around ten minutes to find out the optimum solution; However, it takes more than an hour to "prune all infeasible branches" and "close the gap between upper and lower bounds" (which is the way to guarantee optimality in integer programming).

We now present computational results for harder instances of the problem, *i.e.*, when the cost of placing a LM is relatively large. For the instances presented in tables 1,2, the vehicle travels through 15 waypoints (including its depot) scattered in a 100 × 100 grid. 250 locations are feasible for LM placement. The sensing range is 35 and the FOV is 90°. The cost for placing every LM is set to be 20 units. Table 1 shows the cost of the solutions obtained by the heuristics and the ILP. Table 2 shows the running time of the respective algorithms. Given these results, specially when the LM costs are high, $Heuristic_{int}$ provided the best trade-off between solution quality and running times.

TABLE 1. The cost of the solutions obtained by the heuristics and the ILP.

Instance	$Heuristic_{int}$ Cost			$Heuristic_{app}$ Cost			Optimum from ILP		
	Travel	LM	Total	Travel	LM	Total	Travel	LM	Total
1	337	480	817	337	600	937	348	460	808
2	314	520	834	314	580	894	314	520	834
3	289	500	789	289	580	869	298	440	738
4	300	480	780	300	560	860	300	480	780
5	272	460	732	272	540	812	303	420	723
6	306	520	826	306	580	886	316	500	816
7	299	440	739	299	580	879	299	440	739
8	328	540	868	328	540	868	353	460	813
9	335	540	875	335	560	895	335	540	875
10	304	580	884	304	600	904	318	520	838
11	287	440	727	287	560	847	299	420	719
12	319	520	839	319	540	859	323	500	823
13	301	460	761	301	540	841	319	440	759
14	283	500	783	283	540	823	284	480	764
15	331	540	871	331	620	951	378	460	838

VI. SIMULATION RESULTS

In this section, we implement the entire system architecture in simulations to test the performance of the proposed algorithms and understand the effects of the changes in the problem parameters on the trajectories of the vehicle.

A. SIMULATION SETUP

We utilize our MATLAB code suite *CNS Time-varying Topology System* [25], [26] to simulate the behaviors of the vehicle.

TABLE 2. The computation times required by the heuristics and the ILP in seconds.

Instance	Time (Heuristic$_{int}$)	Time (Heuristic$_{app}$)	Time (ILP)
1	1.20	0.0005	6396.6
2	1.13	0.0005	6134.3
3	1.26	0.0005	2423.8
4	1.08	0.0004	1259.7
5	0.30	0.0004	3161.1
6	0.13	0.0004	3645.9
7	0.70	0.0004	2901.2
8	0.41	0.0005	4623.4
9	0.15	0.0004	2014.9
10	0.11	0.0004	4132.0
11	1.01	0.0005	2903.6
12	0.47	0.0005	1201.8
13	0.21	0.0005	2278.3
14	0.17	0.0005	914.7
15	1.23	0.0005	2341.2

For estimation, it implements the Extended Information Filter (EIF), which is the information form of the Extended Kalman Filter (EKF), to estimate the states of the vehicle as it travels along its assigned path. All simulations are performed using the estimated data computed by the EIF, instead of exact location or orientation information. Meanwhile, for all figures in this section displaying vehicle trajectories, the blue dashed lines correspond to the trajectories directly computed by the EIF, while the red solid lines correspond to the ''actual'' trajectories which have considered errors and noise into account.

A fixed set of 150 potential locations where LM information can be introduced are placed relatively uniformly in a 100×100 m^2 testing area. The cost for setting up every LM is assumed to be identical. To reduce cross-disturbance, any two LMs being introduced must have a clearance of no less than 3m. The sensing range for the range or bearing sensors, ρ_s, vary from 15 to 35 meters.

Every 0.1 second, the vehicle performs one measurement with respect to known LMs in its adjacency. To make the simulations more realistic, we set the initial position uncertainty in both x and y directions as $5m$ for the vehicle, corresponding to the stage of the UV departure. The noise in velocity measurement is $0.1m/s$ and the noise in range measurement is $0.4m$.

B. RESULTS WITH 360° FOV

In this subsection, we present a series of results assuming the on-board sensor(s) have a 360° FOV. The vehicle repeatedly visits a fixed sequence of targets (obtained by solving the TSP). First, we show the necessity of a minimum number of landmarks to reliably traverse the given path (for this instance, the optimal number of LMs needed is 6). In Fig. 6, we place only one LM and show the output trajectories from the simulation. As expected, the vehicle deviates significantly from its desired path with each routing loop. The estimation and localization error become worse as the vehicle moves farther away from the LM. In Fig. 7, we have three LMs which is still fewer than the minimal number required (which is 6). However, the vehicle is able to localize itself

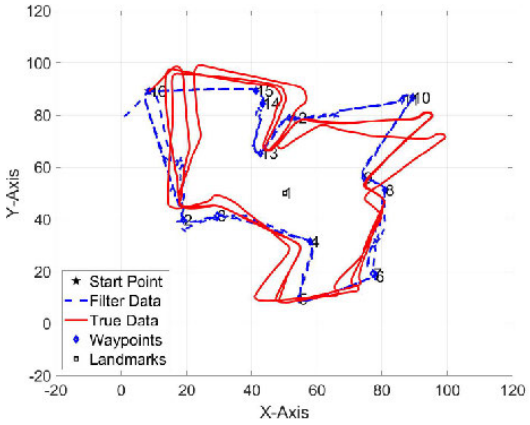

FIGURE 6. Vehicle path for sensing range = 35m and controller gain = 2.0. This simulation uses just 1 LM.

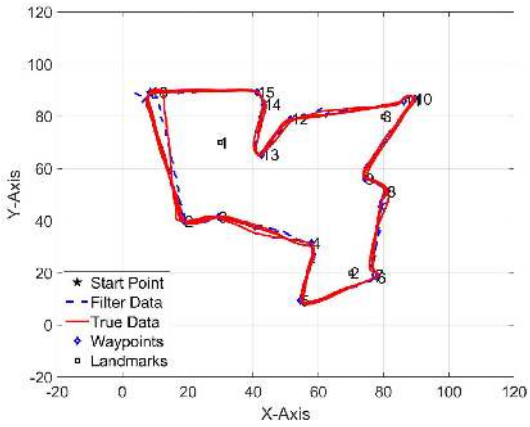

FIGURE 7. Vehicle path for sensing range = 35m and controller gain = 2.0. This simulation uses 3 LMs.

and route through the WPs with some errors in its position and heading. This is because it is able to perform valid measurements to two or more LMs periodically in its entire trajectory. This case becomes weakly observable as proven by A. Chakraborty et. al. in [4]. The two scenarios discussed above have reduced sets of LMs than required, which increase uncertainty in localization and yield unoptimized results, as the lengths of the actual trajectories also increase.

In the third case shown in Fig. 8, 25 LMs are uniformly placed all over the testing area such that wherever the vehicle is located in the area, it will always be able to sense two or more LMs. This kind of LM placement reduces our problem to a simple TSP which can be efficiently solved even if we start considering other restrictions like obstacles. The localization errors are significantly less in this case. However, utilizing a large number of LMs can significantly increase the LM placement cost, which too, is undesirable for engineering applications.

Now, we present our solutions by solving the ILP in Fig. 9 and Fig. 10. It can be observed from Fig. 9 and Fig. 10 that multiple solutions with the same optimal cost may exist, especially when the set of feasible LM placement locations

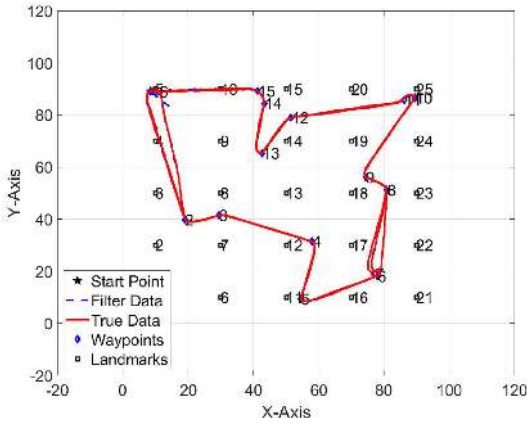

FIGURE 8. Vehicle path for sensing range = 35m and controller gain = 2.0. This simulation uses 25 LMs uniformly distributed in the area.

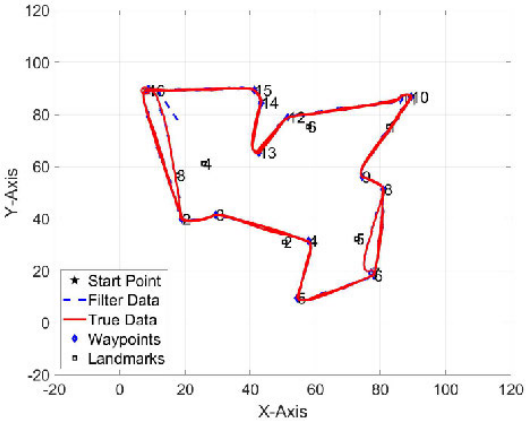

FIGURE 9. Vehicle path for sensing range = 35m and controller gain = 2.0 and 6 LMs (Optimal placement).

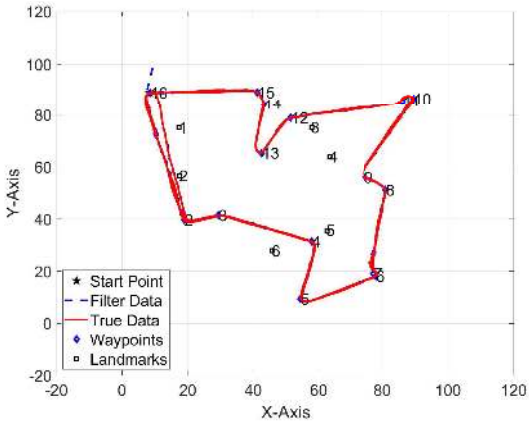

FIGURE 10. Another optimal LM placement. Vehicle path for sensing range = 35m and controller gain = 2.0. 6 LMs are optimally placed at locations different from the ones in Fig. 9.

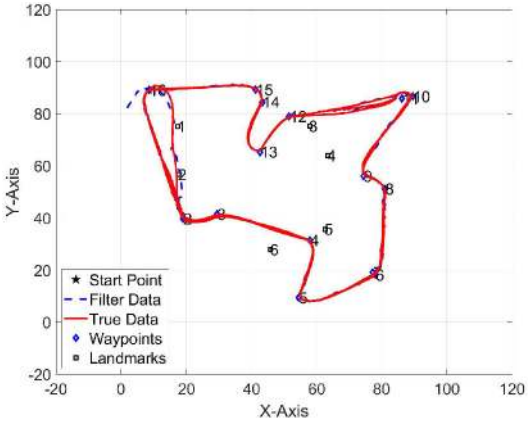

FIGURE 11. Optimal solution for sensing range = 35m and controller gain = 1.0.

is large. In addition, we find that the gain of the proportional controller used for routing determines the turn rate and trajectory smoothness of the vehicle. The controller gain by default is 2.0, and the corresponding minimum distance from path to waypoints (targets) during turning is 1.0m. When the gain decreases, the minimum distance can increase, hence the trajectory can become smoother. An illustration of this change can be seen in Fig. 11 with the vehicle following a smoother trajectory.

C. RESULTS WITH 90° FOV

We present results of simulations under a restricted FOV ($\pm 45°$) for the onboard sensor. This scenario corresponds to the experimental setup if we equip two cameras on-board.

Fig. 12 corresponds to the optimal solution when the cost for placing every LM is 1 unit. The length of the path is 327m (which is only 3m longer than the shortest TSP path possible) and 28 LMs are placed. It is evident that the vehicle is able to localize itself and achieve the routing mission safely. According to our joint optimization, when the cost for placing every LM is raised to 20 units, the new optimal solution would use a longer path of 350m, but only requires

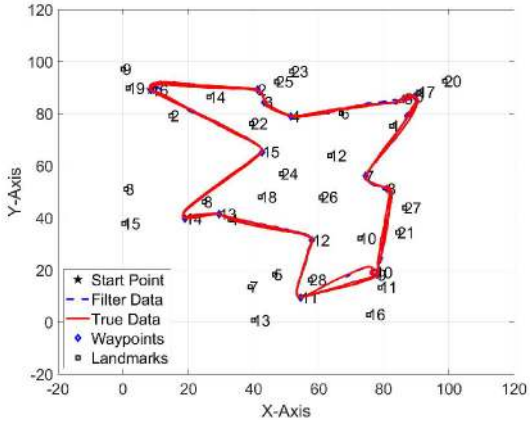

FIGURE 12. Vehicle path for sensing range = 35m and controller gain = 2.0. This solution uses 28 LMs.

25 LMs (displayed in Fig. 13). However, compared to the previous subsection where 6 LMs are sufficient, the number of LMs required here is larger due to the additional FOV constraints.

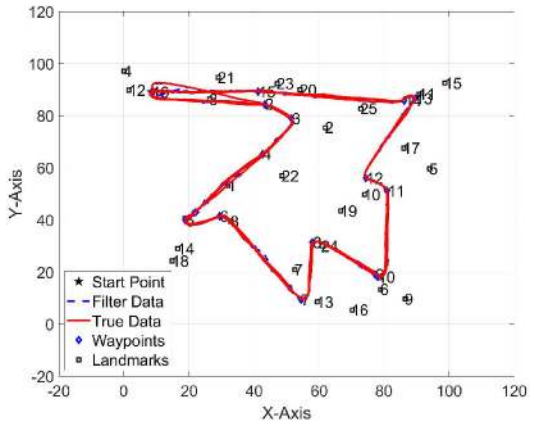

FIGURE 13. Vehicle path for sensing range = 35m and controller gain = 2.0. This solution uses 25 LMs.

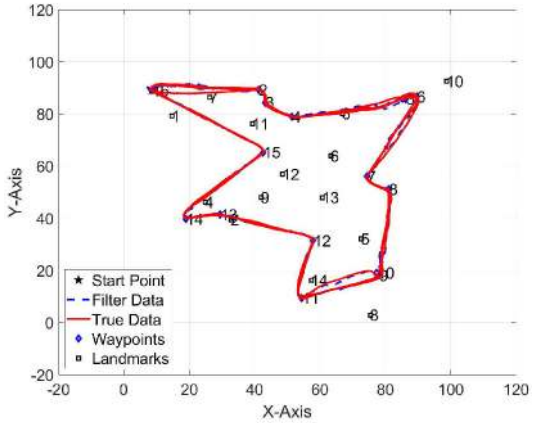

FIGURE 14. Vehicle path for sensing range = 35m and controller gain = 2.0. This solution uses 14 LMs.

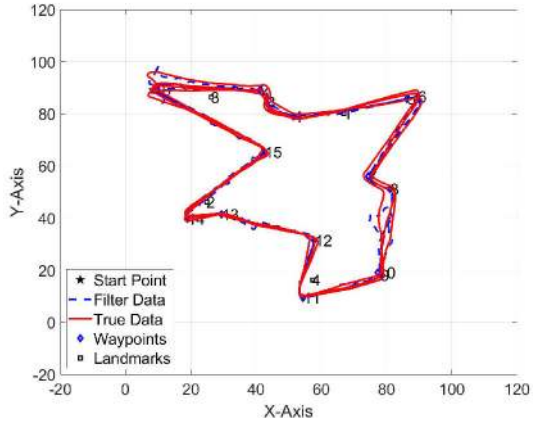

FIGURE 15. Vehicle path for sensing range = 35m and controller gain = 2.0. This solution uses 4 LMs.

In the following discussion, we use the same path as in Fig. 12 and gradually reduce the number of LMs placed. 14, 4 and 2 LMs are placed to aid in localization in Figures 14, 15 and 16, respectively. The vehicle is still able to navigate through the target locations for most of the time, although the level of error clearly increases as the number of LMs placed is reduced. The reason is that whenever the

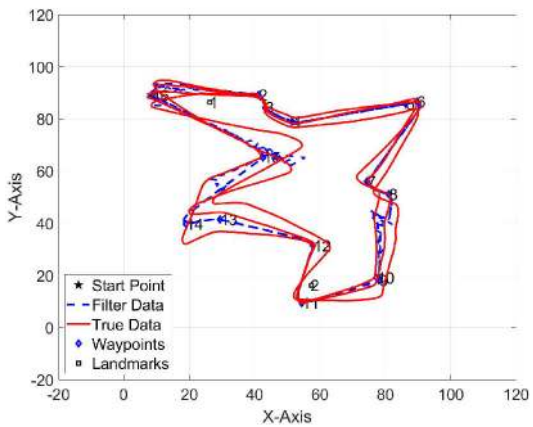

FIGURE 16. Vehicle path for sensing range = 35m and controller gain = 2.0. This solution uses 2 LMs.

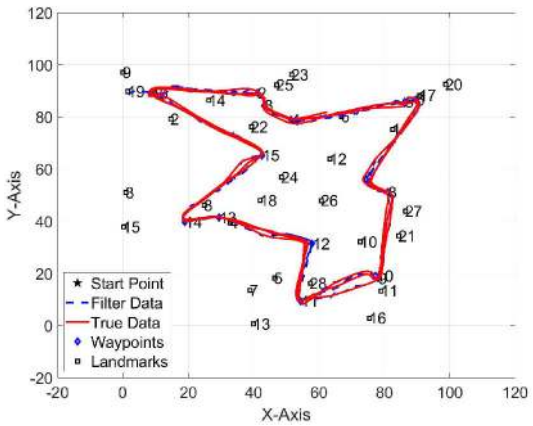

FIGURE 17. Vehicle path for sensing range = 15m and controller gain = 2.0. This solution uses 28 LMs.

vehicle is able to detect two LMs within its proximity, our state estimation algorithm will converge and offer a reasonably good update. Even when only one LM is detectable, the vehicle may still be able to use interoceptive sensors like IMU, encoders, etc. to assess its motion and attitude information such as velocity and turn rate, and additional LM information could help reducing the error in estimation. Thus, it is possible to utilize our heuristics to deliver a plan for routing and LM placement even when the resources for localization become more restricted under certain conditions.

To conclude this section, we would like to show how our architecture can also be used to test variations like changes in sensing range. For example, Figures 17, 18 and 19 utilize the same LM placement as Figures 12, 14 and 15, respectively, but the sensing range here is reduced to 15m from 35m. It is evident that the levels of error increase considerably. When the sensing range is reduced, the number of LMs required to ensure safe localization and routing will increase.

VII. EXPERIMENTAL RESULTS

A. HARDWARE SETUP

The hardware experiments were performed in an area of grid size $4 \times 8\ m^2$. Fig. 20 shows an instance of navigation

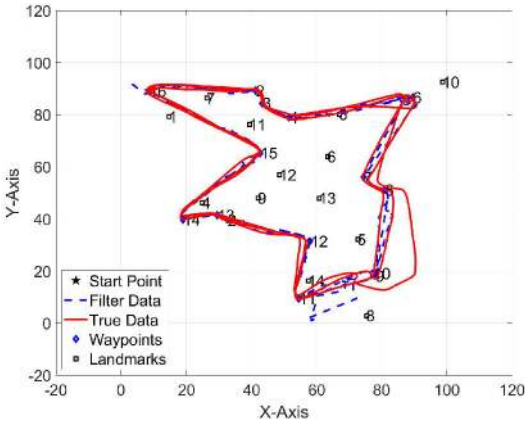

FIGURE 18. Vehicle path for sensing range = 15m and controller gain = 2.0. This solution uses 14 LMs.

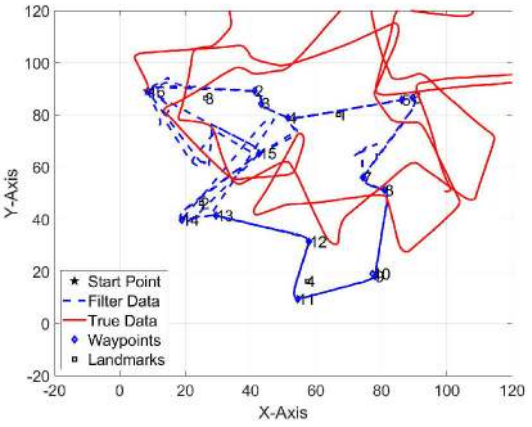

FIGURE 19. Vehicle path for sensing range = 15m and controller gain = 2.0. This solution uses 4 LMs.

FIGURE 20. Experimental setup.

through 4 WPs with the help of 8 LMs in the given space. We used Turtlebot 2.0 (Fig. 22) as our platform to conduct the experiments [35]. Turtlebot 2.0 is a mobile platform that is built on top of iRobot create and used Orbbec Astra camera(s) to sense the environment. It is powered by open source ROS

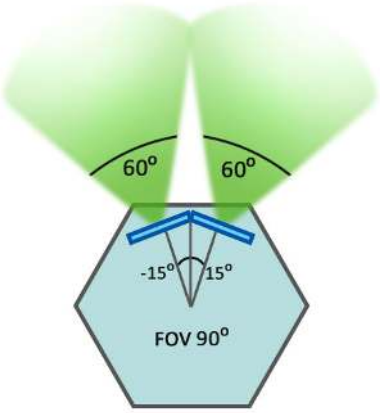

FIGURE 21. Camera setup 1.

platform and comes with many pre-installed ROS libraries to facilitate building customized applications. The Orbbec Astra is an RGBD Camera with 60° field-of-view (FOV) and an image sensing range of 0.6m to 8m. It weighs 0.3 kg with a dimension of 165 × 30 × 40 mm (with Turtlebot 2.0 mounting capability). It also provides an RGB stream of size 640 × 480 pixels at 30 frames per second, making it ideal for our use. We mount multiple such cameras on our Turtlebot 2.0 to increase the total effective FOV of the sensor system. This helps in sensing range or bearing angle or both from multiple LMs. We restrict our experiments to range-only measurements and consider AR-Tags as LMs. The details of camera and AR-Tag setup is explained in the following subsection (VII-B). The LMs (AR-Tags) are manually placed at their respective coordinates with the help of the MOCAP system which itself has a $3-5$ mm level of accuracy. Therefore, their locations and placements are prone to human error. This is taken care by increasing the measurement noise considered during estimation. The optimal LM placement algorithm used for hardware experiments considered vehicle size, however, it didn't consider LM occlusion. We assumed that using communication (RFID) based range sensors would circumvent the occlusion issue in practical outdoor scenarios, thus, obviating the need to use occlusion as an active optimization constraint. During our experiments, we restrict the vehicle to be moving at a maximum speed of 0.05 m/s to maximize point turn capability of the vehicle.

B. CAMERA SETUP, AR-TAG SETUP AND SENSOR FUSION

The update step for the EKF requires the exteroceptive sensors mounted on a vehicle to detect/communicate with external environment agents frequently. In this case, the external environment agents are AR-Tags [18] acting as Landmarks (LMs) and the exteroceptive sensor is a set of Orbbec Astra cameras mounted on the Turtlebot 2.0 [35] platform. As mentioned in the previous subsection, each Astra camera have a 60° FOV. So, we mounted 2 cameras on our platform to have a total of 90° FOV as shown in Fig. 21, and Fig. 22. The Orbbec Astra camera is a stereo camera, which

FIGURE 22. Turtlebot2 with camera setup.

FIGURE 23. AR Tags.

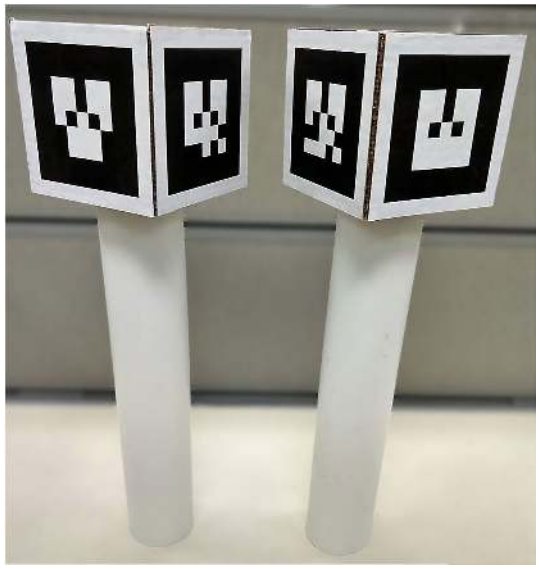

FIGURE 24. Landmark (LM) setup.

can measure depth alongside providing an RGB stream, thus helping in improving the accuracy of range measurements when required. The LMs are equipped with AR-Tags. These tags are generated and detected by using an existing ROS Ar Track Alvar package that can be integrated with the localization and navigation algorithm for range sensing purpose. Sample AR tags are shown in Fig. 23. Their detection works in a way similar to that of QR codes as well, with focus on range and orientation detection rather than information storage optimization as in the case of QR codes. We have used a combination of ROS Astra camera package, ROS AR Track Alvar package [18], ROS tf transform package to obtain the relative $'x'$ and $'y'$ position of the tags with respect to the centroid of the Turtlebot, thereafter, using this information to calculate the range of the LMs from the vehicle as $\rho = \sqrt{x^2 + y^2}$.

Each LM is represented by 4 tags, marking the 4 faces of a cuboid as shown in Fig. 24. The center of one face (say F1) is selected as the center of the LM. The range measurement obtained from the LM is always relative to F1, *i.e.,* the positions of the other 3 faces are all relative to the first face F1. This enables us to consider each LM as a point source thus making the range detection process easy and uniform. As shown in Fig. 25, multiple LMs were detected by the cameras simultaneously within a range of $1m$ to $3m$ with an accuracy of ± 3 cm at $20 - 25$ Hz.

C. NAVIGATION RESULTS

We performed several experiments with true and estimated states. The waypoints were generated in a grid of size 4×8 m^2. The sensing range of the camera was set to 3 m. The viewing angle and range were chosen to match the simulated sensor outputs with the actual sensor setup. A low controller gain ($K_p = 0.2$) was chosen to reduce the effect of non-linear system/process noise while the vehicle is in motion. However, as mentioned above, the vehicle's velocity was kept low too at a constant value of 0.05 m/s) to mimic point-turn capability. The set of grid points in a grid of size 0.5×0.5 was chosen as the set of potential landmark locations. A p value of $0.05m$ (collision avoidance distance) was used for the separation distance between the vehicle and any location where the landmarks were to be placed. Although this distance can be increased, a small value was chosen to make the instances feasible. The algorithm ($Heuristic_{int}$) presented in Sec. V is then used to compute the subset of locations where LMs are to be placed and the vehicle route. Using the LMs at the locations obtained from the algorithm, an Extended Kalman Filter (EKF) was used to estimate the position and heading of the vehicle using the range measurements. A detailed comparison among the various cases for the hardware experiments and their subsequent results are provided in the subsection below (VII-D).

D. HARDWARE RESULTS

The hardware experiments were conducted to test the performance of the combinatorial optimization problem when applied to real world navigation and routing scenarios. The testing and comparisons were categorized in the following subsections.

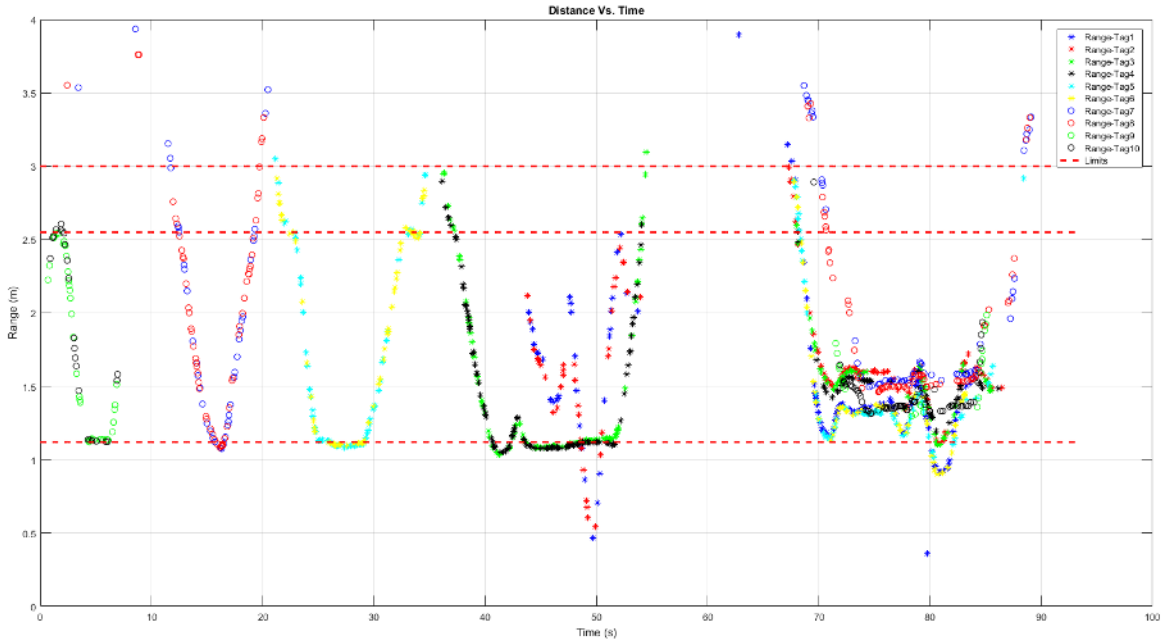

FIGURE 25. Simultaneous 10 LM detection.

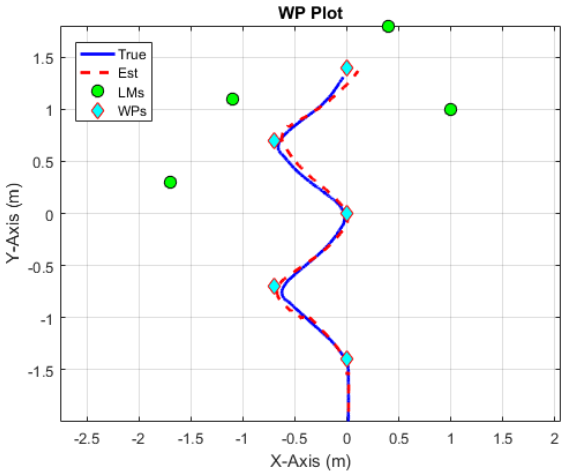

FIGURE 26. Vehicle path with true states being fed back to the controller.

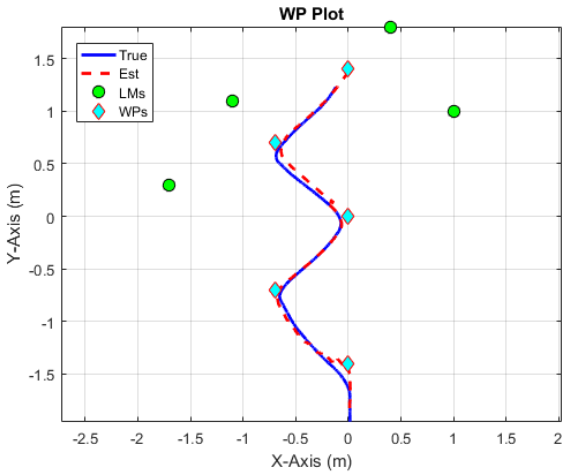

FIGURE 27. Vehicle path with estimated states being fed back to the controller.

1) EFFECT OF ESTIMATION WITH REAL SENSORS ON THE CONTROLLER

The vehicle path in this subsection is fixed and LMs are placed using the second step in $Heuristic_{int}$. It can be seen from Fig. 26, 27, 28 and 29 that the difference between navigation using true states versus estimated states is small. For all the hardware trajectory plots, the blue solid line denotes the actual path followed by the vehicle and the red dashed line denotes the estimation output of the filter. We performed this experiment considering a camera setup as shown in Fig. 21 with 90° FOV, $3m$ sensing range approximately and 4 LMs as shown in Fig. 24. We consider that a WP is reached if the vehicle is anywhere within 15 cm around the concerned WP. This helps in compensating for the inertial lag of the vehicle. It can be seen from the 3σ bound plots that the errors are within bounds, thus, the filter is consistent. This shows that the proposed algorithms found a set of LM positions corresponding to the WPs that enabled successful navigation of the vehicle with proper EKF tuning.

2) PERFORMANCE OF THE ALGORITHMS WITH SIMULATED SENSOR DATA AND MEASUREMENTS

In this subsection, both the vehicle path and the LM placement are obtained using $Heuristic_{int}$. The result obtained in the previous experiments are further verified by using a different set of WPs and their respective LM positions to perform a more complicated Zig-Zag trajectory as shown in Fig. 30 using simulated LMs. The error plot corresponding to this trajectory is shown in Fig. 31 which reflects

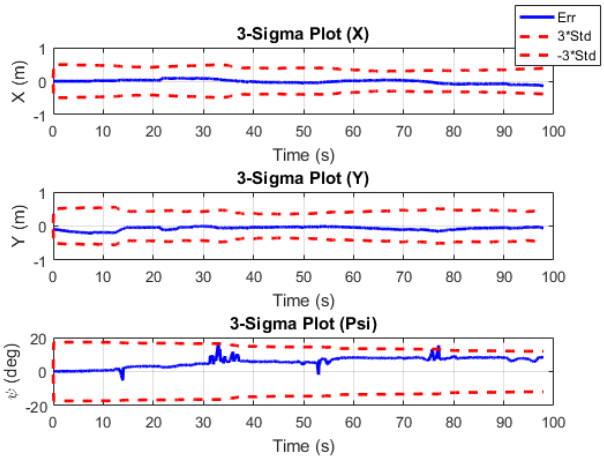

FIGURE 28. Error plot for Fig. 26 with $\pm 3\sigma$ bounds for routing with true states being fed back to the controller.

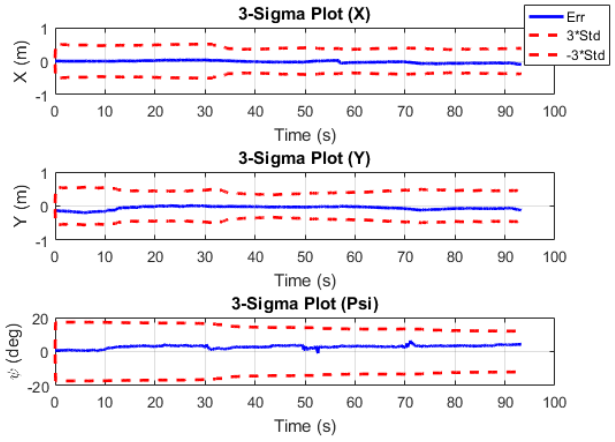

FIGURE 29. Error plot for Fig. 27 with $\pm 3\sigma$ bounds for routing with estimated states being fed back to the controller.

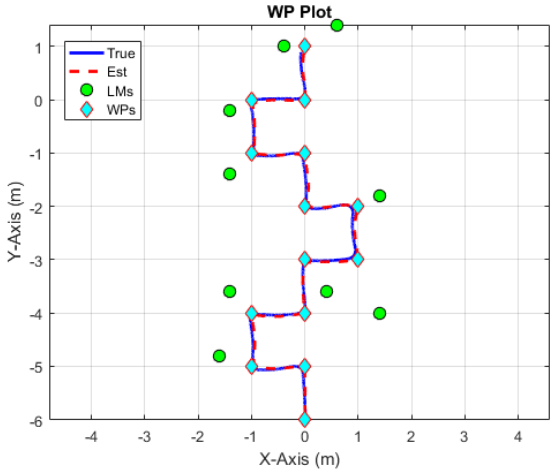

FIGURE 30. Vehicle path with estimated states being fed back to the controller.

that the vehicle navigates properly keeping the error withing 3σ bounds for more than 99.7% of the time (3σ percentage). On achieving successful navigation for open-loop cases

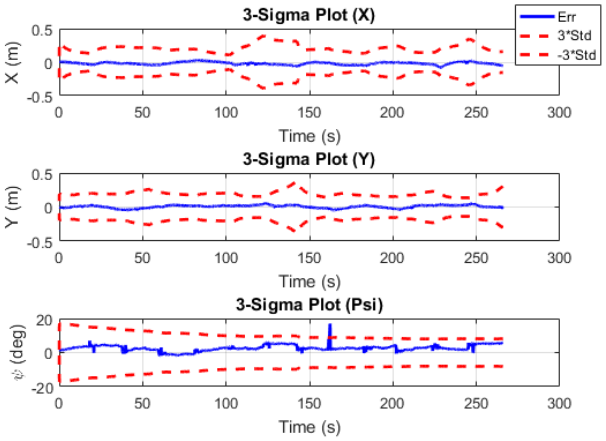

FIGURE 31. Error plot for Fig. 30 with $\pm 3\sigma$ bounds for routing with true states being fed back to the controller.

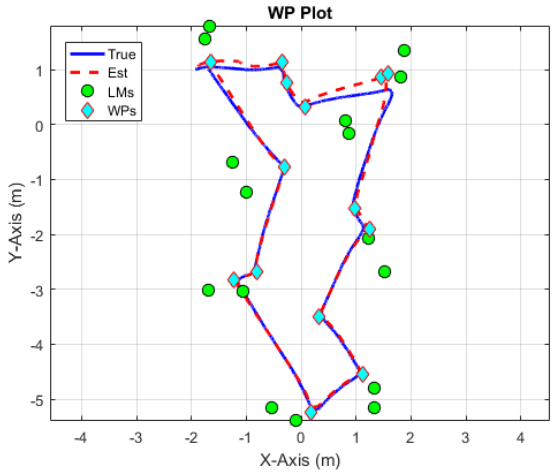

FIGURE 32. Vehicle path with estimated states being fed back to the controller.

(Fig. 26, 27 and 30), we went on to implement our algorithm in a complicated closed-loop scenario. It can be observed from Fig. 32 and 33 that our combined algorithm is also capable of finding out optimal solution for WPs and trajectories similar to the real-world and successfully navigate through them using the estimated states. The experiments were done considering simulated sensor measurements with 90° FOV, $3m$ sensing range.

3) PERFORMANCE OF THE ALGORITHMS WITH REAL SENSOR DATA AND MEASUREMENTS

In this subsection, both the vehicle path and the LM placement are obtained using $Heuristic_{int}$. It can be seen from Fig. 34 and 35 that the Turtlebot was able to successfully start from the depot, route through all the points and return to its depot. The hardware experiment was limited by the availability of real LMs, sensor detection rate (limited to approximately 10 Hz on average), processing speed and capability of the on-board computer and the space available for conducting the experiment. The routing and navigation was completely independent of MOCAP inputs.

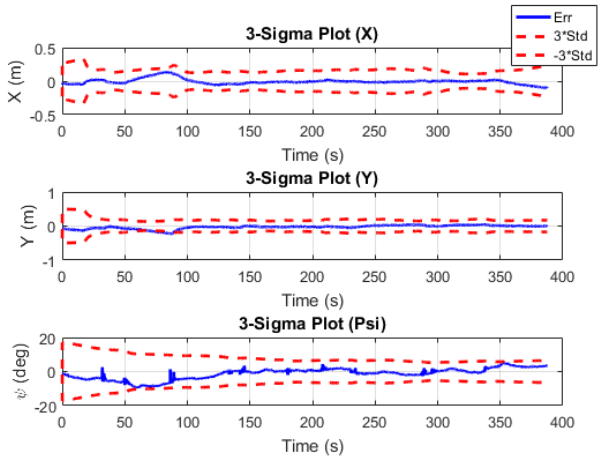

FIGURE 33. Error plot for Fig. 32 with $\pm 3\sigma$ bounds for routing with estimated states being fed back to the controller.

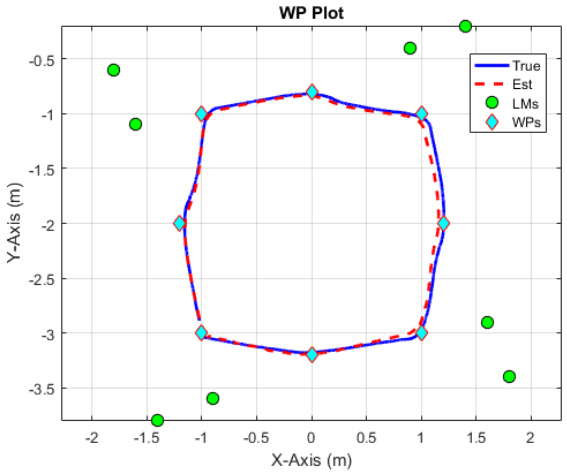

FIGURE 34. Vehicle path with estimated states and real sensors with 90° FOV (Closed Loop).

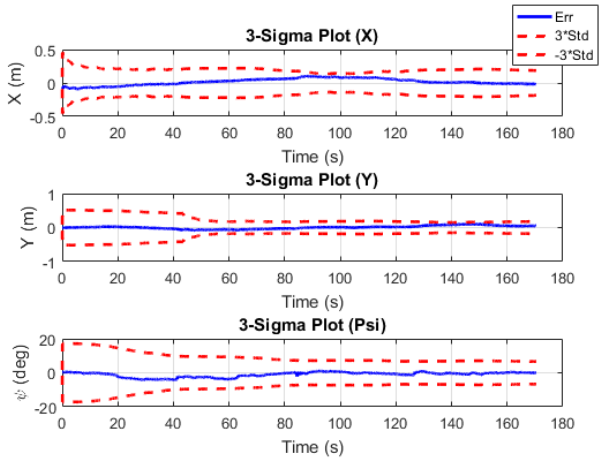

FIGURE 35. Error plot for Fig. 34 with $\pm 3\sigma$ bounds for routing with estimated states and real sensors with 90° FOV (Closed Loop).

The MOCAP system was used only to capture the actual path traversed by the bot during its tour. It can be seen that the estimates are mostly within $\pm 3\sigma$ bounds with intermittent spikes in the data. The spikes are caused due to higher degrees of non-linearity during the turns. The overall measurement noise is factored in by considering a higher error covariance for measurement noise that in turns smooths out the overall trajectory.

The video for the experiment is available using the link https://youtu.be/4F3muxX8zL0.

VIII. CONCLUSION

The numerical, simulation and hardware experiments show that the proposed optimization framework works effectively in practice using range-only measurements from landmarks. Comparing the results in sections VII-D.2 and VII-D.3, it can be seen that the quality of the routing can be affected by the quality and capability of the sensors used for measurements. The best range detection was achieved using simulated LMs as simulated measurements are not prone to detection failures due to ambient conditions. We have used cameras and AR-Tags to measure ranges from respective LMs. This setup provides a good range estimate, however, it is limited by the camera quality, resolution and ambient lighting conditions. Developing better ranging sensors was beyond the scope of our work and can be considered as a future direction of research. We believe that this work can have significant contribution towards advancing automated driving and path planning for aerial package delivery. As mentioned in Section VII-A, the LM placement is prone to measurement approximation and human errors. In the future, we plan to estimate the location of the LMs along with the vehicle states simultaneously. We also plan to impose the turning radius constraints of the vehicle and include them in the optimization.

REFERENCES

[1] P.-J. Bristeau, F. Callou, D. Vissière, and N. Petit, "The Navigation and Control technology inside the AR.Drone micro UAV," *IFAC Proc. Volumes*, vol. 44, no. 1, pp. 1477–1484, 2011.
[2] J. V. Carroll, "Vulnerability assessment of the U.S. transportation infrastructure that relies on the global positioning system," *J. Navigat.*, vol. 56, no. 2, pp. 185–193, 2003.
[3] D. Casbeer, S.-M. Li, R. Beard, R. Mehra, and T. Mclain, "Forest fire monitoring with multiple small UAVs," in *Proc. Amer. Control Conf.*, Aug. 2005, pp. 3530–3535 .
[4] A. Chakraborty, S. Misra, R. Sharma, and C. N. Taylor, "Observability conditions for switching sensing topology for cooperative localization," *Unmanned Syst.*, vol. 5, no. 3, pp. 141–157, 2017.
[5] B. Chan, H. Guan, J. Jo, and M. Blumenstein, "Towards UAV-based bridge inspection systems: A review and an application perspective," *Struct. Monitor. Maintenance*, vol. 2, no. 3, pp. 283–300, Sep. 2015.
[6] E. Cheng, *Aerial Photography Videography Using Drones*. Berkeley, CA, USA: Peachpit Press, 2015.
[7] R. D'Andrea, "Guest editorial can drones deliver?" *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 3, pp. 647–648, Jul. 2014.
[8] D. Gregory, "From a view to a kill: Drones and late modern war," *Theory, Culture Soc.*, vol. 28, nos. 7–8, pp. 188–215, 2011.
[9] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," *Eur. J. Oper. Res.*, vol. 126, pp. 106–130, Oct. 2000.
[10] R. Hermann and A. J. Krener, "Nonlinear controllability and observability," *IEEE Trans. Autom. Control*, vol. 22, no. 5, pp. 728–740, Oct. 1977.
[11] P. J. Hiltner, "The drones are coming: Use of unmanned aerial vehicles for police surveillance and its fourth amendment implications," *Wake Forest JL & Pol'y*, vol. 3, p. 397, Mar. 2013.
[12] D. Hoey and P. Benshoof, "Civil GPS systems and potential vulnerabilities," Tech. Rep., 2005.

[13] A. Khattab, Y. A. Fahmy, and A. Abdel Wahab, "High accuracy GPS-free vehicle localization framework via an INS-assisted single RSU," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 5, May 2015, Art. no. 795036.
[14] R. Kurazume and S. Hirose, "Study on cooperative positioning system: Optimum moving strategies for CPS-III," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, Nov. 2002, pp. 2896–2903.
[15] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments," in *Robotics: Science and Systems*, vol. 4. 2007, p. 1.
[16] S. Lin and B. W. Kernighan, "An effective heuristic algorithm for the traveling-salesman problem," *Oper. Res.*, vol. 21, no. 2, pp. 498–516, Apr. 1973.
[17] G. Mao, S. Drake, and B. D. O. Anderson, "Design of an extended Kalman filter for UAV localization," in *Proc. Inf., Decis. Control*, Feb. 2007, pp. 224–229 .
[18] S. Niekum. *AR Track Alvar*. Accessed: Apr. 7, 2019. [Online]. Available: http://wiki.ros.org/ar_track_alvar
[19] C.-H. Ou, "A roadside unit-based localization scheme for vehicular ad hoc networks," *Int. J. Commun. Syst.*, vol. 27, no. 1, pp. 135–150, Jan. 2014.
[20] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multivehicle systems with nonholonomic constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 98–104, Jan. 2007.
[21] S. Rathinam and R. Sharma, "A multiple vehicle path covering problem with localization constraints: Formulation and algorithms," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 3746–3751 .
[22] G. Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA J. Comput.*, vol. 3, no. 4, pp. 376–384, 1991.
[23] S. Roumeliotis and G. Bekey, "Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots," in *Proc. Millennium Conf. IEEE Int. Conf. Robot. Automat. Symp. (ICRA)*, vol. 3, Nov. 2002, pp. 2958–2965.
[24] A. C. Sanderson,"A distributed algorithm for cooperative navigation among multiple mobile robots," *Adv. Robot.*, vol. 12, no. 4, pp. 335–349, Jan. 1997.
[25] R. Sharma, R. W. Beard, C. N. Taylor, and S. Quebe, "Graph-based observability analysis of bearing-only cooperative localization," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 522–529, Apr. 2012.
[26] R. Sharma, S. Quebe, R. W. Beard, and C. N. Taylor, "Bearing-only cooperative localization," *J. Intell. Robot. Syst.*, vol. 72, nos. 3–4, pp. 429–440, Dec. 2013.
[27] Y. Song and J. W. Grizzle, "The extended Kalman filter as a local asymptotic observer for nonlinear discrete-time systems," in *Proc. Amer. Control Conf.*, Jun. 1992, pp. 3365–3369 .
[28] K. Sundar, S. Misra, S. Rathinam, and R. Sharma, "Routing unmanned vehicles in GPS-denied environments," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 62–71 .
[29] K. Sundar and S. Rathinam, "Route planning algorithms for unmanned aerial vehicles with refueling constraints," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2012, pp. 3266–3271 .
[30] K. Sundar and S. Rathinam, "Algorithms for routing an unmanned aerial vehicle in the presence of refueling depots," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 287–294, Jan. 2014.
[31] K. Sundar, S. Rathinam, and R. Sharma, "Path planning for unmanned vehicles with localization constraints," *Optim. Lett.*, vol. 13, no. 5, pp. 993–1009, Jul. 2019.
[32] K. Sundar, S. Srinivasan, S. Misra, S. Rathinam, and R. Sharma, "Landmark placement for localization in a GPS-denied environment," in *Proc. Annu. Amer. Control Conf. (ACC)*, Jun. 2018, pp. 2769–2775 .
[33] K. Sundar, S. Venkatachalam, and S. Rathinam, "Formulations and algorithms for the multiple depot, fuel-constrained, multiple vehicle routing problem," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 6489–6494.
[34] K. Sundar, S. Venkatachalam, and S. Rathinam, "Analysis of mixed-integer linear programming formulations for a fuel-constrained multiple vehicle routing problem," *Unmanned Syst.*, vol. 05, no. 04, pp. 197–207, Oct. 2017.
[35] TurtleBot. *Turtlebot 2*. Accessed: Apr. 7, 2019. [Online]. Available: https://www.turtlebot.com/
[36] S. Weiss, D. Scaramuzza, and R. Siegwart, "Monocular-SLAM-based navigation for autonomous micro helicopters in GPS-denied environments," *J. Field Robot.*, vol. 28, no. 6, pp. 854–874, Nov. 2011.
[37] D. Wong, D. Deguchi, I. Ide, and H. Murase, "Single camera vehicle localization using SURF scale and dynamic time warping," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 681–686.

SOHUM MISRA received the B.Tech. degree in electronics and instrumentation engineering from the Heritage Institute of Technology, Kolkata, India, in 2015. He is currently pursuing the Ph.D. degree in aerospace engineering with the University of Cincinnati, Cincinnati, OH, USA. He is also a Graduate Research Assistant with the R.I.S.C Lab. He has nine technical publications. He is currently working on path planning and navigation problems in GPS-denied or restricted environments. His research interests are controls and automation, path planning, optimization, sensor integration, robotics, and the Internet of Things.

BINGYU WANG is currently pursuing the Ph.D. degree in mechanical engineering with the Department of Mechanical Engineering, Texas A&M University, College Station, TX, USA. His current research interests include motion planning of autonomous vehicles, connected transportation systems, and multiagent autonomous systems.

KAARTHIK SUNDAR received the Ph.D. degree in mechanical engineering from Texas A&M University, College Station, TX, USA, in 2016. He is currently a Research Scientist with the Information Systems and Modeling Division, Los Alamos National Laboratory, Los Alamos, NM, USA. His research interests include problems pertaining to vehicle routing, path planning, and control for unmanned/autonomous systems; numerical optimal control, estimation, and large-scale optimization problems in power and gas networks; combinatorial optimization; and global optimization for mixed-integer nonlinear programs.

RAJNIKANT SHARMA (Member, IEEE) received the B.E. degree in electrical engineering from the University of Rajasthan, Jaipur, India, in 2003, the M.E. degree in aerospace engineering from the Indian Institute of Science, Bengaluru, India, in 2005, and the Ph.D. degree in electrical engineering from the Brigham Young University, Provo, UT, USA, in 2011. From 2011 to 2013, he was a Postdoctoral Fellow at the Center for UAS Research, US Air Force Academy, CO, USA. From 2005 to 2007, he was a Scientist B at the Center for Airborne Systems, Defense Research and Development Organization, Ministry of Defense, Bengaluru. He is currently an Assistant Professor with the Aerospace and Engineering Mechanics Department, University of Cincinnati. His primary research interests are guidance, navigation, and control of unmanned aerial vehicles and multiple vehicle coordination, control, and localization. He is a member of the AIAA Community.

SIVAKUMAR RATHINAM received the B.Tech. degree in mechanical engineering from IIT Madras, India, in 1999, the M.S. degree in mechanical engineering from Texas A&M University, College Station, TX, USA, in 2001, the M.S. degree in computer science from the University of California, Berkeley, CA, USA, in 2006, and the Ph.D. degree in civil systems engineering from the University of California, in 2007. He is currently an Associate Professor with the Mechanical Engineering Department, Texas A&M University. His researches focus on motion planning and control of autonomous vehicles, collaborative decision making, combinatorial optimization, vision-based control, and air traffic control.