29th Congress of the International Council of the Aeronautical Sciences
St. Petersburg, Russia; September 7-12, 2014

AUTONOMOUS TARGETED FLIGHT OF A ROTARY-WING MICRO AERIAL VEHICLE IN INDOOR, GPS-DENIED ENVIRONMENTS

Svetlana Potyagaylo*, Omri Rand*
***Faculty of Aerospace Engineering, Technion - Israel Institute of Technology, Haifa, 32000, Israel**

Keywords: *Indoor Navigation, Autonomous Systems, MAV, Rotary-Wing Vehicles, Simulation*

Abstract

This work focuses on targeted flight of Rotary-Wing Micro Aerial Vehicles (RW MAV) in indoor environments. In such missions, Global Positioning System (GPS) signals may be unavailable and a map of the environment is unknown prior to the flight. Thus, the MAV should be able to estimate its position, build a map of the environment, and plan an obstacle-free flight path towards the target, while taking into consideration its maneuverability limitations. This paper presents a modular system that comprises required components for these tasks. The proposed methodology relies on using a single laser range scanner. We present the components of the proposed system and algorithms, including a high level module for mission planning and decision making. The system was checked in different simulated environments under the realistic level of disturbances and system noise. The obtained results illustrate the effectiveness of the system for GPS-denied navigation of RW MAVs.

1 Introduction

Nowadays, RW MAVs may be considered as an attractive autonomous platform for various indoor flight missions because of their small size, superior agility, high maneuverability, and hovering capability. Development of a system that enables a RW MAV to fly autonomously in indoor environments is a challenging task. Such a system should consist of numerous algorithms including but not limited to: mission planning, motion planning, sensing, position estimation, mapping and path planning. The existence of system errors, disturbances, uncertainty and changes in the environment during the flight add complications to the development of the required algorithms and to the ability to combine these algorithms.

Indoor environments imply additional difficulties while developing algorithms for autonomous flight since one may not rely on GPS signals for position data, due to poor reception, or no reception at all, of the signal. In such missions, the generation of a map of the unknown environment, while simultaneously localization of the vehicle, based only on available sensory observations is required. This task is commonly known as Simultaneous Localization and Mapping (SLAM) [1]. Existed SLAM algorithms make use of certain features of flight missions like returning to previously visited locations. With the aid of loop closure algorithms, they are able to significantly improve the accuracy of vehicle's position estimation and the quality and consistency of the generated map. However, these algorithms are not suitable for a targeted flight, i.e. a problem of navigating a vehicle from a known initial position to a defined goal location, since the vehicle may not return to a previously visited area.

In this paper, we present a comprehensive approach for targeted autonomous flight of RW MAVs in a GPS-denied environment where the map of the environment is a priori unknown. We assume that only the initial position and orientation of the MAV are given, as well as the goal position and orientation, which are defined by relative distance and azimuth. The primary onboard sensor is a lightweight laser range finder. The methodology presented here is modular and generic in the sense that each of its components is internally uncoupled with the others. We also present simulation results on a single rotor helicopter showing the capability for a fully autonomous targeted flight.

2 Related Work

As mentioned earlier, a system for an autonomous flight of RW MAVs in GPS-denied, unknown environments should include several essential algorithms for vehicle's position estimation, simultaneously with building and updating a map of the environment; planning a path towards the goal, while avoiding obstacles discovered along the path; and motion planning including calculation of control commands that will lead a vehicle along the desired trajectory. A significant progress has been made in each of the above listed tasks. The SLAM task has been studied both for ground robots [2, 3, 4] and aerial vehicles [5, 6, 7]. For path planning, there are a plenty of algorithms including graph search methods [8, 9, 10] and probabilistic planning methods [11, 12]. The motion planning task may be solved by optimal control techniques [13] or using motion primitives [14], to name a few.

In this work, we develop an Autonomous Indoor Targeted Flight System (AITFS) that comprises several modules and algorithms. For solving the SLAM problem, the system makes use of a novel algorithm based on representation of the environment in form of "*occupancy grid*" (OG) [15]. The advantage of OGs is their ability to represent any obstacle without relying on predefined features or forms. OGs also maintain information about undiscovered or unobserved areas of the environment, which can be essential in the path planning task. To estimate the vehicle's position, we employ an adaptive direct search method [16]. The path planning method of the AITFS includes an A* search algorithm [8] and a Potential Field method [17], combination of which allows the system to plan an obstacle-free path while taking into account vehicle's maneuverability limitations. Several essential features are added to these methods to maintain low computational load and to guarantee feasibility of the solution. For computing control commands, an Inverse Simulation (IS) approach [18] is applied, while the generation of the required trajectory is provided by a novel scheme that takes into account the current flight regime (flight, hover, etc.) of the RW MAV.

3 Methodology Description

We present now an overview of our methodology and of the modular system that executes it. A schematic diagram of the system and of its components is depicted in Fig. 1. The system includes a "Simultaneous Localization and Mapping module" (SLAM module), a "Path Planning module" and a "Motion Planning module". The system architecture is modular so that each pair of its components or even each single component can be used independently of the others as well.

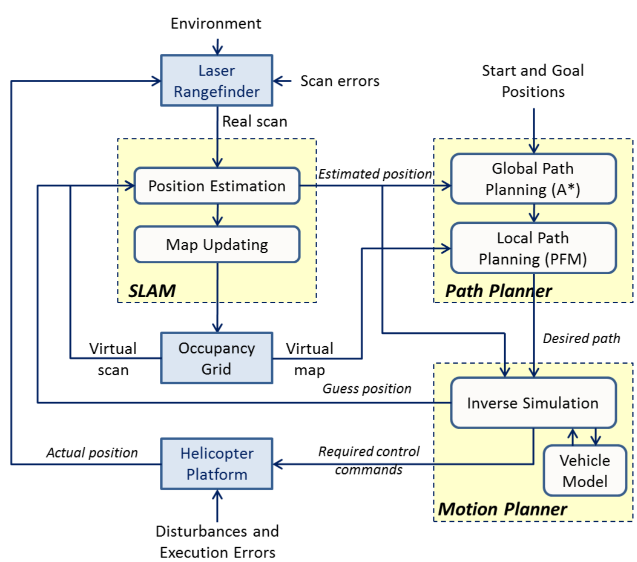


Fig. 1 The main modules of the proposed system (AITFS).

The obtained control commands are executed in a "*simulator*", a test-bed simulative tool for autonomous helicopters that includes a detailed simulative helicopter model. This model accounts for the execution errors and external disturbances, such as wind gusts. The simulator is considered as part of the proposed system, but is utilized also for the simulation of the real laser range finder and for providing the comparison between actual and estimated helicopter performances, position estimation and map quality. In real missions two blocks that simulate the helicopter platform and the laser range finder are replaced by the actual physical components. The detailed description of the system modules and algorithms is given in Section 4.

In the AITFS, two main reference frames are used: the global East-North-Up (ENU) coordinate system and the body coordinate system Fig. 2. The global ENU coordinate system is an inertial Earth-fixed frame and is used for navigation and motion planning. The origin is arbitrarily fixed to a point on the Earth surface, the X_e-axis points toward the East, the Y_e-axis points toward the North, and the X_e-axis points upwards. In this frame, the position vector $\mathbf{x}_e = \{x_e, y_e, z_e\}$ [1] and the velocity vector $\mathbf{v}_e = \{v_{xe}, v_{ye}, v_{ze}\}$ are defined.

The body coordinate system is fixed to the body of the vehicle and is used to define the forces and moments of the helicopter. The origin O_b is located at the center of gravity (CG) of the helicopter, the X_b-axis points forward, the Y_b-axis points to the left side of the vehicle, the Z_b-axis points upward. It was assumed that CG is located at a constant position relative to the vehicle geometry. In this frame, the linear velocity vector of the vehicle $\mathbf{v}_b = \{u, v, w\}$ and the angular velocity $\omega_b = \{p, q, r\}$ are defined. The orientation of the helicopter or, more specifically, the orientation of the body-fixed frame with respect to the global frame is described by Euler angles ψ, θ and ϕ.

In the SLAM and Path Planning modules, we assume a static 2D environment and use the planar projection of the aforementioned coordinate frames. In this case, the vehicle pose $\mathbf{p} = \{x, y, \psi\}$ that consists of the vehicle's coordinates $\{x_e, y_e\}$ and orientation (azimuth angle) ψ is defined as well.

The main steps of computing a flight path ac-

[1] Hereinafter bold stands for vectors.

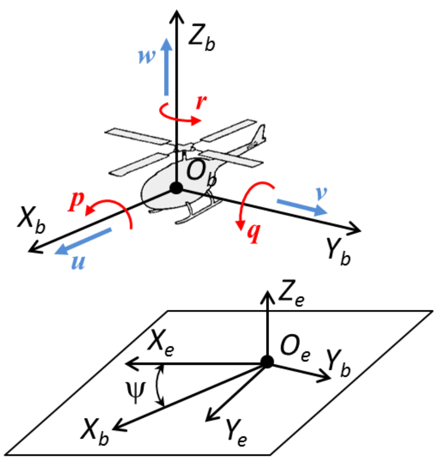


Fig. 2 The helicopter coordinate systems: the global ENU frame and the body frame.

cording to our methodology may be described as follows. Initially, the MAV is located at a start pose $\mathbf{p}_0$. A laser range finder provides initial data about the obstacles ahead. Then, the Path Planning module is employed to provide a desired path towards a target that is free of collision with the obstacles and represents a sequence of poses $\mathcal{P} = \{\mathbf{p}_0, \ldots, \mathbf{p}_{target}\}$. The Motion Planning Module takes this path $\mathcal{P}$, transforms it to a trajectory $\mathcal{Q}$ by time parameterization technique and calculates a sequence of controls $\mathcal{U} = \{\mathbf{u}(t_i)\}$ for each time instance t_i that are needed to fly along this trajectory. After the execution of the control commands for the next time step, the vehicle will be located at a pose $\mathbf{p}_{true} = (x_t, y_t, \psi_t)$ that is unknown and subject to be found. The SLAM modules outputs the estimated vehicle's pose $\mathbf{p}_{est} = (x_e, y_e, \psi_e)$ based on the sensor measurements, taken from the actual pose, and the virtual map of the environment stored in the memory. Simultaneously, the map updating process is executed. On the next iteration, all the above stages are repeated, until arriving at the target.

4 Main Modules of the System

4.1 SLAM

As mentioned above, navigation in a priori unknown GPS-denied environments requires building a map of the environment simultaneously with an estimation of the vehicle position within this map. These two problems can be solved apart relatively easily, assuming either the presence of a map of the environment, for the localization problem, or knowing the vehicle pose (e.g. by using a GPS), for the mapping problem. However, when there is no map of the environment and the pose of the vehicle has to be estimated, the two problems must be solved simultaneously by a SLAM method.

The proposed SLAM method consists of several essential algorithms: a virtual scan, a scan matching procedure, and a map building and updating algorithm. The algorithm uses the OG, each cell of which contains the number of "hits" of the laser scan registered for the corresponding sub-area of the environment. The OG is also used for performing a *virtual scan* of the environment produced by a series of ray casting operations, searching for occupied cells along the virtual laser ray. Fig. 3 shows the proposed solution schematically. At each time point, the vehicle is located at the "true" (actual) pose $\mathbf{p}_{true} = (x_t, y_t, \psi_t)$. From this point, the laser range finder senses obstacles in the environment and produces an actual scan of the environment. In order to estimate the vehicle position, an initial "guess" pose $\mathbf{p}_{guess} = (x_g, y_g, \psi_g)$ is provided by the Motion Planning Module. The virtual scan of the environment is produced from this guessed position and is similar to the actual scan. The difference between the virtual and actual scans is served as a basis for the position estimation obtained by the *scan matching* procedure. The scan matching algorithm matches the scans one against another and searches for a shift between the actual and guess vehicle's poses that minimizes a cost function using the adaptive direct search algorithm [16]. The cost function represents the discrepancy between the roto-translated and interpolated real scan and the virtual scan, only for the range covered by both [19]. This means that the virtual scan points with range values out of scanner FOV are not matched with the appropriate points from the real scan, and vice versa. The cost function is normalized by the number of valid points N_{valid}. This enables to normalize the total cost values for different matching attempts with different number of contribution points:

$$f = \frac{1}{N_{valid}} \sum_{i=1}^{N_{valid}} \left| \left(r''\right)^i - r^i_{virt} \right|, \tag{1}$$

where r'' and r_{virt} are the range of the real scan after roto-translation and the range of the virtual scan, respectively.

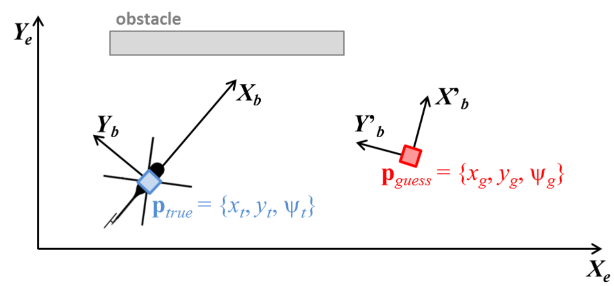


Fig. 3 The SLAM problem statement.

After the scan matching algorithm completes with the estimates of the position of the MAV, the map updating process is performed only if the matching succeeds, i.e. the result of the scan matching is below a defined threshold. This is required since the OG serves as an average of all previous laser scans and thus, if the virtual scan is not accurate enough, the updating of the OG will not be accurate as well. This would lead to a rapidly growing error in the position estimation.

In order to measure the quality of the SLAM algorithm, we use the following measure that estimates the weighted mean distance from the points in the virtual map to the real obstacles while taking into account the hit values of the cells:

$$\mathrm{ME} = \frac{\sum_{i=1}^{N_i} \sum_{j=1}^{N_j} w_{ij} d_{ij}}{\sum_{i=1}^{N_i} \sum_{j=1}^{N_j} w_{ij}}, \tag{2}$$

where w_{ij} is the hit value of the OG's cell ij; d_{ij} is the distance from the cell center to the nearest real obstacle; N_i, N_j is the number of the cells in the OG along the X_e and Y_e axes, respectively.

The simulation test-bed that was constructed in this work and described in Section 4.4 enables to calculate such measure in contrast with the case of real experiments where exact information about the true positions of the obstacles may be hardly achievable.

4.2 Path Planning

As mentioned above, the path planner combines an A* algorithm and a Potential Field Method (PFM). The A* algorithm provides a global path from the current vehicle's position towards the goal position based on the currently available information of the environment. Once the global path is found, the waypoint as a farthest visible point along the A* path, within line of sight with respect to the current MAV location, is calculated, and the PFM method is employed to plan a path to that waypoint. In the case the goal is located in the field of view of the vehicle, the waipoint and the goal become identical.

In this module, the A* algorithm is applied over a coarse occupancy grid using significantly larger cells, as compared to the SLAM OG. Eight directions of movement (North/South/West/East, and diagonals) are allowed. The algorithm makes use of the diagonal heuristic function [20]:

$$h = D_d h_d + D_s \left(h_s - 2h_d\right), \tag{3}$$

where $D_d = \sqrt{2}$ is the traverse cost along the diagonal, $D_s = 1$ is the traverse cost in the straight direction, h_d is the number of diagonal steps; h_s is the number of straight steps. In order to maintain admissibility of this heuristics, the diagonal cost should be less than the cost of two straight steps: $D_d \leq 2D_s$.

The PFM considers the vehicle as a point mass, moving in a force field produced by repulsive forces from the obstacles, and a single attractive force to the goal (or waypoint). In this work, the intensity of the attractive force is positive and constant over all possible position of the vehicle. Each occupied cell in the OG is considered as an obstacle, producing a repulsive force with negative intensity, and proportional to the occupancy value of that cell, and inversely proportional to the distance from the cell's center to the vehicle's position:

$$\begin{aligned} \mathbf{F}_{goal} &= I_{goal} \frac{\mathbf{x}_{goal} - \mathbf{x}}{d_{goal}}; \\ \mathbf{F}_i &= I_{obst_i} w_i \frac{\mathbf{x}_i - \mathbf{x}}{d_i}, \end{aligned} \tag{4}$$

where I_{goal} is the constant positive intensity level of the goal, $\mathbf{x}_{goal} = \left(x_{goal}, y_{goal}\right)$ is the goal position, $\mathbf{x} = (x, y)$ is the current (estimated) vehicle position obtained by the SLAM module, d_{goal} is the distance from the current UAV position to the goal position. Similarly, I_{obst} is the negative intensity level of the obstacle, d_i is a distance from the cell center to the center of mass of the vehicle, w_i is the hit count of the cell, and $\mathbf{x}_i$ is the position of the cell center.

Additionally, the PFM considers only cells that are located in the fore third of the field of view. This was done to reduce effects of areas that are no longer between the vehicle and its next waypoint.

4.3 Motion Planning

In this work, the notion "*motion planning*" stands for the task that combines two problems: the generation of a continuous time sequence of vehicle's positions and velocities from one given location to another specific location, in addition to the computing of the time history of control commands that will move the vehicle to a desired position. This module is the only module that works with the actual model of the vehicle since it feeds the specific helicopter characteristics to the whole system. The development of a simulation model (linear or non-linear) is therefore a prerequisite for further use in any of the motion planning methods.

We use both nonlinear and linear representations of the helicopter model. The nonlinear and linear dynamic models have 6 DOF and the following state and control vectors:

$$\begin{aligned} \mathbf{s} &= \{u, v, w, p, q, r, \psi, \theta, \phi\}; \\ \mathbf{u} &= \{\theta_0, \theta_{1c}, \theta_{1s}, \theta_{tr}\}, \end{aligned} \tag{5}$$

where θ_0 is the main rotor collective pitch angle; θ_{1c}, θ_{1s} are the main rotor lateral and longitudinal cyclic pitch angles; θ_{tr} is the tail rotor collective pitch angle.

It was assumed that the main rotor has rigid blades with uniformly distributed mass along the blade and linear twist. The blades have an additional DOF of flapping. At each blade section the lift and drag forces are calculated using nonlinear aerodynamic tables. The total lift and drag forces are obtained by integrating the forces of each blade section along the blade. The tail rotor blades are assumed to be rigid and non-flapping. The expressions for tail rotor forces and moments are similar to those for the main rotor but include the interaction effects. We assume a quasi-steady flapping dynamics, according to which the rotor responses to cyclic commands is extremely fast.

The proposed methodology for motion planning and control task includes two-phase algorithm. Firstly, a desired trajectory $\mathcal{Q}$ that will be used as an input within the IS algorithm is developed based on a predefined path from the Path Planning Module. This trajectory is constructed to maintain the principal geometry of the path while providing smooth changing of the desired parameters and vehicle command limitations. The second phase is the IS algorithm that calculates a sequence of controls $\mathcal{U}$ that are required to fly along the planned trajectory. In this paper, the integration-based inverse simulation method is used [18]. It initially guesses the control inputs, integrates the vehicle motion equations, to achieve the desired output vector at the next iteration. The difference between the actual flight vector and the desired flight vector is then used to calculate the estimation of the control inputs for the next path computation step. In this work, the dimension of the control sequence is equal to the dimension of the trajectory sequence that consists of three velocities and the azimuth angle $\mathbf{q} = \left\{v_{xe}, v_{ye}, v_{ze}, \psi\right\}$ [19].

4.4 Integration of the System Modules

In this section, the main components of the methodology described in the previous subsections are combined to an Autonomous Indoor Targeted Flight System (AITFS) for RW UAVs. The modular structure of the system allows independent creation and usage of the system modules, clear and flexible design, and multiple functionalities. In the proposed system, each module is aimed towards one specific function, and this enables studying the effects and performance of a specific module individually.

However, the integration of the system components is not a trivial task since interconnecting links between modules have to be set in order to enable the collective operation of the system modules. In addition, a high decision making level has to be developed to ensure achievement of the mission goal and to plan actions that will diminish negative outcomes in cases of uncertainties of the environment and the vehicle state. The two problems that may arise from such situations and lead to interruption of the common flow "*position estimation and map updating* $\rightarrow$ *path planning* $\rightarrow$ *trajectory and command computation*" will be discussed.

For example, there may be situations in which arrival to the predefined goal location is impossible due the obstacles discovered during the flight. In this case, a MAV has to make an appropriate decision whether to fly back, to fly to the position nearest to the goal or to remain over a specific location. One additional situation that requires special attention is faults, i.e. difficulties of the system modules to produce their outputs. In the proposed AITFS, the A* algorithm and the PFM used in the Path Planner Module are known to be complete if a goal (or a waypoint) is reachable. The IS algorithm used in the Motion Planning Module has very high convergence since the trajectory generation method produces smooth trajectories with small changes of the required parameters. However, the SLAM algorithm may produce solutions with non-minimal values of the cost function due to a bad initial guess or a small mutual area of actual and virtual scans. Isolated instances of this fault do not affect the overall performance of the system since in the next steps the vehicle will localize itself and the map updating will be resumed. However, such a fault in several consecutive steps may lead to divergence and vehicle loss. To prevent this, the high level module causes the vehicle to slow down, to hover or to search for the best azimuth position until the

SLAM algorithm will succeed to estimate the vehicle's pose. These situations will be illustrate by examples in Section 5 in details.

To assess the effectiveness of the proposed system, a simulation test-bed was developed. The test-bed includes a simulation model of a conventional configuration helicopter and a simulation model of a real laser range finder along with realistic levels of execution errors and external disturbances. The simulation test-bed is implemented using MATLAB®. The computational cost of the system modules was evaluated in terms of the percentages of the total running time. The running time of the SLAM module, including the Virtual Scan, the Scan Matching procedure and the Map Updating, is approximately 60% of the total cost, while the running time of the Motion Planning is about 20% of the total cost and the running time of the Path Planning is about 2%.

5 Results and Discussion

5.1 Simulation Setup

The helicopter and the real laser rangefinder we chose as prototypes is the SR RTF helicopter, manufactured by Blade, Horizon Hobby, Inc. [21] and the Hokuyo laser rangefinder [22]. The main parameters of the helicopter, laser range finder and algorithms are given in Table 1.

Table 1 The values of the main parameters.

Parameter	Value
Helicopter main rotor diameter [mm]	552
Helicopter tail rotor diameter [mm]	82
Helicopter weight [g]	340
Helicopter length [mm]	485
Minimum detection radius [mm]	25
Maximum detection radius [mm]	30000
Angular resolution [deg]	0.25
Maximum bearing angle [deg]	± 135
OG resolution [mm]	10
A* OG resolution [mm]	500

Trimming of the helicopter model was carried out using the RAPiD rotorcraft analysis software package [23] that is designed to model and analyze general rotorcraft and rotary-wing based configurations. The simulations were carried out assuming the realistic level for sensor errors of 1.5% and for execution errors of 1%.

5.2 Simulation Results

We will start with the discussion of two special cases mentioned earlier: the problem of the SLAM algorithm failure and the problem when a goal location is unreachable. Fig. 4 presents two snapshots of a simulated structured environment. The scenario was realistic and included the MAV autonomous flight from the known start position to a given goal location. In the first case (left side of Fig. 4), the vehicle had to enter a room to arrive to a goal. This part of the flight represents a quite challenging task for autonomous platforms with an only environment sensor. While entering a new closed segment of the unmapped environment, the changes between the map already stored in the memory (or virtual scan) and the real scan of a new area may be dramatic. Such discrepancies may lead to difficulties in position estimation and, as a consequence, to the failure of the SLAM algorithm. This situation may be observed on Fig. 5(a) where values of the cost function are shown for the entire flight. Note, that the admissible threshold for the cost function was 30 mm. To resolve the problem, the high level of the AITFS causes the vehicle to slow down, hover and change only its heading angle to successfully complete the position estimation and map updating processes.

The problem when the MAV can not reach a goal position due to inaccessibility of full information of the environment is shown on the right side of Fig. 4. In this case, the goal was located in the closed room. The MAV had to explore all the environment in order to ensure that there are no possible paths to the goal. The corresponding values of the cost function are shown on Fig. 5(b) and they are much lower for this case.

The map quality for this case is approximately ME = 8.8 mm. This measure is within the order of magnitude of the OG's cells (10 mm), i.e. the resultant virtual map is highly accurate.

Fig. 6 presents three snapshots of another

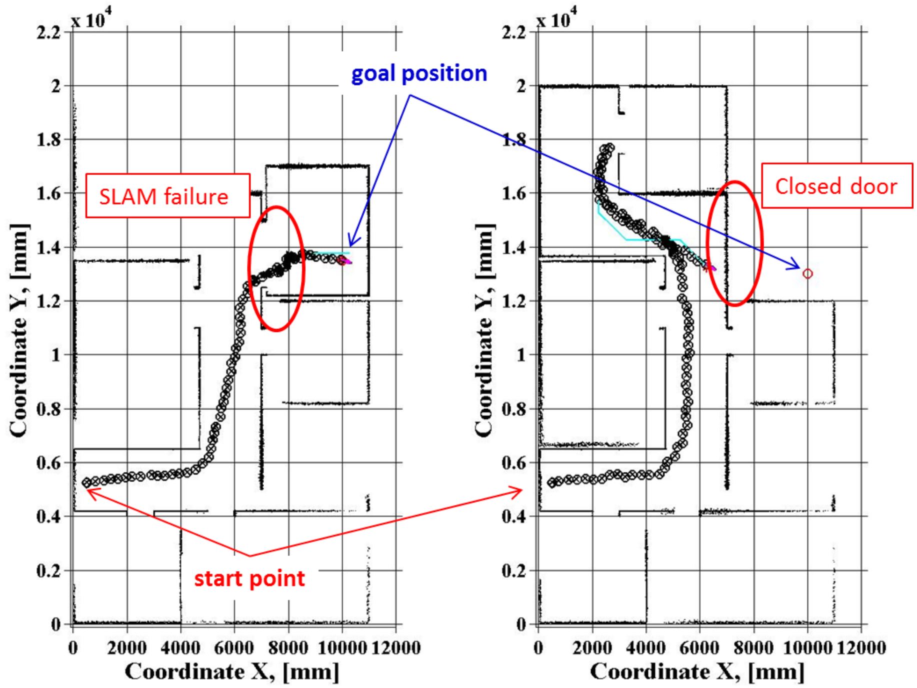


Fig. 4 The snapshots of the generated mp of a simulated environment for the cases of reachable and unreachable goal.

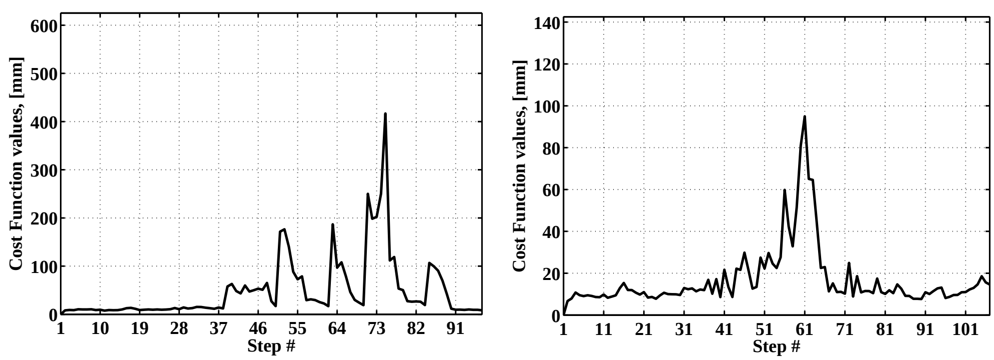
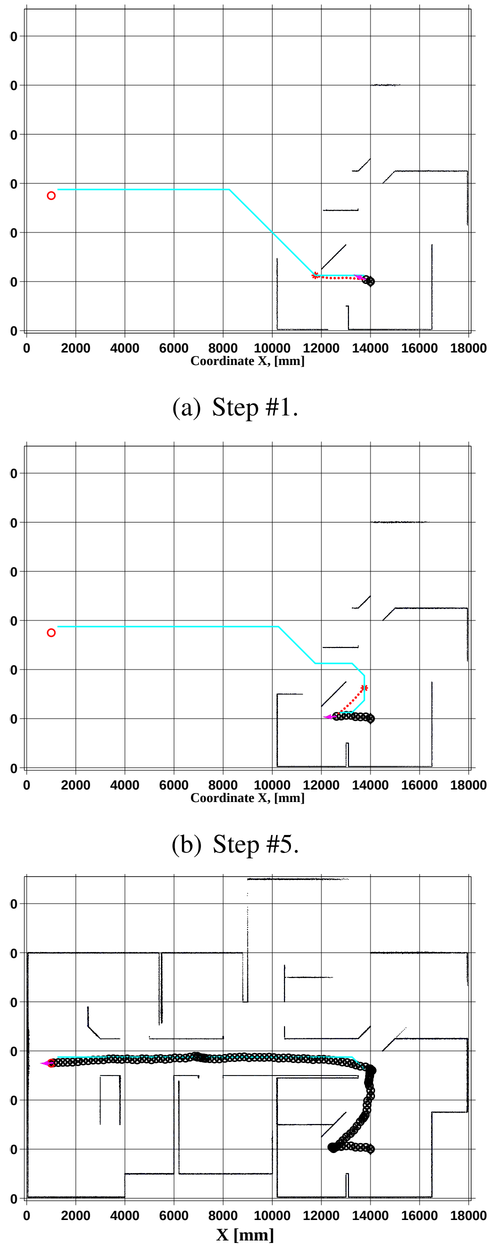


Fig. 5 The values of the cost function for two cases (reachable and unreachable goal).

complicated environment and the evolving map, planned paths, and executed path. During the flight, the estimated A* and PFM paths are unremittingly checked for obstacle avoidance, with respect to newly detected obstacles, and update accordingly. This evolution of the map and paths is illustrated by consecutive snapshots. For example, at the first time step (Fig. 6(a)), the A* algorithm estimates the path towards the target based only on the first obtained scan. The vehicle begins to turn left to go around the nearby obstacle. After 4 more steps, the passageway becomes too narrow, the updated underlying A* occupancy grid therefore detects an obstructed path, and recalculates (Fig. 6(b)). This path is in turn updated further, and the process repeats. Fig. 6(c) presents the final estimated and actual paths along with the final map.

6 Conclusions

We presented a comprehensive, modular system, designed for a targeted flight of autonomous RW MAVs in unknown GPS-denied environments.



(a) Step #1.

(b) Step #5.

(c) The final virtual map.

Fig. 6 Snapshots of the map and path evolutions for a simulated environment. A* planned path is in solid cyan line, asterisk shows the next way point, while red dots mark the PFM locally planned path, black circles an 'x' marks show the vehicle estimated and true positions, respectively.

The proposed system consists of independent modules for simultaneous estimation of vehicle's position and mapping of the environment, planning a feasible and obstacle-free path towards a goal, and generation a trajectory and control commands required to fly along that trajectory. The system also includes an additional module that exposes such features as mission planning, situation awareness, and decision making. The main challenges in the task of a targeted flight were the following: (1) Localization of the vehicle in a priory unknown environments without additional aid of external sensors or algorithms for loop closure. (2) Avoidance of collision of the MAV with obstacles while keeping the flight path to be as short as possible (3) Taking into consideration maneuvering limitations of the vehicle.

The simulation results of autonomous MAV flight in several simulated environments were presented as well. These results demonstrated that the system modules provide highly accurate results and successful arriving of the MAV to goal positions without significant accumulated drift. As part of future work, the AITFS will be fully validated with an aerial platform in real world environments.

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous Localisation and Mapping (SLAM): Part I The essential algorithms," *IEEE Robotics and Automation Magazine*, vol. 2, 2006.

[2] S. Thrun and M. Montemerlo, "The GraphSLAM algorithm with applications to large-scale mapping of urban structures," *International Journal on Robotics Research*, vol. 25, no. 5/6, pp. 403–430, 2005.

[3] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates," in *International Journal of Robotics Research*, vol. 26, pp. 1125–1153, 2007.

[4] D. Hähnel, W. Burgard, D. Fox, and S. Thrun, "An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Las Vegas, Nevada, USA), October-November 2003.

[5] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," in *Proceedings of the IEEE International Conference on Robotics and Automa-*

tion (ICRA), (Kobe, Japan), pp. 2878–2883, May 2009.

[6] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, "Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments," in *IARC First Symposium on Indoor Flight Issues*, July 2009.

[7] S. Shen, N. Michael, and V. Kumar, "Autonomous multi-floor indoor navigation with a computationally constrained MAV," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, (Shangai, China), pp. 20–25, May 2011.

[8] S. J. Russell and P. Norvig, *Artificial Intelligence - A Modern Approach*. Pearson Education, 3rd ed., 2010.

[9] A. T. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3310–3317, May 1994.

[10] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime dynamic A*: An anytime, replanning algorithm," in *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2005.

[11] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[12] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Technical Report 98-11, Computer Science Dept., Iowa State University*, 1998.

[13] A. E. Bryson, *Applied Optimal Control*. Hemisphere Publishing, New York, 1975.

[14] E. Frazzoli, M. A. Dahleh, and E. Feron, "Maneuver-based motion planning for nonlinear systems with symmetries," *IEEE Transactions on Robotics*, vol. 21, pp. 1077–1091, December 2005.

[15] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, 2003.

[16] C. Friedman, I. Chopra, and O. Rand, "Highly accurate simultaneous localization and mapping for Rotary Wing UAVs," in *Proceedings of American Helicopter Society* 68^{th} *Annual Forum*, (Fort Worth, Texas), 2012.

[17] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *International Journal of Robotics Research*, vol. 5, no. 1, 1986.

[18] R. A. Hess and C. hau Gao, "A generalized algorithm for inverse simulation applied to helicopter maneuvering flight," *Journal of the American Helicopter Society*, vol. 38, no. 4, pp. 3–15, 1993.

[19] S. Potyagaylo, *Planning and Operational Algorithms for Autonomous Helicopters*. PhD thesis, Technion, Haifa, Israel, 2013.

[20] A. Patel, *AmitâĂŹs thoughts on path-finding and A-star*. http://theory.stanford.edu/amitp/GameProgramming/, 2003.

[21] Blade, Horizon Hobby, Inc. http://www.bladehelis.com.

[22] Hokuyo, "UTM-30LX," tech. rep., 2009. http://www.hokuyo-aut.jp/02sensor/07scanner/utm_30lx.html.

[23] O. Rand and S. M. Barkai, "Numerical evaluation of the equations of motion of helicopter blades and symbolic exactness," *Journal of the American Helicopter Society*, vol. 40, no. 1, pp. 59–71, 1995.

7 Contact Author Email Address

Svetlana Potyagaylo: svetap@tx.technion.ac.il
Omri Rand: omri@sni.technion.ac.il

Copyright Statement