AIAA Guidance, Navigation and Control Conference and Exhibit
18 - 21 August 2008, Honolulu, Hawaii

AIAA 2008-6836

A Dynamic Path Generation Method for a UAV Swarm in the Urban Environment

David Hyunchul Shim[1]
KAIST, Daejeon, South Korea

and

Shankar Sastry[2]
University of California, Berkeley, CA94720

In this paper, a dynamic path planning method for swarming of unmanned aerial vehicles (UAVs) in the urban environment is presented. For the missions of a team of UAVs in a complex environment, conflict-free paths for all vehicles should be calculated dynamically in real-time using the latest information on the changes in the surroundings such as pop-up threats. Therefore, we propose a hierarchical dynamic path planner that consists of a offline path planning and a real-time model predictive trajectory generator. In this framework, many existing and proven off-line algorithms can be deployed for globally optimal path planning. The pre-computed trajectory is sent to the model predictive layer, which generates locally feasible trajectory free from conflicts. In this manner, each vehicle is able to fly along its designated path to reach the destination while avoiding obstacles or vehicles in collision path with minimal deviation from the designated path. For validation, the proposed algorithm is applied to a deployment scenario of sixteen rotary-wing UAVs flying in a cluttered urban area and showed a satisfactory performance.

I. Introduction

Unmanned aerial vehicles have been proven as an effective and affordable solutions that complement and support many operations traditionally performed by human. For now, UAVs are limitedly applied to solo missions and they require constant monitoring by a number of well-trained operators. However, as the onboard computing power increases and the reliability of overall system improves, it is anticipated that UAVs will operate as a team in a complex environment to achieve given mission objectives in near future.

In Nature, there are many species live in a large colony or in a group. Bees, ants, or termites form a large-scale colony where the individuals function as a part of the entire system, such as foragers, builders or soldiers. More evolved animals such as fish or birds protect themselves from predators by voluntarily joining a large and dense group. Wolves live in a hierarchical group and hunt a prey using group tactics. Their group behaviors referred to as shoaling, schooling, flocking, or swarming, differ slightly in shapes, numbers, or purpose. Inspired by these cases, it has been suggested that, in a simplest form of argument, if a single UAV has been so effective, a swarm of UAVs would be even more capable to carry out more difficult missions with higher resilience to adverse conditions. Termed as *swarm intelligence,* a group of simple agents can be organized to perform high level tasks as a collective entity. A UAV swarm will carry out given missions as a collective entity so that, even if some members of the teams are lost, the whole team can still accomplish the given mission by reallocating the tasks to the surviving members.

Maintaining a swarm of UAVs from the perspective of guidance, navigation, and control poses many unseen challenges in mission allocation, coordination, and communication. As for path planning, when UAVs flies in a swarm at a higher altitude, a relatively simple approach may suffice. Constraints such as no-fly zone, if there is any, are usually known a priori and many existing off-line path planning algorithms[4] may be applied to solve for a globally optimal solution. The trajectories for individual vehicles obtained by the planner will be uploaded to each vehicle for tracking. However, when the vehicles need to fly close to the ground for close-range support, the path

[1] Assistant Professor, Department of Aerospace Engineering, KAIST, 335 Gwahangno, Daejeon, South Korea, Member, Corresponding Author (hcshim@kaist.ac.kr)
[2] Professor, Department of EECS, 320 Mc Laughlin Hall, University of California, Berkeley CA 94720

planning and envelope protection problems become very complicated. Firstly, the path planning requires a detailed and accurate geometry of terrain and objects such as buildings, trees or power lines, which may not be detailed or accurate enough and therefore should be collected on the fly using onboard sensors. Secondly, when vehicles travel together as a swarm in a dynamic environment, one agent may affect the behavior of others in an unpredictable manner. For example, one may inadvertently enter other vehicles' perimeter due to control error or urgent needs to avoid imminent collision from obstacles or other vehicles. Therefore, for a safe operation of a UAV swarm in an environment with obstacles, each vehicle should be equipped with a real-time collision avoidance capability.

Precedent to swarming, there have been strong research interests in the formation flight of UAVs[1,2,3] as an attempt to improve fuel efficiency using the leading aircraft's wingtip vortex[14] or as an operational procedure[2]. Swarming is usually considered as a group behavior of a significantly large number of agents while the formation flight is mainly about the precision control of relative distances among smaller number of vehicles to maintain the required shape of the overall group. Therefore, in this paper, we interpret the concept of swarming as a collective behavior of a large number of UAVs flying in close proximity, not necessarily maintaining certain shape (as in V-formation or similar).

In this paper, for swarming scenarios in a complex environment(Fig. 1), we propose a hierarchical system that consists of a global path planner as the upper layer and a local trajectory replanner as the lower layer. Using the map available prior to mission, the path planner generates a reference trajectory for each vehicle that leads to the destination without conflict. Based on the reference trajectory, the local planner solves for a conflict-free trajectory based on the information about the locations of other vehicles or any unexpected obstacles, which is collected using onboard sensors or by inter-vehicular communication. For trajectory replanning, a model predictive approach[8,12] is chosen, which solves for a trajectory that minimizes the cost function on the tracking error and potential collision. Such online optimization over a finite horizon suits our scenario very well since it can take latest information and prediction of on the surrounding environment into consideration. The tendency of model predictive approach to converge to local minima during global path planning can be avoided by combining it with the higher-level global path planner.

In this paper, we introduce the formulation of local planner based on model predictive approach while we opt to avoid a detailed discussion of global planner as it has been exhaustively investigated in many literatures including those from the robotics community. We will present the overall architecture of the proposed swarming controller and validate it in a complex scenario of a deployment of a UAV swarm in an area full of buildings.

Figure 1. A swarm of UAVs (conceptual drawing)

II. Formulation

In this paper, we are particularly interested in the conflict-free path generation problem of a swarm of UAVs in a complex environment. Therefore, we need to find a path for each UAV that leads from one point to another without any collision with obstacles or other vehicles. Under the assumption of complete knowledge on the surroundings and perfect tracking performance of each vehicle, a traditional path planning approaches can be used to compute collision-free trajectories for all participating UAVs using a centralized algorithm. However, the assumption needed here is highly impractical. In reality, the mapping always contains error, and, the individual motion of a large number of UAVs cannot be known during the path planning stage due to the disturbance, tracking error, interaction among vehicles, or any unpredictable changes. The swarming of large number of UAVs is challenging because traditional path planning methods cannot be applied to a partially known dynamic environment and uncertainty of the vehicles' motion. In this research, we propose to compliment the high-level path planner with a model-predictive trajectory replanner, which solves for conflict-free trajectory over a finite horizon.

A. MPC Formulation

Suppose we are given a nonlinear time-invariant dynamic system such that

$$x(k+1) = f(\mathbf{x}(k)) + g(x(k))u(k) \tag{1}$$

$$y(k) = h(x(k)) \tag{2}$$

where $x \in X \subset \mathbb{R}^{n_x}, u \in U \subset \mathbb{R}^{n_u}$. The optimal control input sequence over the finite receding horizon N is obtained by solving the following nonlinear programming problem:

$$\text{Find } u(k),\ k = i, \ldots i+N-1 \text{ such that } u(k) = \arg\min V(x,k,u) \tag{3}$$

where

$$V(x,k,u) = \sum_{i=k}^{k+N-1} L(x(i),u(i)) + F(x(k+N)) \tag{4}$$

where L is a positive definite cost function and F is the terminal cost. Suppose $u^*(k),\ \ k = i, \ldots, i+N-1$ is the optimal control sequence that minimizes $V(x,k,u)$ such that $V^*(x,k) = V(x,k,u^*(x,k)) \le V(x,k,u)$, $\forall u(k) \in U$. The cost function term L is chosen such that

$$L(x,u) = \frac{1}{2}\left(x^r - x\right)^T Q\left(x^r - x\right) + \frac{1}{2}u^T R u + S(x) + \sum_{l=1}^{n_o} P(x,\eta_l) \tag{5}$$

The first term penalizes the deviation from the original course. The second term penalizes the control input. $S(x)$ is the term that penalizes any states not in X as suggested in Ref. 11. Finally, $P(x_v,\eta_l)$ is to implement the collision avoidance capability in this MPC framework: $P(x_v,\eta_l)$ is a function that increases with bound as $\| x_v - \eta_l \|_2 \to 0$, where $x_v \in \mathbb{R}^3$ is the position of the vehicle and η_l is the coordinates or l-th out of total n_o obstacles being simultaneously tracked. As well known, MPC-based approaches require online optimization. During this process, the control input can be enforced to meet the saturation requirement. It is done by enforcing

$$u_i(k) = \begin{cases} u_i^{\max} & if\ u_i > u_i^{\max} \\ u_i^{\min} & if\ u_i < u_i^{\min} \end{cases} \tag{6}$$

where $u \triangleq \left[u_1 \ \ldots \ u_{n_u}\right]^T$. In this manner, one can find the control input sequence that will be always within the physical limit of the given dynamic system. We use the optimization method based on indirect method of Lagrangian multiplier suggested in Ref. 10.

B. Obstacle Sensing and Trajectory Replanning

For swarming scenario, it is mandatory to know where the nearby vehicles are located, in addition to obstacles if present. Due to the large number of agents, it is highly desired to sense those using onboard sensors, not receiving the broadcast information on a wireless communication channel. Obstacle detection can be done in either active or passive manner and the choice depends on many factors including operating condition, accuracy, and maximum detection range. Laser scanning method[12] can be very accurate and straightforward, so it is favored for short-range detection and three-dimensional mapping. However, as the detection range depends on the intensity of the light that radiates from the laser source, the range depends on the class of the laser head. Active radar has similar attributes since it operates in a similar principle: however, the resolution is much lower while the detection range can be significantly longer. Both methods are not applicable when the mission should be a covert one. For such cases, vision-based methods are favored as it is a passive sensing method and can offer a wealth of information if processed adequately. The ranging of objects using 2-D cameras can be performed either by using stereo parallax or optic flow algorithms.

For collision avoidance, as validated in Ref. 12, we choose $P(x_v,\eta_l)$ in Eq. (5) such that

$$P(x_v, \eta_l) = \frac{1}{(x_v - \eta_l)^T G(x_v - \eta_l) + \varepsilon}, \tag{7}$$

where G is positive definite and $\varepsilon > 0$ is to prevent ill conditioning when $\| x_v - \eta_l \|_2 \to 0$. One can choose $G = diag\{g_x, g_y, g_z\}$, $g_i > 0$ for an orthogonal penalty function. The penalty function Eq. (7) serves as a repelling field and has nonzero value for entire state space even when the vehicle is far enough from obstacles. The crucial difference from the potential field approach here is that we optimize over a finite receding horizon, not only for the current time as in the potential field approach. For obstacle avoidance, we consider two types of scenarios: 1) a situation when the vehicle needs to stay as far as possible from the obstacles even if no direct collision is anticipated and 2) a situation when the vehicle can be arbitrarily close to the obstacle as long as no direct conflict is caused. For the second situation, one can choose to enable Eq. (7) only when $\| x_v - \eta_l \|_2 < \sigma_{\min}$, where $\sigma_{\min}$ is the minimum safety distance from other vehicles.

Since MPC algorithms optimize over a receding finite horizon into future, for moving obstacles, their predicted trajectories over $k = i, \ldots, i + N - 1$ are needed in Eq. (7). It is observed that the inclusion of predicted obstacle locations in the optimization will produce more efficient evasion trajectory if the prediction is reasonably accurate. The simplest yet reasonable estimation is the extrapolation using the current position and velocity over the prediction horizon such that

$$\eta_l(k+i) = \eta_l(k) + \Delta t v_l(k)(i-1), \tag{8}$$

It is noted that the prediction can be done in more elaborated manners using a Kalman filter[15], exploiting any knowledge available on the motion of obstacles.

C. System Architecture

For swarming of UAVs, unlike some swarming scenarios where agents can physically touch one another, each UAV should stay clear from other vehicles and nearby objects to avoid damage. Due to the highly volatile situation that involves many active agents and obstacles including those that are not identified *a priori*, it is practically impossible to solve for a globally optimal path using conventional path planning methods, which typically take very long time to run and require knowledge over the entire area. On the other hand, the model predictive approach has been shown effective to solve for collision-free trajectories[8,11,12] in real time, but it may suffer from local minimal in a situation like *cul-de-sac*. Therefore, we propose to combine the global planner with the MPC-based trajectory replanner so that the globally optimal path can be generated using the best information available and, during the flight, the local replanner generates collision-free trajectory if needed. The adjustment of the given trajectory is performed by the model predictive approach introduced in Section II-A. The trajectory replanning algorithm is run on each vehicle in a fully decentralized manner so that the computing and communication loads can be distributed in contrast to many algorithms that assume a central planner with high-bandwidth communication. The overall system architecture is show in Fig. 2. The geographic information database can be constantly updated by the local map built locally on each agent using onboard sensors. In cooperative scenarios, information collected by other vehicles in vicinity can be shared along with their location and future intentions.

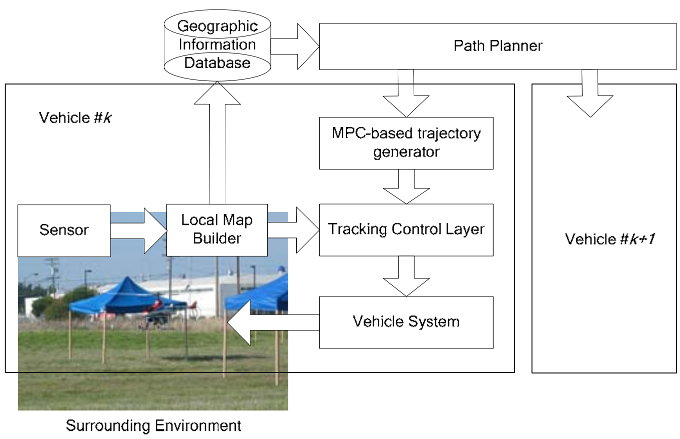

Figure 2. System architecture of swarming algorithm scenario

III. Simulation Results

In order to validate the swarming algorithm proposed above, we consider a scenario where sixteen UAVs are to be deployed into an urban are. Fig. 3 shows aerial photographs of Fort Benning in Georgia, where the scenario takes to occur. This area is partitioned into a number of cells and each cell is assigned with unique numbers. The geometry of buildings and their locations are known a priori. In this area, some of corridors that the UAVs are required to fly through is as narrow as five meters. Therefore the scenario calls for small-size rotorcraft UAVs that can fit into the area and fly slow enough. In this scenario, sixteen helicopter UAVs are initially flying in a four by four formation, and they should fly to the designated cells. After initial formation, they are required to divide into three subgroups. Some of these subgroups ingress into the area below the roof line (Team A and C) between the buildings while others fly over the roof top (Team B).

With this scenario, the path planner computes initial paths that are clear from the buildings, but not necessarily from other agents. In fact, if we were to consider the inter-vehicular conflict at the initial planning, the computation can be not only heavy due to iteration but also wasteful if the vehicle deviates from the initial planning due to any unforeseen event occurs. The model predictive layer plays an important role here when such events do occur; the online trajectory layer solves for safe trajectory by minimizing the cost function that penalizes the near-collision conditions.

The simulation is implemented using MATLAB/Simulink. Sixteen blocks of a detailed helicopter model with fifteen states and nonlinear kinematics matched with a MPC solver are built and run in parallel during simulation. The vehicles' positions are shared among the sixteen blocks, assuming the existence of a communication channel that broadcast the position information at the sampling rate of 40 ms, which is deemed rather fast in reality. More study will be performed on the necessary communication rate. We impose a 30 m sensing limit of other agents. Since the model predictive algorithm is numerically heavy, it is implemented using the CMEX feature of Simulink to speed up. The core algorithm is architected to be compatible with CMEX and the flight control software in C/C++ so that the code validated in MATLAB can be readily ported into the actual flight software. In fact, the model predictive algorithm has been already implemented in the test vehicle shown in Fig. 4 and used to demonstrate MPC-based trajectory algorithm in Ref. 8. Tests show that the implemented model predictive algorithm may run in real-time on a Pentium III 700MHz core of an industrial PC board known as PC-104.

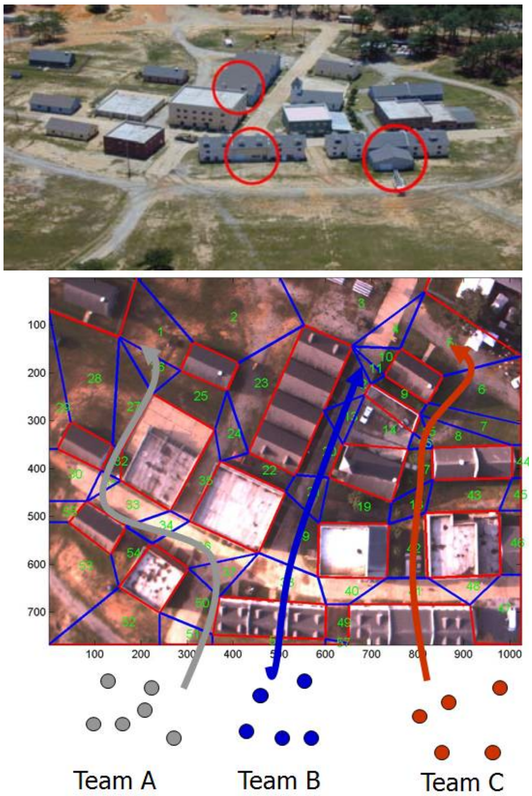

Figure 3. Aerial photographs of Fort Benning, Georgia.

Figure 4. A UAV testbed with flight control software enabled with model-predictive algorithm (UC Berkeley, 2006)

The simulation result is given in Fig. 5. Initially, the helicopter UAVs fly at 10m/s in a loose four by four formation (Fig. 5-(a)) where the trajectory generator only considers the tracking of the reference path segment generated by the path planner off-line and the collision avoidance with other agents. Then, just before they enter the town, they are divided into three subgroups. This behavior is achieved by generating a common reference path for each team. The collision-aware tracking capability commands each vehicle to divide into three subgroups nicely without any collision (Fig. 5-(c)). As they approach the area with fixed obstacles, i.e., the buildings, the collision avoidance term Eq. (8) in Eq. (5) begins to increase substantially. In this simulation, since the algorithm is fully decentralized, the vehicles sometimes show mob-like behavior: as one vehicle is about to enter the corridor, due to the collision avoidance cost, it is pushed away from by the following vehicles from the opening of the corridor while the followers succeed to enter the corridor. This observation suggests a coordination among neighboring UAVs may be necessary for more efficient deployment. In Fig. 5-(f), all UAVs arrives their destinations without any collision.

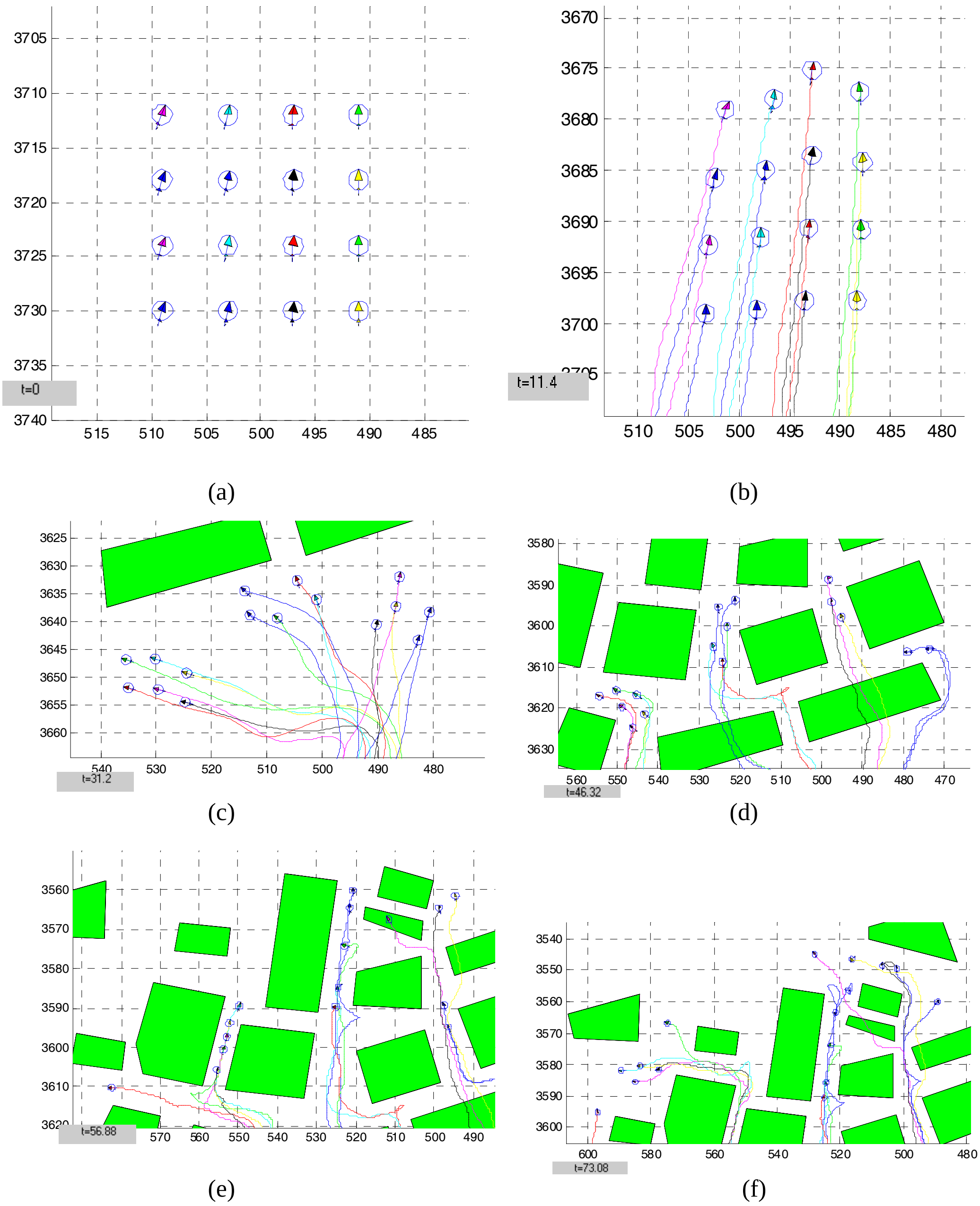

Figure 5. Simulation result of a swarming into an urban area

IV. Conclusion

This paper presented a hierarchical system for swarming. The upper layer is responsible for offline path planning while the lower is the trajectory replanner based on model predictive algorithms. The complementary structure generates initial path globally optimal offline and adjusts it if collision is anticipated. The proposed algorithm is applied to a deployment scenario in a cluttered urban environment and shows the fully decentralized approach demonstrates satisfactory performance. It is also observed that the overall behavior may improve further if coordination among neighboring agents can be achieved under the constraints of communication bandwidth. In near future, the proposed algorithm will be applied to a series of experiments using the rotorcraft UAV shown in Fig. 4.

V. Acknowledgments

This research is funded by US ARO SWARMS (W911NF-0510219).

References

[1]Fowler, J. M. and D'Andrea, R., "A Formation flight Experiment," *IEEE Control Systems Magazine*, vol. 23, no. 5, pp. 35-43, Oct. 2003.

[2]Chung, H., and Sastry, S., "Autonomous Helicopter Formation using Model Predictive Control", *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, Colorado, Aug. 21-24, 2006

[3]Dunbar, W., and Murray, R. M., "Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization," Automatica, Vol. 42, No. 4, pp. 549-558, 2006.

[4] Latombe, J-C., *Robot Motion Planning*, Kluwer Academic Publishers, 1991.

[5]Frazzoli, E., "Quasi-Random Algorithms for Real-Time Motion Planning and Coordination," *Acta Astronautica*, 54(4-10):485-495, 2003.

[6]Pettersson, P. O., and Doherty, P., "Probabilistic Roadmap-based Path Planning for an Autonomous Unmanned Helicopter," *SAIS-SSLS*, 2005

[7]Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P.O.M., "Constrained model predictive control: stability and optimality," *Automatica*, 36(6), 789-814, 2000.

[8]Shim, D. H. and Sastry, S., "A Situation-aware Flight Control System Design using Real-time Model Predictive Control for Unmanned Autonomous Helicopters," *AIAA Guidance, Navigation, and Control Conference*, Keystone, Colorado, 2006.

[9] Polak, E., *Optimization: Algorithms and Consistent Approximations*, Springer, New York, 1997.

[10]Sutton G. J., and Bitmead, R. R., "Computational Implementation of NMPC to Nonlinear Submarine," in F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, volume 26, pp. 461-471, Birkhäuser, 2000.

[11]Shim, D. H., Kim, H. J., and Sastry, S., "Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots," *IEEE Conference on Decision and Control*, Maui, HI, December 2003.

[12]Shim, D. H., Chung, H., Sastry, S., "Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles," *IEEE Robotics and Automation Magazine*, vol. 13, pp. 27-33, Sep. 2006.

[13]Ergen, M., Lee, D., Sengupta, R., and Varaiya, P., "Wireless Token Ring Protocol-Performance Comparison with IEEE 802.11", *Proceedings of the Eight IEEE International Symposium on computers and Communication*, 2003

[14]Wagner, Jr., E. H., *An Analytical Study of T-38 Drag Reduction in Tight Formation Flight,* Thesis, Air Force Institute of Technology, 2002.

[15]Watanabe. Y., Calise, A. J., Johnson, E. N., and Evers, J. H., "Minimum-Effort Guidance for Vision-based Collision Avoidance," *AIAA Guidance, Navigation, and Control Conference*, San Francisco, California, 2005.