Motion Planning in an Uncertain Environment: Application to an Unmanned Helicopter

Joshua D. Davis and Suman Chakravorty

Abstract—**In this work we present a methodology for intelligent motion planning in an uncertain environment using a non-local sensor, such as a radar. This methodology is applied to an unmanned helicopter navigating a cluttered urban environment. We show that the problem of motion planning in a uncertain environment, under certain assumptions, can be posed as the adaptive optimal control of an uncertain Markov decision process, characterized by a known, control dependent system, and an unknown, control independent environment. The strategy for motion planning then reduces to computing the control policy based on the current estimate of the environment, also known as the "certainty equivalence principle" in the adaptive control literature. Our methodology allows the inclusion of a non-local sensor into the problem formulation, which significantly accelerates the convergence of the estimation and planning algorithms. Further we show that the motion planning and estimation problems, as formulated in this paper possess special structure which can be exploited to significantly reduce the computational burden of the associated algorithms. We apply this methodology to the problem of motion planning for an unmanned helicopter in a partially known model of the Texas A&M campus.**

I. INTRODUCTION

In this paper, a methodology for motion planning of an autonomous agent in an uncertain environment is proposed and applied to an unmanned helicopter navigating a cluttered urban environment. The optimal path for the unmanned helicopter is planned using a priori knowledge about the environment and non-local sensor data as the helicopters navigates through obstacles in the environment. The motion planner involves a high-level planner which plans against the uncertainty in the environment and issues it commands in a series of waypoints for a lower-level controller to track. A commercially available lower-level controller was interfaced with the high-level motion planner so that the motion planning algorithm could be implemented on a six degree of freedom flight simulator.

The high-level controller implements "intelligent motion planning" in an uncertain environment using a radar sensor. The methodology is applicable for any non-local sensor, i.e., a sensor that allows sensing of environment states surrounding the current system state.

The state space of any motion planning problem can be expressed as the ordered pair *(s,q(s))* were *s* represents the system state and *q(s)* represents the state of the environment at the state s. For example, in the case of an unmanned helicopter exploring an urban environment, s corresponds to the *(x,y,z)* coordinates of the helicopter and *q(s)* corresponds to whether there is an obstacle or no obstacle at the point *(x,y,z)*. The goal of the motion planning strategy is to use all available information about the environment, until the current time instant, in order to plan the "best possible" path. It is known that the planning problem can be modeled as a Markov decision process, characterized by a known, control dependent exploration system and unknown, uncontrollable environment [1], [2]. Our formulation allows the integration of a radar or a similar non-local sensor into the planning methodology.

There has been substantial research in the adaptive control of controlled Markov Chains, or Markov Decision Processes, in the past two decades. In indirect adaptive control, the transition probabilities of the underlying Markov chain are estimated and the control is applied based on the most recent estimate of the transition probabilities [3]-[5]. This is known as the so-called "certainty equivalence principle". The "direct" approach to stochastic adaptive control falls under the category of "reinforcement learning" methodologies wherein the optimal control is calculated directly with resorting to estimating the transition probabilities of the underlying Markov chain [6]-[8]. Underlying all these methods is Bellman's "principle of optimality" or Dynamic Programming, which is a methodology for sequential decision-making under uncertainty [9]. In this work, it is shown that the motion planning problem can be reduced to the adaptive optimal control of a Markov decision process and thus the above methodologies can be applied to the same. The indirect approach to adaptive control is adopted since mapping the environment is of interest too.

Motion planning for unmanned systems has been an active area of research over the past few decades. Various different approaches have been devised for the collision-free path planning of mobile robots in known environments [10]. In the past decade, there has been an increasing interest in

Manuscript received March 17, 2006. This work was supported in part by the U.S. Air Force Research Laboratory.

J. D. Davis is a graduate student researcher in the Aerospace Engineering Department at Texas A&M University, College Station, Texas 77843 USA (phone: 979-204-1098; fax: 979-845-6051; e-mail: joshuaddavis@gmail.com).

S. Chakravorty is with the Department of Aerospace Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: schakrav@aeromail.tamu.edu).

the case when the environment in which unmanned system is operating is partially or completely unknown. The uncertainty in the environment is treated on a deterministic worst case [11], [12] or in a probabilistic average case basis [13]. In "probabilistic robotics", there has been substantial research in the localization of a mobile robot while simultaneously mapping the environment [14]-[18]. In [13], a game-theoretic framework is proposed for robotic motion planning. The authors resort to Bellman's principle of optimality [9] in order to tackle the motion-planning problem.

The original contributions of the current work are as follows. We propose a hierarchical motion planner wherein, the problem of "intelligent motion planning" for the high-level planner is reduced to the adaptive optimal control of an uncertain Markov decision process, characterized by a known, control dependent system and an unknown, control independent environment; and a low-level controller is then used to track the commands from the high-level planner. Posing the problem in this fashion significantly reduces the computational burden of the estimation and planning algorithm. The motion planner is applied to the problem of a UAV helicopter navigating the Texas A&M campus and is tested in a six degree of freedom flight simulator of the helicopter.

The rest of the paper is organized as follows. Section II details the structure of the motion planner. Section III contains the formulation of the motion planning problem as a Markov decision problem. In section IV, we present the results of implementing the planning methodology on a UAV helicopter navigating the Texas A&M campus by testing the algorithms on a six DOF flight simulator.

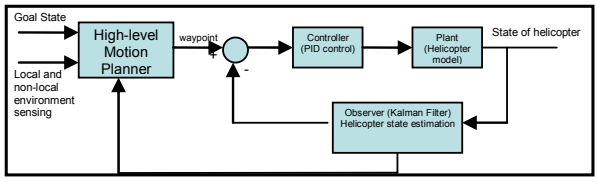

Fig. 1. Control Architecture for the high-level and low-level motion planner

II. Controller Structure

The motion planning algorithm is implemented in a hierarchical fashion wherein the high-level motion planner determines the optimal waypoints for the low level controller to track during its traverse to the goal state. The planner takes into consideration the shortest distance to the goal state, while avoiding obstacles along the way. After each waypoint is issued to the low-level controller, the motion planner waits for the waypoint to be achieved. Once the waypoint is achieved, the motion planner takes into consideration the goal state, local and non-local sensor data, collected during the past, and the state of the helicopter to generate a new waypoint. This process continues until the goal state is achieved. Figure 1 below outlines the basic motion planning architecture.

The low-level flight controller used in the simulations was developed by Rotomotion LLC and is capable of executing waypoint commands given to the helicopter. This flight controller has a series of PID control loops that stabilize the attitude, position, and velocity of a remote control helicopter using an observer (state estimator), which is implemented using a Kalman Filter. As the helicopter flies, the Kalman Filter also gives state feedback to the high-level motion planner so that the high-level planner knows that a waypoint has been achieved. The motion planner is in the form of composite feedback control wherein the high-level planner plans against the uncertainty in the environment on a longer time/length scale while the lower-level controller is robust to uncertainties at shorter time/length scaling, and dynamic uncertainties in the system model.

III. Motion Planning Under Uncertainty

First, we recount some results that will be required for the motion planning [1].

A. Preliminaries

Let the state of the exploration system be denoted by *s*, $s \in S$. Denote the state of the environment at the system state *s* by *q(s)*. Denote the local state of the exploration system by the ordered pair *(s, q(s))*. Denote any particular control action by *u*. The following Markovian assumption is made about the system. Let $F^t = \{(s_0, q_0(s_0)), u_0, ..., (s_{t-1}, q_{t-1}(s_{t-1})), u_{t-1}\}$ represent the history of the process till time *t*.

A **3.1** T*he current system state, s_t, is dependent only on the system state and control input at the previous time instant, i.e.,*

$$p(s_t / F^t) = p(s_t / s_{t-1}, u_{t-1}). \quad (1)$$

A **3.2** *The environment process is "incoherent", i.e., the environment process is spatially uncorrelated and temporally stationary. In other words, if $\{q_t(s), s \in S\}$ denotes the environment process, $q_t(s)$ is a stationary process for all $s \in S$. Moreover, $q_t(s)$ is independent of $q_\tau(s')$ whenever $s \neq s'$, for all t, τ. Note that $q_t(s)$ is a random variable and the above assumption is used in the probabilistic sensing..*

Proposition 1 *Under assumptions A3.1,A3.2, the following holds:*

$$p((s_t, q_t(s_t)) / F^t) = p(s_t / (s_{t-1}, u_{t-1})) p(q_t(s_t)) \quad (2)$$

The transition probabilities $p(s_t / (s_{t-1}, u_{t-1}))$ quantify the control uncertainties inherent in the system and are assumed to be known beforehand. The environmental uncertainty *p(q(s))* is unknown and successive estimates are made of this uncertainty as the planning proceeds to completion. Motion planning may be framed as an infinite horizon discounted stochastic optimization problem, i.e., given the initial state

$(s_0, q_0(s_0))$, the optimal control policy $\mu^*(s_0, q_0(s_0)) = \{\mu_1, \mu_2, ...\}$ is defined by:

$$\mu^*(s_0, q_0(s_0)) = \arg\min_\mu E_\mu \left(\sum_{t=1}^{\infty} \beta^t c\left(\left(s_t, q_t(s_t)\right), \left(s_{t-1}, q_{t-1}(s_{t-1})\right), \mu_{t-1} \right) / \left(s_0, q_0(s_0)\right) \right) \quad (3)$$

where $c\left(\left(s_t, q_t(s_t)\right), \left(s_{t-1}, q_{t-1}(s_{t-1})\right), \mu_{t-1} \right)$ is a positive pre-defined cost that the system incurs in making the transition from state $\left(s_{t-1}, q_{t-1}(s_{t-1})\right)$ to $\left(s_t, q_t(s_t)\right)$ under the control action μ_{t-1}, $E_\mu(.)$ denotes the expectation operator with respect to the policy μ, and $\beta < 1$ is a given discount factor.

The following environment sensing model is adopted:

At every instant *t*, the system (UAV) at state s_t, can observe the environment state $q_t(s)$, (i.e., the current environment state at the state *s*), if $s_t \in F(s) \subseteq S$, where *F(s)* is assumed to be known beforehand. The set *F(s)* constitutes a "footprint" of the sensor system.

Associated with every observation-vantage point pair, *(q(s),s′), s′∈ F(s)*, (i.e., we are observing the environment at *s*, *q(s)*, from the state *s′*), there exists a known measurement error model, $p(\hat{q}(s)/q(s), s'), \hat{q}(s), q(s) \in Q$, and *s*, *s′* ∈ *S*.

B. Heuristic Analysis

B.1 Estimation

Consider the following relationship:

$$\pi(\hat{q}(s)) = \frac{1}{\pi(F(s))} \sum_{s', q(s)} p(\hat{q}(s)/q(s), s') p(q(s)) \pi(s') \quad (4)$$

where

- $\pi(\hat{q}(s))$ denotes the probability of observing the noise corrupted environment state $\hat{q}(s)$ during the course of the exploration, i.e., the fraction of the time that the environment at state *s* is observed to be at $\hat{q}(s)$ during the course of the exploration.
- *p(q(s))* denotes the true probability that the environment state is *q(s)* at the state *s*.
- $\pi(s')$ denotes the fraction of the time that the system is at state *s′* and $\pi(F(s))$ represents fraction of time that the system spends in the set *F(s)*.

Since the noise model is known, and the values of $\pi(\hat{q}(s))$ and $\pi(s)$ can be estimated during the course of the exploration using the Monte-Carlo method, it is possible to obtain the true environment probabilities using equation 4. Mathematically:

$$\pi_t(\hat{q}_n(s)) := \frac{1}{t} \sum_1^t 1(\hat{q}_n(s) = \hat{q}(s)) \quad (5)$$

$$\pi_t(s) := \frac{1}{t} \sum_{n=1}^t 1(s_n = s) \quad (6)$$

where *1(A)* denotes the indicator function of the event *A*. Then, the true probabilities of the environment process *p(q(s))* can be obtained recursively as:

$$P_t(s) := \arg\min_{P \in \bar{V}} \| \Pi_t(s) - \Gamma_t(s) P \|^2 \quad (7)$$

where

$$P_t(s) = \left[p_t(q_1(s)), ..., p_t(q_D(s)) \right]' \quad (8)$$

$$\Pi_t(s) = \left[\pi_t(\hat{q}_1(s)), ... \pi_t(\hat{q}_D(s)) \right]' \quad (9)$$

$$\Gamma_t(s) = \alpha(s) \left[\gamma_t^{ij}(s) \right] \quad (10)$$

$$\gamma_t^{ij}(s) = \sum_{s'} p(q_i(s)/q_j(s), s') \pi_t(s') \quad (11)$$

$\alpha(s)$ is a normalizing factor that renders the matrix $\Gamma(s)$ stochastic, $\|.\|$ denotes the Euclidean norm in $\mathbb{R}^D$, and $\bar{V}$ represents the space of all probability vectors in $\mathbb{R}^D$.

B.2 Control

Consider the stochastic optimal control problem posed in equation 3. Using the Bellman principle of optimality [5], it can be shown that the optimal policy is stationary, i.e., the optimal control is independent of time, and that the optimal control at the state (s,q(s)), $\mu^*(s, q(s))$, is given by the following equation:

$$u^*(s, q(s)) = \arg\min_u \sum_{(r, \bar{q}(r))} p(r/s, u) p(\bar{q}(r)) \left[c((r, \bar{q}(r), (s, q(s)), u) + \beta J^*(r, \bar{q}(r)) \right], \quad (12)$$

where $J^*(r, \bar{q}(r))$ is the optimal cost-to-go from the state $(r, \bar{q}(r))$.

Let $T = \{t_1, t_2, ..., t_k, ...\}$ denote the set of all times at which the control policy is updated during the path planning. Let the updated control policy at time instant t_k be denoted by $\mu_k(s, q(s))$. Let $p_t(q(s))$ denote the estimated environmental uncertainty at the time *t*, obtained from the estimation equation 7. Then, the control update at time $t_k \in T, \mu_k(.)$, using the principle of optimality and the "certainty equivalence principle", is given by

$$\mu_k(s, q(s)) = \arg\min_u \sum_{(r, \bar{q}(r))} p(r/s, u) p_{t_k}(\bar{q}(r)) \left[c\left(\left(r, \bar{q}(r)\right), \left(s, q(s)\right), u \right) + \beta J_k(r, \bar{q}(r)) \right] \quad (14)$$

where

$$J_k(s, q(s)) = \min_u \sum_{(r, \bar{q}(r))} p(r/s, u) p_{t_k}(\bar{q}(r)) \left[c\left(\left(r, \bar{q}(r)\right), \left(s, q(s)\right), u \right) + \beta J_k(r, \bar{q}(r)) \right].$$

Next, we list a few of the salient properties of the control problem as presented above.

Proposition 2 *Under assumptions A2.1-A2.2, the cost-to-go functions* $J_t(s, q(s)) \to J^*(s, q(S))$ *as* $t \to \infty$, *if* $p_t(q(s)) \to p(q(s))$, *for all* $s \in S$, $q(s) \in Q$.

It can be shown that

$$u^*(s, q(s)) = \arg\min_u \left[\bar{c}(s, u, q(s)) + \beta \sum_r p(r/s, u) \bar{J}(r) \right],$$

(20)

Hence, we can conclude that we need only have an average value of the cost-to-go function $J(s,q(s))$, at the system state s, namely $\bar{J}(s) = \sum_{q(s)} p(q(s))J(s,q(s))$, in order to be able to evaluate the optimal control at any state $(s,q(s))$.

It can be seen that $\bar{J}$ is the fixed point of the "average" dynamic programming operator $\bar{T}:\mathbb{R}^N \to \mathbb{R}^N$, defined by the following equation:

$$\bar{T}\ \bar{J}(s) = \sum_{q(s)} p\left(q(s)\right).$$

$$\min_u \left[\bar{c}\left(s,u,q(s)\right) + \beta \sum_r p\left(r/s,u\right)\bar{J}(r) \right], \forall s \quad (22)$$

Proposition 3 *The average DP operator, $\bar{T}$, is a contraction mapping in R^N under the ∞ norm.*

IV. Testing on an Unmanned Helicopter

The methodology presented thus far in this paper was used for motion planning of an unmanned helicopter in an uncertain environment. The motion planning algorithms were implemented in a six DOF simulation of a UAV helicopter navigating through an urban environment model of the Texas A&M campus. The results show that the planning methodology proposed maybe suitable for autonomous navigation in a cluttered urban environment.

A. Developing an urban environment

The system state, *s*, represents the *(x,y)* grid point coordinate of the vehicle and the environment state, $q(s)$, represents the presence of an obstacle or otherwise at that particular grid point. A 750x 550 meter area on the campus of Texas A&M was discretized into 75 x 55 grid of points (*i* rows, *j* columns) so that the number of possible system states *N* was equal to 4125. The environment state at any grid point has 2 possible values, obstacle or no obstacle. At any given location the helicopter had four possible high-level control actions: Go (Forward/Back/Right/Left) to the adjacent grid point.

B. Non-local state estimation using a radar

A millimeter wave radar (24.7 GHz) called the Eaton VORAD (Vehicle On-board RADar) was chosen as the non-local sensor to be modeled in this research. The VORAD has an operating range of 1 meter to 107 meters (3 feet -350 feet), with an uncertainty of 5%, and a field of view of 12 degrees, with an uncertainty of 0.2 degrees. The radar antenna, onboard processor, and batteries require a helicopter with a lift capability of 6-7lbs. The radar software was developed to track up to seven obstacles every 65ms (15Hz), by reporting azimuth to each obstacle, range to each obstacle, and gain of the return from each obstacle. The simulation treats all obstacles as having the same radar reflectivity characteristics. Two changes were made to the sensing model to accommodate the simulation: first, the field of view was changed to 90 degrees and second, radar tracking was set to nine obstacles.

C. Interfacing with a low-level flight controller

A low-level flight controller known as the Automated Flight Control System (AFCS), developed by Rotomotion Inc. for unmanned helicopters, was used to track the high-level flight control commands. The AFCS uses two extended Kalman filters for state estimation: the first is a 7 state Kalman filter, 4 states for quaternion and 3 states for gyro bias, and the second is 6 state Kalman filter, 3 states for North East Down *(NED)* position and 3 states for orthogonal velocity components *(UVW)*. The flight controller implements nine Proportional Integral Derivative (PID) loops- three for position *(x,y,z)*, three for velocity $(\dot{x},\dot{y},\dot{z})$, and three for attitude (ψ,θ,ϕ).

D. Motion Planning in the Flight Simulator

D.1 Navigation and Exploration

The high-level motion planner calculates an initial cost-to-go map $\bar{J}(s)$, which is an array of *i* rows and *j* columns, based upon the a priori environment data available. Before the high-level motion planner implements a control, it senses and updates probabilities of the environment state at 8 of the adjacent grid-points. Then the motion planner recalculates the control and the command is issued to the low-level flight controller for execution.

The operation of the high-level planner involves four different tuning parameters. The cost associated with hitting an obstacle is ζ= *100*, while the transition cost to visit a state with no obstacle is η= *1*. Various tuning values were used for ζ and η. An infinite horizon discount factor of $\beta = .99$ was used to ensure that states in the immediate future have a greater effect on the control than states farther out on the horizon. For lower β values the vehicle is more likely to be trapped once the vehicle enters a boxed in area. Also a tuning parameter of $\alpha = 6$ was included in the simulations to represent a weighting value for distance to the goal. For large α values the vehicle is drawn to the final destination and hits obstacles, while the value of $\alpha = 6$ draws the vehicle to the final destination but does not hit obstacles. It is necessary to tune the parameters because the controller structure is posed as an infinite horizon problem.

Figures 2 through 5 included an a priori environment model accuracy of 90%. This number quantifies the reliability of the a priori environment model. Various levels of a priori environment model accuracy were used in the initial testing of the algorithm. If the a priori environment model accuracy was treated as $p_0\left(q(s)\right) = 0.5$, then the a priori model did not have a meaningful contribution to the motion planning (since a 50% reliability is as good as flipping a fair coin). As the a priori accuracy of the model increased from 0.5 to 0.9 the initial motion plan became more reliable and less likely to hit obstacles.

The sensor model was generated by simulating sensor data using a Monte Carlo approach for development of the sensor model. The simulations using the Eaton VORAD model had

an accuracy of $p(q(s')/q'(s'),s)=0.95$ for the non-local state sensing; i.e. when $(s' \neq s)$.

The simulations were also performed with various different uncertainty models so that the effects of using non-local sensors, with different accuracy levels, on the performance of the algorithm could be quantified. For example, in figures 2, 4, and 5 the non-directional sensor model has an accuracy of $p(q(s')/q'(s'),s)=0.96$ for the local sensing, an accuracy of $p(q(s')/q'(s'),s)=0.93$ at a distance of 10 meters, and an accuracy of $p(q(s')/q'(s'),s)=0.91$ at a distance of 14 meters. This model was developed by assuming non-local measurements at 1 meter have an accuracy of $p(q(s')/q'(s'),s)=0.96$, while measurements at 100 meters have an accuracy of $p(q(s')/q'(s'),s)=0.57$. The accuracy of the radar at a certain range was found by linearly interpolating between the accuracy of the sensor at 1 meter and 100 meters. The figure 3 use a directional sensor model with an approximate accuracy of $p(q(s')/q'(s'),s)=0.99$. For a sensor uncertainty of $p(q(s')/q'(s'),s)=0.5$, the vehicle performed as if no sensor data was available. For sensor uncertainties between $p(q(s')/q'(s'),s)=0.7$ and $p(q(s')/q'(s'),s)=0.8$, the performance increased significantly.

The motion planning software was tested multiple times before executing the waypoints in the flight simulator. Figures 3 has been included below to show a path that the high-level flight controller generated to navigate through campus while using a priori map data as well as radar sensor information. The plots have been overlaid with aerial photos to display the simulated vehicle trajectory through the Texas A&M campus.

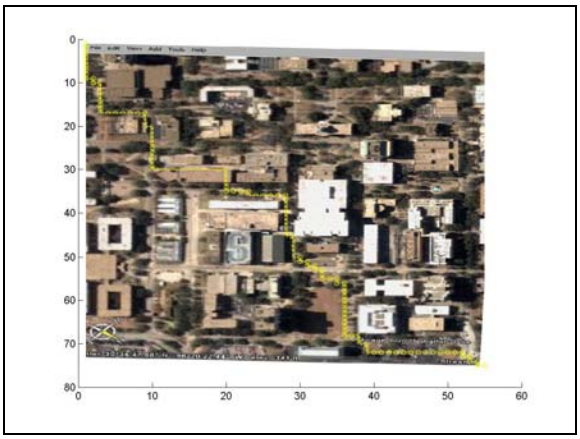

Fig. 2. Top view of navigation through campus

D.2 Avoiding obstacles

The simulations in figure 3 used a static environment model. However, it is realistic to consider the situation where the environment model can be dynamic. If a newly sensed obstacle such as a vehicle or fallen building is encountered, the high-level flight controller should be able to adapt. When the helicopter radar senses a new obstacle, it creates a large transition cost so that the helicopter avoids flying into the newly sensed obstacle. In figure 4, the trajectory of the helicopter is plotted in yellow. Additional obstacles were added to the map after the helicopter started moving, so that the helicopter encountered a dynamic obstacle field and a blocked passage by the Texas A&M Library.

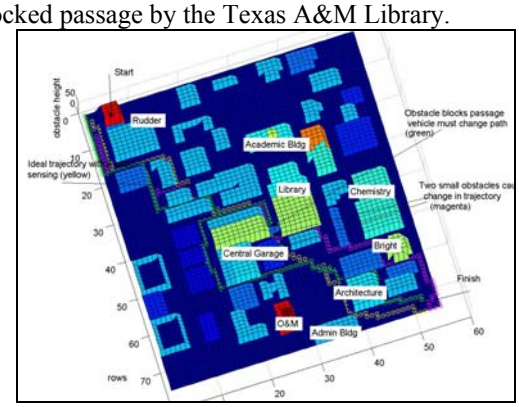

Fig. 3. Obstacles and blocked passage

D.3 Re-planning at a blocked passage

Occasionally, the vehicle may encounter an obstacle that entirely blocks a passage. The vehicle may not have time to recalculate the entire cost map J. In this paper, updating the local region's cost map (a 10x10 grid) is sufficient for the vehicle to plan a new trajectory to get out of an impasse. Figure 4 is included below to display the motion planner's response to blocked passages at various different locations on campus.

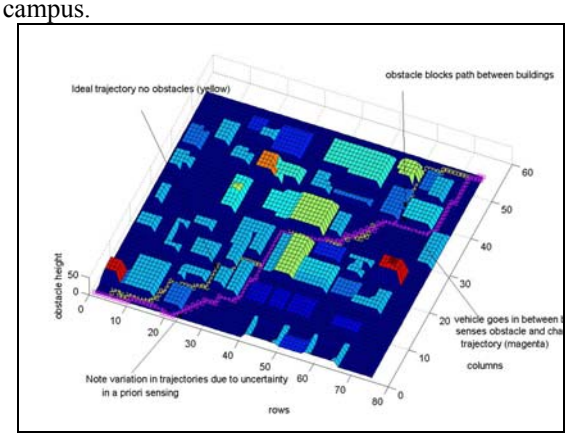

Fig. 4. Blocked passage by Langford

D.4 Motion planning in the flight simulator

Multiple tests were run while the high-level motion planner and low-level flight controller were connected together. The high-level controller computation time was relatively minor compared to the time necessary to execute the waypoint commands. Running each MATLAB simulation took approximately 15-20 seconds, while a simulation of actually flying the helicopter to 180 waypoints, which were 10 meters apart, took approximately 12-15 minutes. If 180 waypoints were executed during the initial cost to go calculation then approximately $180*N*M \approx$ 3,000,000 major

computations were required, which requires a computation time of approximately (5-10 sec). During the MATLAB simulations the vehicle made a minimum of 1 local and 8 non-local sensor measurements and up to 4 local and 32 non-local sensor measurements per waypoint. This resulted in approximately 1800 optimizations run on environment probabilities and 720 waypoint calculations if 180 waypoints were executed. This requires a computation time of approximately 5-10 seconds. The following plot (figures 5) demonstrate the performance of the motion planner and the flight simulator while connected together

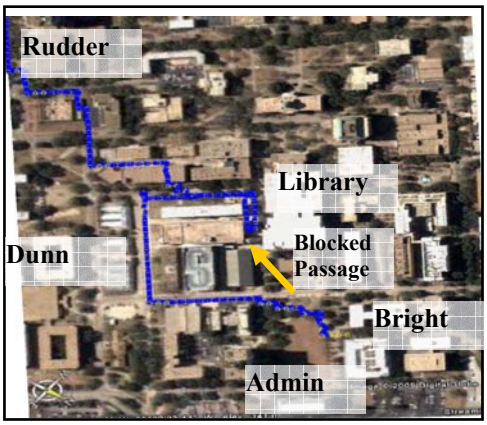

Fig. 5. Simulator testing 3- large view This includes a blocked passage by the Library in which the Helicopter recalculated to exit

V. Conclusion

In this work, we have presented a methodology for intelligent exploration of a partially known environment using a non-local sensor. We have shown that a motion planning problem, under certain assumptions, can be reduced to the adaptive control of an uncertain Markov decision process, consisting of a known control dependent system state and an unknown control independent environment. We have illustrated the feasibility of the planning methodology by testing it on an unmanned helicopter navigating through an urban environment in a six DOF flight simulation.

An area of interest is to consider implementing heuristics in the dynamic programming algorithm that would allow for an accelerated computation of the cost-to-go map by initially excluding the evaluation of states that are far from the desired path of the helicopter. Also, a low-level controller needs to be designed that guarantees the local avoidance of obstacles while tracking the high-level motion plans, so that integration of the high-level motion planner and low-level flight controller can result in a truly intelligent autonomous system.

References

[1] S. Chakravorty, J. L. Junkins, "Intelligent Exploration of Unknown Environments with Vision like Sensors", *Proceedings of the 2005 IEEE/ ASME International conference on Advanced Intelligent Mechatronics*, Monterey, CA, pp. 1204-1209

[2] S. Chakravorty, J. L. Junkins, "A Methodology for Intelligent Path Planning", *Proceedings of the 2005 IEEE International Symposium on Intelligent Control*, Limassol, Cyprus, pp. 592-597

[3] P. R. Kumar & P. Varaiya, *Stochastic Systems: Estimation, Identification and Adaptive Control*, Englewood Cliffs, NJ: Prentice Hall, 1986.

[4] V. Borkar & P. Varaiya, "Adaptive Control of Markov Chains, I: Finite parameter set," *IEEE Trans. Aut. Contr.*, vol. AC-24, pp953-958, 1979.

[5] P. Mandl, "Estimation and control in Markov chains," *Adv. Appl. Prob.*, vol. 6, 1974, pp. 40-60.

[6] D. P. Bertsekas & J. N. Tsitsiklis, *Neuro-Dynamic Programming,* Belmont, MA: Athena, 1996.

[7] R. S. Sutton & A. G. Barto, *Reinforcement Learning: An Introduction,* Cambridge, MA: MIT Press, 1998.

[8] R. S. Sutton, A. G. Barto & R. J. Williams, "Reinforcement Learning is Direct Adaptive Optimal Control," *IEEE Control Systems Magazine*, vol. 12, no. 2, Apr. 1992, pp. 19-22.

[9] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[10] J. C. Latombe, "Robot Motion Planning," Kluwer, Boston, MA, 1991.

[11] J. C. Latombe, A. Lazanas & S. Shekhar, "Robot Motion Planning with Uncertainty in Control and Sensing", *Artificial Intelligence*, 52:1-47, 1991.

[12] M. T. Mason, "Automatic Planning of Fine Motions: correctness and Completeness", *Proc. of IEEE Conf. Robotics. Automat*., pp 484-489, 1989.

[13] S. M. Lavalle, "Robot Motion Planning: A Game-Theoretic Foundation", *Algorithmica*, vol. 26, 2000, pp. 430-465.

[14] S. Thrun, "A Probabilistic On-Line Mapping Algorithm for Teams of Mobile Robots," *The International Journal of Robotic Research*, vol. 20, no. 5, May 2001, pp. 335-363.

[15] S. Thrun, "Probabilistic Algorithms in Robotics," *AI Magazine*, 21(4): 93-109

[16] W. Burgard, D. Fox, H. Jans, C. Matenar & S. Thrun, "Sonar-based Mapping of large-scale mobile robot environments using EM," *Proceedings of the International Conference on Machine Learning*, Bled., Slov., 1999.

[17] J. A. Castellanos, J. M. M Montiel, J. Neira & J. D. Tardos, "The SP map: A Probabilistic Framework for Simultaneous Localization and Mapping," *IEEE Transactions on Robotics and Automation*, vol. 15, 1999, pp. 948-953.

[18] G. Dissanayake, H. Durant-White & T. Bailey, "A Computationally Efficient Solution to the Simultaneous Localization and Mapping Problem," *ICRA'2000 Workshop W4: Mobile Robot Navigation and Mapping*, April 2000.